

CALCULATRICES

Loïc Fieux

HP 48

Le livre du programmeur
en RPL et RPL-système



DUNOD

Initiation et
références

CALCULATRICES

Loïc Fieux

HP 48

**Le livre du programmeur
en RPL et RPL-système**

DUNOD

Ce livre n'est pas le manuel des calculatrices Hewlett-Packard HP-48. Le contenu de ce livre n'engage pas la société Hewlett-Packard, ni ses distributeurs. Cet ouvrage est destiné aux utilisateurs des calculatrices Hewlett-Packard HP-48 versions S, SX, G, GX et autres versions compatibles à venir.

Ce pictogramme mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du **photocollage**.

Le Code de la propriété intellectuelle du 1er juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établisse-

ments d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation du Centre français d'exploitation du droit de copie (CFC, 3 rue Hautefeuille, 75006 Paris).



Ouvrage mis en page par l'auteur

Illustrations de couverture : Rachid Marai

© DUNOD, Paris, 1996

ISBN 2 10 0030 37 X

Code 043037

Toute représentation ou reproduction, intégrale ou partielle, faite sans le consentement de l'auteur, ou de ses ayants droit, ou ayants cause, est illicite (loi du 11 mars 1957, alinéa 1er de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal. La loi du 11 mars 1957 n'autorise, aux termes des alinéas 2 et 3 de l'article 41, que les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective d'une part, et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration.

Table des matières

Première partie : initiation

| | |
|---|----|
| 1 • Créer un programme RPL | 3 |
| 1 • <i>Préambule</i> | 3 |
| 2 • <i>Créer un programme</i> | 4 |
| 3 • <i>Variables utilisées par un programme</i> | 10 |
| 2 • Structures RPL | 15 |
| 1 • <i>Boucles</i> | 15 |
| 2 • <i>Tests</i> | 23 |
| 3 • <i>Gérer les erreurs</i> | 29 |
| 4 • <i>Sous-programmes</i> | 33 |
| 3 • Entrées et sorties | 35 |
| 1 • <i>Entrées</i> | 35 |
| 2 • <i>Sorties</i> | 42 |

| | |
|----------------------------------|----|
| 4 • Graphisme | 45 |
| 1 • <i>Commandes graphiques</i> | 45 |
| 2 • <i>Les objets graphiques</i> | 47 |
| 3 • <i>Animation graphique</i> | 53 |
| 5 • Le RPL en prépa | 57 |

Deuxième partie : références pour le RPL

| | |
|--|-----|
| 6 • Familles d'instructions | 71 |
| 1 • <i>Mathématiques et physique</i> | 71 |
| 2 • <i>Mémoire, variables et bibliothèques</i> | 80 |
| 3 • <i>Interface-utilisateur et graphisme</i> | 82 |
| 4 • <i>Listes et chaînes de caractères</i> | 88 |
| 5 • <i>Programmation et gestion des erreurs</i> | 90 |
| 6 • <i>Autres fonctions</i> | 92 |
| 7 • Index alphabétique des instructions RPL | 95 |
| 8 • Drapeaux, indicateurs ou flags | 267 |
| 1 • <i>Notion d'indicateur</i> | 267 |
| 2 • <i>Les indicateurs-système</i> | 271 |
| 9 • Variables réservées | 281 |
| 1 • <i>Variables réservées</i> | 281 |
| 2 • <i>Structure des variables réservées</i> | 282 |
| 10 • Codes clavier | 287 |

| | |
|--------------------------------|-----|
| 11 • Codes menus | 293 |
| 12 • Codes ASCII | 297 |
| 13 • Codes des messages | 303 |

Troisième partie : le RPL-système

| | |
|--|-----|
| 14 • Programmer en RPL-système | 309 |
| 1 • <i>Créer un programme en RPL-système</i> | 309 |
| 2 • <i>Exemples</i> | 315 |
| 15 • Objets | 321 |
| 1 • <i>Introduction</i> | 321 |
| 2 • <i>Entier système</i> | 324 |
| 3 • <i>Nombre réel</i> | 325 |
| 4 • <i>Réel long</i> | 326 |
| 5 • <i>Nombre complexe</i> | 327 |
| 6 • <i>Complexe long</i> | 328 |
| 7 • <i>Caractère</i> | 329 |
| 8 • <i>Chaîne de caractères</i> | 330 |
| 9 • <i>Entier binaire</i> | 331 |
| 10 • <i>Liste</i> | 332 |
| 11 • <i>Expression algébrique</i> | 333 |
| 12 • <i>Objet avec unité</i> | 334 |
| 13 • <i>Objet graphique</i> | 335 |
| 14 • <i>Programme RPL</i> | 336 |

| | |
|--|-----|
| 15 • <i>Nom global</i> | 337 |
| 16 • <i>Nom local</i> | 337 |
| 17 • <i>Remarques importantes</i> | 338 |
| 16 • Instructions RPL : classement alphabétique | 341 |
| 17 • Instructions RPL : classement par adresses | 347 |
| 18 • Constantes | 353 |
| 19 • Calculs et manipulation des nombres | 357 |
| 20 • Pile et ligne de commande | 373 |
| 21 • Listes et chaînes de caractères | 377 |
| 22 • Gestion des types | 379 |
| 23 • Manipulation des booléens | 385 |
| 24 • Variables, répertoires et mémoire | 387 |
| 25 • Gestion du clavier | 393 |
| 26 • Graphisme | 395 |
| 27 • Modes et flags | 399 |
| 28 • Communication série | 403 |
| 29 • Adresses en vrac | 405 |

Avant-propos



Après l'initiation (*HP-48 : permis de conduire*), le parascolaire (*HP-48 pour le bac*, *HP-48 : maths en prépa*, *HP-48 : physique et chimie en prépa*) et la programmation avancée (*HP-48 : faites vos jeux en assembleur*), nous traitons avec le présent ouvrage le dernier grand sujet non encore abordé par nos ouvrages consacrés à la HP-48 : la programmation en RPL. Bien que le RPL soit le langage officiel des HP-48, Hewlett-Packard ne fournit pas avec ses machines la documentation nécessaire au programmeur.

Ne souhaitant pas concurrencer directement le *manuel de référence pour l'utilisateur avancé* (HP) qui reste incontournable dans de nombreux domaines (équations, variables réservées, etc.), nous avons voulu proposer au lecteur une panoplie d'explications et de références lui permettant de programmer facilement en RPL et en RPL-système. Volontairement limité à 500 pages, cet ouvrage s'adresse avant tout aux débutants et souhaite aider ceux-ci à faire leurs premiers pas en programmation.

Bon courage à tous !

Avertissement

Afin de faciliter la lecture, le texte est balisé d'icônes :



Attention ! Ce pictogramme signale une erreur à éviter...



Chouette ! Une astuce !



Cet éléphant signale les paragraphes dont le contenu est à apprendre (une mémoire d'éléphant sera la bienvenue...).



Un petit commentaire...



La réponse à une question embarrassante



Cet expert donne parfois un petit conseil. Il signale aussi des informations à réserver aux initiés.

Les noms des touches à presser sont placés entre crochets. Ainsi, [ENTER] signifie « Appuyez sur la touche nommée ENTER ». Les noms des articles de menu sont placés en italique et entre parenthèses, par exemple, (BIN).

Toutes les manipulations élémentaires (saisie dans la pile, manipulation des variables et des répertoires, création de variable, etc.) doivent être maîtrisées. Si vous êtes débutant, consultez le livre *HP-48 permis de conduire* (DUNOD) avant d'aborder le présent ouvrage.

Dans le cadre de la programmation en RPL-système, le lecteur doit connaître la numération binaire, la numération hexadécimale, le DCB, les fonctions logiques, le codage hexadécimal des quartets, etc. Si ces notions vous sont étrangères, reportez-vous à l'ouvrage *HP-48 faites vos jeux en assembleur* (DUNOD).

En cas de problème, utilisez le service d'assistance technique (Voir *Assistance technique* en fin d'ouvrage).

1

Initiation



La maîtrise du contenu de cette première partie est essentiel. Vous y apprendrez :

- la saisie des programmes,
- les structures de programmation élémentaires,
- les entrées et sorties,
- la manipulation des objets graphiques,
- des exemples.

Bref, cette première partie rassemble tout ce dont vous avez besoin pour faire facilement vos premiers pas en RPL.

Créer un programme RPL

1

Préambule

■ Un programme RPL



Comme une liste, un programme RPL contient différents objets. Parmi ceux-ci se trouvent des fonctions et leurs arguments. Normalement, tous les objets constituant un programme sont évalués les uns après les autres depuis le premier et jusqu'au dernier. Cependant, des instructions dites «de structure» permettent de modifier l'ordre d'exécution.

Toutes les données manipulées (arguments des fonctions, résultats) passent par la pile. Un programme RPL respecte la notation postfixe (ou notation polonaise inversée, RPN). Les arguments d'une fonction précèdent donc presque toujours ladite fonction.

Le RPL (*Rom based Procedural Language*) est un langage de programmation dit de haut niveau. Il offre de puissantes fonctions mathématiques et dispose de toutes les structures de programmation classiques.

■ Les trois niveaux de programmation de la HP-48

Bien que le RPL soit le langage de programmation officiel de la HP-48, il est aussi possible d'utiliser sur la HP-48 d'autres langages : le RPL-système et le langage assembleur.



Le RPL assure la compatibilité ascendante des programmes entre les familles HP-28 et HP-48. C'est un langage sûr car une erreur provoquée par un programme se contente d'interrompre l'exécution du programme sans pour autant détruire le contenu de la mémoire.

Chaque instruction RPL correspond à l'exécution de plusieurs sous-instructions (appelées primitives de l'instruction). Par exemple, l'exécution de la fonction `SIN` (calcul de sinus) s'accompagne de la vérification de la présence d'un élément au niveau 1 de la pile et de sa compatibilité avec la fonction. Le RPL-système est un langage de programmation ne faisant appel qu'à ces primitives. Il est ainsi possible de n'exécuter que les primitives utiles en n'oubliant celles qui ne sont chargées que de la sécurité du programme (détection des arguments, vérification de leurs types). Le RPL-système est donc avant tout un RPL dégraissé qui permet de modeler des instructions nouvelles à partir des primitives disponibles. La programmation en RPL-système revient donc à désigner par leurs adresses les primitives utilisées. La rapidité d'exécution et la possibilité d'élaborer les instructions « à la carte » sont les deux avantages du RPL-système. La programmation en RPL-système est l'objet de la troisième partie de ce livre.

L'assembleur est un langage de bas niveau. Il est donc très proche des possibilités réelles du microprocesseur mais n'est pas du tout convivial et exige une véritable gymnastique intellectuelle de la part du programmeur. Par contre, il donne un accès direct et total à l'ensemble des possibilités de la machine. Les programmes écrits en assembleur sont très rapides mais ne sont pas sûrs. En effet, une erreur du programmeur peut provoquer la perte du contenu de la mémoire. La programmation en assembleur fait appel à de nombreuses notions de mathématiques et de logique exposées dans l'ouvrage *HP-48 : faites vos jeux en assembleur* édité par DUNOD.

2

Créer un programme



Il va maintenant être question d'objets, de pile et de variables. Si la manipulation de la pile et les notions d'objet, de variable et de répertoire courant sont pour vous floues, reportez-vous à l'ouvrage *HP-48 : permis de conduire* (DUNOD) qui a été conçu pour les débutants.

■ Les délimiteurs « et »

Un programme RPL est un objet comme un autre. Un objet de type nom, par exemple `ABC`, un objet de type nombre réel, par exemple `4.56` ou une fonction intégrée, par exemple `SIN`, ne sont pas encadrés par des délimiteurs afin de faciliter leur saisie.

Cependant, la plupart des autres objets sont placés entre des délimiteurs, ainsi :

- un caractère est placé entre ``` et ```.
- une chaîne de caractères est placée entre `"` et `"`,
- une liste d'objets est placée entre `{` et `}`,
- un vecteur est placé entre `[` et `]`,
- un entier binaire est précédé par `#` et est suivi par une lettre `b`, `o`, `d` ou `h` indiquant la base avec laquelle est exprimé l'entier binaire,
- etc.



De la même manière, un objet de type programme RPL est placé entre les délimiteurs « et ». Ceci dit, certaines structures peuvent, au sein d'un programme, être elles aussi délimitées par « et ».

■ Saisir un programme et le placer dans une variable

Un programme RPL doit être introduit comme n'importe quel autre objet au moyen de la ligne de commande.

Pour commencer la saisie d'un programme, initialisez la ligne de commande en appuyant sur `[ON]` puis saisissez les délimiteurs « et » à l'aide de `[←]` `[-]`. Le point d'insertion, symbolisé par le curseur, se place automatiquement entre les délimiteurs. Dès la saisie des délimiteurs, l'indicateur `PRG` s'affiche en haut à droite.

Au sein d'un programme les différents objets sont séparés par des espaces obtenus en appuyant sur `[SPC]`.

```

PRG
{ HOME }
2:
1:
«  ♦
»

```

GRCH TEST TYPE LIST GR08 PICT

```

PRG
{ HOME }
2:
1:
« 440 1 BEEP  ♦
»

```

BEEP

Introduisez par exemple le programme :

```
«
440 1 BEEP
»
```

Ce programme produit un signal sonore durant une seconde et dont la fréquence est 440 hertz (la du diapason). 440 et 1 sont les arguments de l'instruction BEEP. Conformément aux usages de la notation polonaise inversée, les arguments d'une fonction ou instruction précèdent cette dernière.

L'instruction BEEP peut être saisie de deux façons différentes mais parfaitement équivalentes :

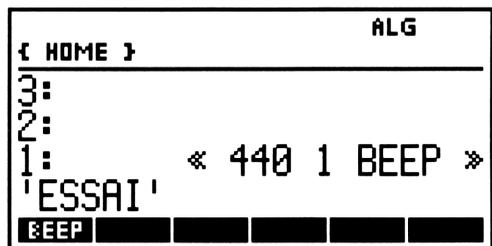
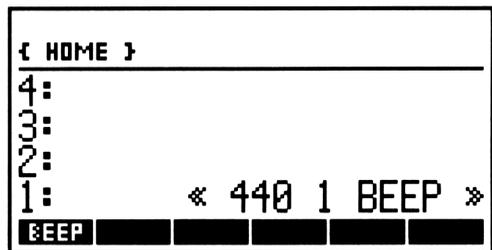
- caractère par caractère, bloquez le clavier en mode alphabétique avec [α] [α] puis introduisez [B] [E] [E] [←],
- en sélectionnant dans le menu approprié l'article correspondant à BEEP. BEEP est ainsi accessible avec [PRG] [NXT] (OUT) [NXT] (BEEP) sur HP-48 G/GX ou avec [PRG] (CTRL) [NXT] [NXT] (BEEP) sur HP-48 S/SX.

La seconde méthode doit être préférée à la première car elle est souvent plus efficace. Certaines instructions, telles que STEP sur HP-48 G/GX, sont accessibles à partir de différents articles de menu : [PRG] (BRCH) (START) (STEP) ou [PRG] (BRCH) (FOR) (STEP). L'emploi de l'un ou l'autre de ces articles n'introduit aucune différence.

Pendant la saisie du programme, les touches de déplacement et les touches d'effacement [DEL] et [←] restent disponibles et permettent la modification du programme.

Une fois le programme saisi, une pression sur [ENTER] le place dans la pile.

Il convient alors de stocker le programme dans une variable. Dans le cadre de cet exemple, plaçons le programme dans une variable que nous nommerons 'ESSAI'.



Le programme étant au niveau 1 de la pile, saisissez 'ESSAI' puis appuyez sur [STO]. Le programme se trouve maintenant dans la variable ESSAI du répertoire courant. Pour faire apparaître cette variable dans le menu horizontal, il suffit d'appuyer sur [VAR].

```
{ HOME }
4:
3:
2:
1:
ESSAI
```

■ Editer un programme

?

Editer un programme revient à le modifier alors que celui-ci se trouve déjà dans une variable. Il est ainsi possible de corriger ou d'améliorer un programme après son stockage dans une variable.

Pour modifier un programme déjà enregistré dans une variable, affichez les variables avec [VAR] puis placez-vous dans le répertoire contenant la variable correspondant au programme à éditer. La variable souhaitée étant affichée dans la barre des menus, appuyez sur ['] puis sur la touche blanche se trouvant sous l'article correspondant à la variable mémorisant le programme. Le nom de la variable étant apparu dans la ligne de commande, appuyez sur [ENTER] pour le placer au niveau 1 de la pile. Appelez ensuite l'environnement d'édition avec [↵] [+/-]. Le programme apparaît alors sur l'écran. Utilisez les touches de déplacement pour placer convenablement le point d'insertion et effectuez vos modifications. Une fois les modifications effectuées, une pression sur [ENTER]

```
{ HOME }
4:
3:
2:
1:
'ESSAI'
```

```
{ HOME } PRG
«440 1 BEEP
»
←SKIP SKIP→ ←DEL DEL→ INS ▢ ↑STK
```

```
{ HOME } PRG
« 880 1 BEEP
»
←SKIP SKIP→ ←DEL DEL→ INS ▢ ↑STK
```

provoque la mémorisation du programme modifié dans la variable où était stockée la précédente version. Inversement, une pression sur [ON] permet de quitter l'environnement d'édition sans enregistrer les modifications.



Au sein d'un programme RPL, les objets (instructions, variables, arguments...) peuvent être séparés par [SPC] (espace) ou par [↵] [.] (retour à la ligne). Plus d'un espace ou plus d'un retour à la ligne consécutifs ne provoquent pas d'erreur car, dans tous les cas, la HP-48 remodèle le listing afin de limiter la longueur des lignes. La mise en forme du listing a lieu dès sa validation par [ENTER]. Ainsi, il est normal que le programme n'apparaisse pas comme pendant sa saisie initiale à l'occasion d'une édition ultérieure.

■ Exécuter un programme

Pour exécuter un programme il suffit d'afficher le menu présentant le répertoire où se trouve la variable contenant le programme puis d'appuyer sur la touche blanche placée sous l'article désignant ladite variable.

Comme une fonction ou une instruction, un programme peut prendre des arguments dans la pile. L'absence de ces arguments ou leur inadaptation au programme peuvent provoquer une erreur (ce n'est cependant pas là la seule source d'erreur !).



Un programme RPL peut être interrompu à tout moment en appuyant sur [ON]. Cette possibilité est particulièrement utile lorsque le programme se place accidentellement dans une boucle sans fin.

■ Effacer un programme

Un programme est une variable comme une autre. Pour l'effacer il suffit donc de supprimer la variable qui le contient, pour cela :

- placez-vous dans le répertoire contenant la variable correspondant au programme,
- affichez les variables de ce répertoire à l'aide de [VAR] (et, éventuellement, de [NXT]),
- appuyez sur ['],
- appuyez sur la touche blanche se trouvant sous l'article associé à la variable où est placé le programme,
- supprimez la variable à l'aide de la fonction PURGE accessible à l'aide de [↵] [EEX].

■ Vérifiez un programme

L'instruction `BYTES` permet de vérifier qu'un programme a été convenablement saisi. `BYTES` prend en argument l'objet placé au niveau 1 de la pile puis renvoie :

- une somme de contrôle sous la forme d'un entier binaire,
- l'espace occupé en mémoire par l'objet.



Attention ! Ne confondez pas un objet avec la variable qui le contient. La somme de contrôle est exprimée sous la forme d'un entier binaire et peut donc être représentée en base 2, 8, 10 ou 16. L'usage voulant que les sommes de contrôle soient exprimées en base 16, exécutez `HEX` pour faire de la base 16 la base de numération courante. L'espace occupé est exprimé en octets. L'unité d'information de la HP-48 étant le quartet, il est normal que l'espace occupé soit multiple de 0.5 octets.

Pour afficher la somme de contrôle et la taille d'un objet, affichez le contenu du répertoire où se trouve la variable contenant l'objet à vérifier. Appuyez sur `[↵]` puis sur la touche blanche se trouvant sous l'article du menu désignant ladite variable. L'objet doit alors apparaître au niveau 1 de la pile mais n'est pas évalué. L'instruction `BYTES` est ensuite mise en œuvre en saisissant `[↵] [VAR] (BYTES)`. La somme de contrôle est renvoyée au niveau 2 alors que l'espace occupé est affiché au niveau 1.

```
{ HOME }
4:
3:
2:
1:      « 880 1 BEEP »
MEM
```

```
{ HOME }
4:
3:
2:      # FF8Eh
1:      25.5
MEM BYTES NEWD DIR ARITH
```



La somme de contrôle, aussi appelée CRC (code de redondance cyclique) est un calcul fondé sur un polynôme et effectué à partir des données transitant sur le bus. Ce calcul est extrêmement rapide car il est effectué en logique câblée par un composant spécialisé. La somme de contrôle est un moyen très fiable pour vérifier la saisie correcte d'un objet. Les sommes de contrôle sont, sur la HP-48, des entiers binaires comportant quatre chiffres hexadécimaux.



Nous fournirons désormais les sommes de contrôle hexadécimales des programmes proposés en exemples.

3

Variables utilisées par un programme

■ Variables globales

Les variables du répertoire courant sont utilisables à partir d'un programme. Par exemple :

Commencez par placez le réel 3 dans la variable 'A'.

```
3      [ENTER]
'A'    [ENTER]
       [STO]
```

Introduisez ensuite le programme :

```
<< A >>
```

Exécutez ce programme et constatez que, l'exécution d'un programme consistant en l'évaluation des objets constituant le programme, le programme évalue le nom A et place l'objet 3 associé au nom A dans la pile.

Un programme est aussi capable de créer une variable globale qui sera placée dans le répertoire courant. Pour cela, le programme place dans la pile un objet suivi d'un nom et exécute STO. Par exemple :

```
<<
3 'A' STO
>>
```

Le programme ci-dessus crée la variable A dans le répertoire courant. Celle-ci contient un réel valant 3.



Un programme est aussi capable de naviguer entre les répertoires :

- HOME fait du répertoire racine (HOME) le répertoire courant,
- UPDIR permet de remonter d'un rang dans l'arborescence (le répertoire-père devient le répertoire courant),
- CRDIR prend un nom global en argument et crée un sous-répertoire portant ce nom,
- bien entendu, l'évaluation d'un nom correspondant à un sous-répertoire fait de ce sous-répertoire le répertoire courant.



L'instruction `PATH` renvoie une liste correspondant au chemin courant. Rappelons que le chemin courant est la liste des répertoires s'étendant de la racine (`HOME`) jusqu'au répertoire courant.

Le programme ci-dessous crée le sous-répertoire `SOUREP` (`'SOUREP' CRDIR`) dans le répertoire courant, fait de ce sous-répertoire le répertoire courant (`SOUREP`) puis renvoie au niveau 1 de la pile le chemin courant (`PATH`).

```
«
'SOUREP' CRDIR
SOUREP
PATH
»
```

- CRC : #5073h / 39

Si le programme a été lancé depuis `HOME`, le chemin renvoyé au niveau 1 est { `HOME SOUREP` }. Une fois le programme exécuté, le répertoire courant est `SOUREP`. Pour que le répertoire courant à l'issue de l'exécution soit le répertoire courant au moment du lancement de l'exécution, il suffit d'ajouter `UPDIR` à la fin du programme. On obtient :

```
«
'SOUREP' CRDIR
SOUREP
PATH
UPDIR
»
```

- CRC : #DF70h / 41.5

■ Les variables réservées



Les variables réservées sont, elles aussi, accessibles depuis un programme et peuvent être manipulées comme n'importe quelle autre variable globale. Il existe même des instructions facilitant leur modification. Par exemple, l'instruction `STEQ` remplace la séquence `'EQ' STO`.

■ Variables locales et fonctions-utilisateur

Bien que les variables globales soient d'un emploi commode, elles présentent l'inconvénient de demeurer dans le répertoire courant après l'exé-

cution du programme. La conservation de données intermédiaires étant généralement inutile, l'emploi de variables locales dont la durée de vie est limitée à une portion de programme présente de grands intérêts.

Deux structures sont possibles pour mettre en œuvre des variables locales. La première fait intervenir une expression algébrique et se compose de trois éléments :

- ❶ → marque le début de la structure où sont utilisées les variables locales,
- ❷ les noms des variables locales, toujours exprimés à l'aide de lettres minuscules,
- ❸ une expression algébrique où interviennent les noms des variables locales,

Cette première structure permet d'appliquer simplement une formule algébrique. La seconde structure possible est d'un emploi nettement plus étendu puisque la mise en œuvre des variables locales est possible à l'intérieur d'une portion du programme délimitée par « et ». Notez que cette portion de programme peut, elle aussi, contenir une autre structure imbriquée faisant appel à des variables « encore plus » locales.

Emploi des variables locales dans une portion de programme :

- ❶ → marque le début de la structure où sont utilisées les variables locales,
- ❷ ici aussi, les noms des variables locales, toujours exprimés à l'aide de lettres minuscules,
- ❸ la portion de programme où sont utilisées les variables locales est délimitée par « et ».



L'emploi de lettres minuscules pour désigner les variables locales n'est pas absolument obligatoire mais est préférable afin d'éviter une confusion avec les variables globales homonymes. Rappelons que les minuscules sont obtenues avec $[\alpha]$ [\leftarrow] et que \rightarrow l'est avec $[\rightarrow]$ [0].

```
«  
→ a b  
'a+b'  
»
```

• CRC : #3F52h / 38

Le programme ci-dessus prend deux objets dans la pile. L'objet placé au niveau 2 est associé à la variable locale **a** alors que l'objet du niveau 1 est associé à la variable locale **b**. L'expression algébrique **'a+b'** est ensuite

immédiatement évaluée et le résultat de son évaluation est renvoyé au niveau 1 de la pile.

```

«
→ A B X Y
«
  '-X+2*A'
→NUM
  '-Y+2*B'
→NUM
»
»

```

- CRC : #4E71h / 88.5

Le programme ci-dessus est tiré du livre *HP-38/HP-48 pour le bac* (DUNOD). Il emploie une structure de variables locales afin de calculer l'image d'un point du plan par une symétrie centrale. Le programme prend dans la pile :

- au niveau 4 de la pile, l'abscisse du centre de la symétrie. Cette abscisse est copiée dans A,
- au niveau 3 de la pile, l'ordonnée du centre de la symétrie. Cette ordonnée est copiée dans B,
- au niveau 2 de la pile, l'abscisse du point transformé par la symétrie. Cette abscisse est copiée dans X,
- au niveau 1 de la pile, l'ordonnée du point transformé par la symétrie. Cette ordonnée est copiée dans Y.



La portion du programme où sont utilisées les variables locales est délimitée par « et ». Ici, cette portion de programme comprend deux expressions algébriques dont nous forçons l'évaluation numérique à l'aide de →NUM. Finalement, le programme renvoie au niveau 2 l'abscisse de l'image du point et son ordonnée au niveau 1.

Par exemple, utilisons ce programme pour calculer les coordonnées de P', image du point P(-1,-2) par la symétrie de centre Ω(1,2). Pour cela, on introduit dans la pile les coordonnées de Ω puis celles de P :

```

1 [ENTER]
2 [ENTER]
-1 [ENTER]
-2 [ENTER]

```

Exécutez le programme, celui-ci renvoie 3 au niveau 2 et 6 au niveau 1, donc P' est de coordonnées (3,6).

■ Portée d'une variable

Les variables locales présentent l'avantage d'avoir une portée (aussi appelée visibilité) limitée à l'expression algébrique ou à la portion de programme pour laquelle elles sont définies.



Les variables globales, elles, existent indépendamment du programme et leur création ou leur destruction ne sont pas automatiques. Notez qu'un nom de variable globale ne suffit pas pour désigner celle-ci sans ambiguïté car plusieurs variables globales peuvent porter le même nom à condition de ne pas se trouver dans le même répertoire. La vérification de l'identité du répertoire courant est donc importante.

Structures RPL



Les objets composant un programme sont évalués les uns après les autres depuis le début du programme et jusqu'à la fin de celui-ci. Cette exécution linéaire est dite séquentielle. Cependant, on peut répéter l'exécution d'une partie d'un programme en insérant cette partie au sein d'une boucle. Par ailleurs, il est possible de n'exécuter une instruction ou une portion de programme que si une condition est satisfaite. Boucles et tests constituent les structures d'un programme.

1

Boucles

■ Boucle à compteur *FOR... NEXT...*



La structure de boucle *For... Next...* est l'une des plus courantes. Les instructions placées dans la boucle font l'objet d'un nombre fixe de répétition.

Une boucle *For... Next...* fait intervenir un compteur, généralement noté *i*. Ce compteur est une variable locale propre à la boucle. Sa valeur varie entre une valeur minimale et une valeur maximale. Après chaque exécution de la boucle le compteur est incrémenté (on lui ajoute 1). Il est possible d'utiliser le compteur comme argument des instructions constituant le corps de la boucle.

```

«
1 5 ← Valeurs minimale et maximale du compteur
FOR
i ← Nom du compteur
┌───┐
i 220 *
.2
BEEP
└───┘ Instructions formant le corps de la boucle
NEXT ← Fin de la boucle
»
    
```

- CRC : #23C8h / 55

Le programme ci-dessus comporte une boucle For... Next répétée cinq fois. Cette boucle se compose de :

- la valeur minimale du compteur (1),
- la valeur maximale du compteur (5),
- le nom du compteur (i) se place toujours immédiatement après FOR,
- à la suite du nom du compteur se placent les instructions formant le corps de la boucle. Ce sont ces instructions qui seront répétées,
- la fin de la boucle est signalée par NEXT.

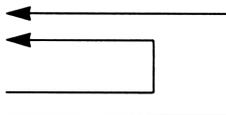
1 5 FOR i... fait varier i entre 1 et 5. Rappelons que i est une variable locale employée comme compteur. Lors de la première exécution de la boucle i vaut 1. Lors des exécutions suivantes, i vaut 2, 3, 4 puis 5. NEXT provoque automatiquement l'incrémentement du compteur. La valeur maximale du compteur (ici, 5) limite le nombre d'exécutions du contenu de la boucle. Une boucle For... Next est donc une boucle finie.

Remarquez que la déclaration de la variable locale employée comme compteur est implicite. Le compteur étant une variable locale, son emploi est limité à la boucle.

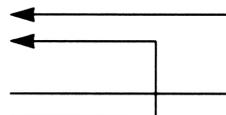
■ Imbrication de boucles



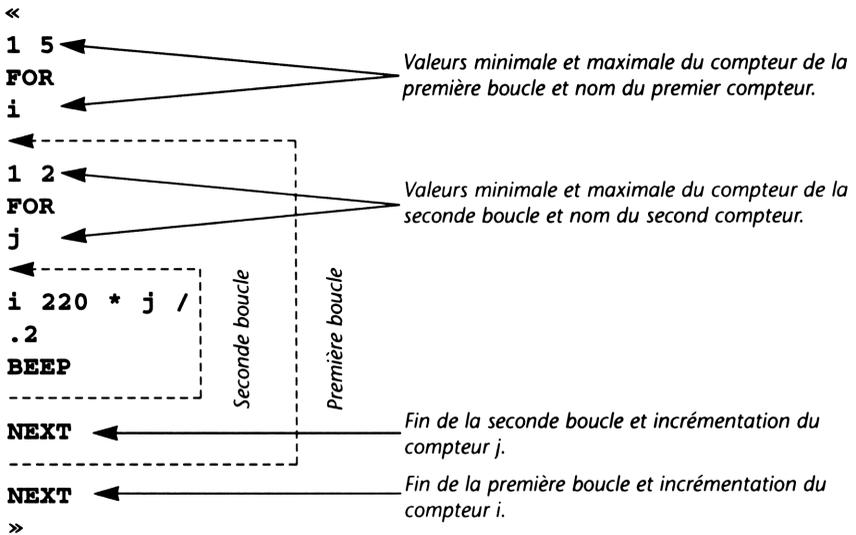
Les boucles peuvent contenir d'autres boucles. Elles peuvent être imbriquées les unes dans les autres mais ne peuvent pas se chevaucher.



OUI



NON



- CRC : #C672h / 76.5

Dans le cadre de cet exemple, la première boucle est exécutée cinq fois (*i* varie entre 1 et 5). A chaque exécution de la première boucle, la seconde boucle est exécutée deux fois (*j* varie entre 1 et 2).



Le compteur *i* est une variable locale. Sa portée se limite à la boucle qui l'utilise. Ici, la seconde boucle est intégrée à la première. Le compteur *i* est donc visible (utilisable) dans le corps de la seconde boucle puisque le corps de la seconde boucle est un sous-ensemble de la première. Par contre, le compteur *j* n'est visible que par le corps de la seconde boucle.

■ Boucle à compteur **FOR... STEP...**



Les boucles **FOR... NEXT** que nous venons de voir permettent aux instructions du corps de la boucle d'exploiter la valeur prise par le compteur. Dans le cadre des boucles **FOR... NEXT**, le compteur varie toujours entre deux valeurs entières et sa valeur augmente d'une unité après chaque exécution de la boucle.

Les boucles **FOR... STEP** constituent une extension du principe des boucles **FOR... NEXT**. En effet, le compteur utilisé par une boucle **FOR... STEP** peut augmenter d'une valeur quelconque après chaque exécution de la boucle. Le compteur, utilisable ici aussi par le corps de la boucle, peut ainsi prendre des valeurs réelles quelconques.

```

«
.2 .5 ← Intervalle sur lequel varie le compteur
FOR
i
  ←
  220 i /
  i
  BEEP
  ←
.1 ← Incrément
STEP
»

```

Corps de la boucle

- CRC : #89D9h / 75.5

Ci-dessus, le compteur *i* varie entre 0,2 et 0,5 et augmente de 0,1 après chaque exécution de la boucle.

■ Décrémentement du compteur



Permettant d'affecter une valeur autre que 1 à l'incrément, les boucles FOR... STEP autorisent une décrémentement du compteur (la valeur du compteur diminue après chaque exécution de la boucle).

```

«
.5 .2 ← Valeurs entre lesquelles varie le compteur
FOR
i
  ←
  220 i /
  i
  BEEP
  ←
-.1 ← Valeur dont le compteur est diminué après
STEP
      chaque exécution de la boucle.
»

```

Corps de la boucle

- CRC : #E0AEh / 75.5

Ci dessus, la première valeur du compteur (.5) est supérieure à sa dernière valeur (.2), par ailleurs, l'incrément (-.1) est négatif. Le compteur *i* varie donc entre 0,5 et 0,2 et vaut successivement 0,5, 0,4, 0,3, 0,2 puis 0,1.

■ Boucle à compteur *START... NEXT...*



Les boucles *FOR... NEXT* et *FOR... STEP* permettent d'exploiter un compteur dans le corps de la boucle. Les boucles *START... NEXT* sont aussi des boucles à compteur mais le programmeur ne peut ni donner un nom au compteur (qui est une variable locale) ni exploiter la valeur du compteur dans le corps de la boucle.

```

«
1 'A' STO ← Création de la variable A contenant le réel 1

1 3 START ← Le compteur varie entre 1 et 3
┌───┐
│ 'A' INCR │ Corps de la boucle
└───┘

NEXT
»

```

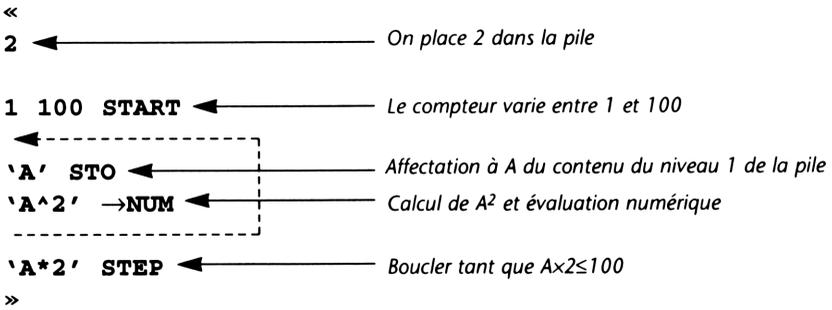
- CRC : #12D1h / 46.5

Le programme ci-dessus crée la variable globale *A* dans le répertoire courant et lui affecte la valeur 1 (*A* contient donc un objet de type réel). Le compteur de la boucle varie entre 1 et 3. Le corps de la boucle sera donc exécuté trois fois, le compteur valant successivement 1 puis 2 puis 3. Le corps de la boucle comprend ici la séquence *'A' INCR* qui incrémente (augmente par pas de 1) le contenu de la variable *A*. La variable *A* valant 1 au départ et étant incrémentée trois fois, elle vaut 4 à la l'issue de l'exécution du programme. Après chaque incrémentation la valeur de *A* a été placée dans la pile. La pile contient donc finalement 2, 3 et 4.

Les structures de boucles (finies et à compteur) *START... NEXT* et *START... STEP* sont à utiliser lorsque la valeur du compteur ne doit pas être exploitée par le corps de la boucle.

■ Boucle à compteur *START... STEP...*

Il existe entre les boucles *START... NEXT* et les boucles *START... STEP* la même différence qu'entre les boucles *FOR... NEXT* et les boucles *FOR... STEP*. Autrement dit, les boucles *START... STEP* permettent de préciser la valeur dont est incrémenté le compteur. Nous avons déjà vu qu'il est possible d'utiliser un compteur négatif (voir l'exemple fourni pour *FOR... STEP*). Comme nous allons maintenant le constater, une expression algébrique (dont l'évaluation est ici automatique) permet aussi de déterminer la valeur de l'incrément.



- CRC : #846Eh / 74

Ici, le compteur est incrémenté de la valeur correspondant à l'évaluation numérique de 'A*2'. Initialement le compteur vaut 1. La boucle est répétée tant que le compteur est inférieur ou égal à 100. pour chaque exécution de la boucle le tableau ci-dessous fournit la valeur de A et celle du compteur.

| Exécution | A | A ² | A×2 | Valeur compteur |
|-----------|-----|----------------|-----|-----------------|
| 1ère | 2 | 4 | 4 | 1+4 = 5 |
| 2ème | 4 | 16 | 8 | 5+8 = 13 |
| 3ème | 16 | 256 | 32 | 13+32 = 45 |
| 4ème | 256 | 65536 | 512 | 45+512 = 557 |

Initialement, 2 est placé dans A. Ensuite, le carré de A remplace la précédente valeur de A. A l'issue de la quatrième exécution de la boucle, le compteur vaut 557, la répétition du contenu de la boucle cesse donc. Ici, la boucle est donc répétée quatre fois. 65536 se trouve au niveau 1 de la pile et A contient 256 à l'issue de ces quatre exécutions de la boucle.

■ Boucle avec test de sortie en début de boucle **WHILE... REPEAT... END**

La structure de boucle **WHILE... REPEAT... END**, aussi appelée boucle « tant que », n'est pas munie d'un compteur. Le nombre d'exécutions de la boucle n'est donc pas déterminé par avance. Une condition placée au début de la boucle permet, si la condition est remplie, d'exécuter une première fois les instructions formant le corps de la boucle. Le corps de la boucle est ensuite exécuté tant que la condition est satisfaite.



Les structures **FOR... STEP, FOR... NEXT, START... NEXT** et **START... STEP**, imposent au moins une exécution du corps de la boucle. Ce n'est pas le cas avec la structure **WHILE... REPEAT... END**.

```

«
WHILE
.9 RAND > ←
REPEAT
  ←
  RAND 1000 *
  .5
  BEEP
  ←
END
»

```

Entrer dans la boucle si 0,9 est supérieur à un nombre aléatoire puis continuer à boucler tant que cette condition est satisfaite.

Ici le corps de la boucle provoque l'émission d'un son durant 0,5 seconde et dont la fréquence est aléatoire (elle varie entre 0 Hz et 1000 Hz)

- CRC : #2B48h / 66.5

■ Boucle avec test de sortie en fin de boucle **DO... UNTIL... END**

Une boucle **DO... UNTIL... END** est toujours exécutée au moins une fois. Le corps de la boucle est ensuite répété tant que la condition (placée à la fin de la boucle) n'est pas satisfaite.



Une boucle **WHILE... REPEAT... END** est répétée **tant que** la condition est satisfaite alors qu'une boucle **DO... UNTIL... END** est exécutée **jusqu'à ce que** la condition soit satisfaite.

```

«
DO ←
  ←
  RAND 1000 *
  .5
  BEEP
  ←
UNTIL
.9 RAND < ←
END
»

```

En-tête de la boucle

Corps de la boucle

Répéter la boucle jusqu'à ce que la condition soit satisfaite

- CRC : B6DEh / 61.5



Le programme ci-dessus est très proche du programme illustrant la boucle **WHILE... REPEAT... END** mais ici, le corps de la boucle est exécuté au moins une fois et la condition testée est à opposer à celle utilisée pour l'autre exemple. Ici la condition permet de sortir de la boucle alors qu'avec **WHILE... REPEAT... END** la condition permet d'entrer dans la boucle puis de s'y maintenir.

■ A propos des conditions...



Les conditions utilisées pour illustrer les boucles WHILE... REPEAT... END et DO... UNTIL... END sont en fait des tests faisant intervenir les symboles $<$, $>$, $==$ (égalité), \leq , \geq ou \neq .

La notation polonaise inversée est aussi appliquée aux tests. Ainsi, un test formulé algébriquement sous la forme $a < b$ devient $a b <$ en RPL. Si a est effectivement inférieur à b , la condition est remplie.

Quelques exemples de conditions :

A 1 ==

La condition est remplie si la variable globale A contient le réel 1.

AB "BONJOUR" ==

La condition est remplie si la variable globale AB contient la chaîne de caractères "BONJOUR".

A 1 == AB "BONJOUR" == and

La condition est remplie si la variable globale A contient le réel 1 *et* si la variable globale AB contient la chaîne de caractères "BONJOUR".

A 1 == AB "BONJOUR" == or

La condition est remplie si la variable globale A contient le réel 1 *ou* si la variable globale AB contient la chaîne de caractères "BONJOUR".



Vous pouvez utiliser `and`, `or` et `xor` pour combiner les conditions :

- avec `and`, les différentes conditions doivent être satisfaites simultanément,
- avec `or`, l'une ou l'autre des conditions suffit,
- avec `xor`, il faut qu'une et une seule condition soit satisfaite (la condition n'est globalement pas remplie si les deux conditions liées par `or` sont remplies simultanément).

Il est possible de combiner différentes conditions élémentaires :

X 10 > X 20 < AND Y 15 > Y 35 < AND OR

... mais les tests peuvent aussi profiter de la notation algébrique...

'X>10' 'X<20' AND 'Y>15' 'Y<35' AND OR

...d'ailleurs, toute la «phrase» peut profiter de la notation algébrique :

'X>10 and X<20 or Y>15 and Y<35'

Il est donc possible d'utiliser des conditions complexes.

2

Tests



Les tests permettent d'exécuter une portion de programme si une condition donnée est remplie.

■ Structure *si... alors...*



La structure IF... THEN... END est la structure de test la plus évidente. Si la condition placée entre IF et THEN est satisfaite alors les instructions placées entre THEN et END sont exécutées.

```

«
→ i ←————— Le contenu du niveau 1 de la pile est placé dans i
  «
  IF
  i 2 > ←————— Ici, i>2 est la condition à satisfaire
  THEN
  ┌───┐
  │ i │
  │ DUP │
  │ + │
  └───┘
  ←───┘
  END
  » ←————— Fin de la visibilité de i
»

```

Cette portion du programme n'est exécutée que si la condition est satisfaite. Le contenu de i est placé dans la pile, puis dupliqué et ajouté à lui-même.

- CRC : #FBA3h / 53.5

Avant de lancer le programme ci-dessus, placez un réel dans la pile, tapez par exemple 8 [ENTER]. Ce réel est affecté à la variable locale **i**. Si **i** est supérieur à 2 (**i 2 >**), la portion de programme **i DUP +** est exécutée, sinon, elle est ignorée.

■ Structure *si... alors... sur la pile*

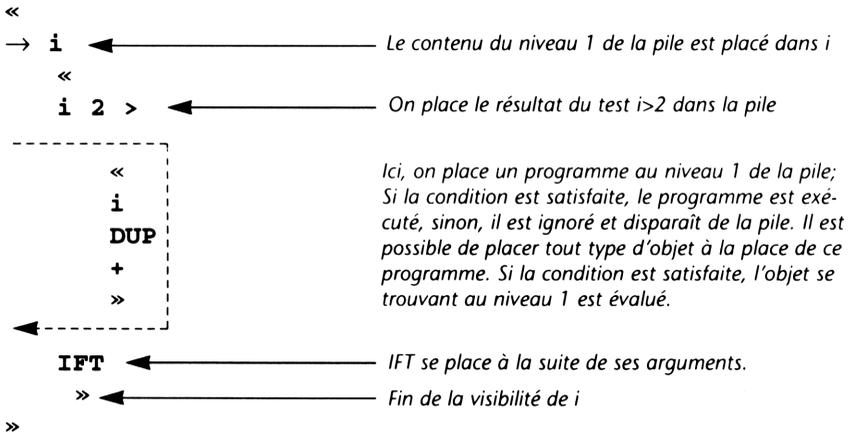


La structure IF... THEN... END a l'avantage d'être très lisible mais demeure relativement lourde. L'instruction **IFT** joue exactement le même rôle que IF... THEN... END mais est beaucoup plus légère.



Pour utiliser `IFT` il faut que le résultat d'un test se trouve au niveau 2 et que l'objet à évaluer, si la condition testée est satisfaite, se trouve au niveau 1 de la pile.

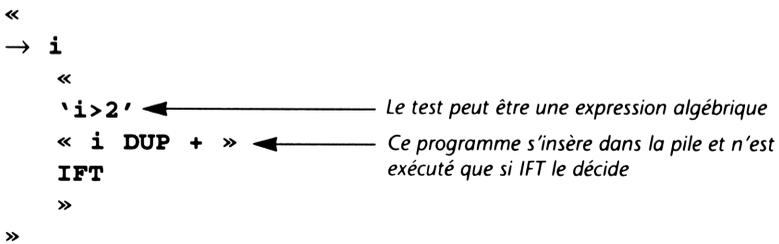
Le programme ci-dessous est absolument équivalent au précédent (placez un réel dans la pile avant de l'exécuter). On effectue d'abord un test puis on insère dans la pile l'objet à évaluer si la condition est satisfaite.



• CRC: #37FAh / 56



Ce programme peut aussi être présenté sous une forme plus compacte mettant en évidence l'insertion dans la pile des deux arguments de `IFT` :



• CRC: #2115h / 61

■ Structure *si... alors... sinon...*

Avant et après une structure de test les instructions sont exécutées sans condition. Avec l'instruction `IFT` ou une structure `IF... THEN... END` une portion du programme n'est exécutée que si une condition est remplie.

La structure IF... THEN... ELSE... END est une extension de la structure IF... THEN... END. Avec IF... THEN... ELSE... END une portion du programme ne sera exécutée que si une condition est remplie alors qu'une autre portion ne sera exécutée que si cette condition n'est pas remplie.

```

«
→ i
  « ← Début de la visibilité de la variable locale i
  IF
  'i>2' ← La condition est ici une expression algébrique

  THEN
  " >2 " ← Les instructions à exécuter si la condition est remplie
  sont placées entre THEN et ELSE
  ELSE
  " ≤2 " ← Les instructions à exécuter si la condition n'est pas
  remplie sont placées entre ELSE et END
  END
  » ← Fin de la visibilité de la variable locale i
»

```

- CRC #DC3Ah / 60,5

Le programme ci-dessus exploite une structure de variable locale. Placez un réel dans la pile avant de lancer le programme. Le programme place dans la pile la chaîne ">2" si le réel est supérieur à 2 ou renvoie "≤2" si le réel est inférieur ou égal à 2.

■ Structure si... alors... sinon... sur la pile



Nous savons déjà que IFT est une forme allégée mais moins lisible de IF... THEN... END. De même il existe pour IF... THEN... ELSE... END une forme allégée. L'instruction IFTE correspond à une structure IF... THEN... ELSE... END mais travaille à partir de la pile.

IFTE prend trois arguments :

- au niveau 3, un test (condition à remplir),
- au niveau 2, les instructions à exécuter si la condition est satisfaite,
- au niveau 1, les instructions à exécuter si la condition n'est pas satisfaite.

```

«
→ i ← Déclaration de la variable locale i
«
  'i>2' →NUM ← Le test peut être une expression algébrique
  ">2" ← Objet évalué si la condition est remplie
  "≤2" ← Objet évalué si la condition n'est pas remplie
  IFTE ← IFTE se place à la suite de ses trois arguments
»
»

```

- CRC : 4D95h / 55.5

Le programme ci-dessus fonctionne exactement comme le précédent. Il prend en argument un réel placé dans la pile avant de lancer le programme.



Les tests utilisés par IFT et IFTE doivent renvoyer un nombre différent de 0 si la condition est remplie ou 0 si elle ne l'est pas. Ci-dessus on force l'évaluation numérique de 'i>2' avec →NUM.

■ Imbrication de tests



Comme les boucles, les tests peuvent s'imbriquer les uns dans les autres. Ci-dessous, un test n'est effectué que si une première condition est satisfaite.

```

«
→ i
«
  IF 'i>2' ← Première condition
  THEN
    IF 'i>10'
    THEN
      "TRES GRAND"
    ELSE
      "GRAND"
    END
  ELSE
    "PETIT" ← Chaîne renvoyée si 'i>2' n'est pas vérifié
  END
»
»
»

```

Ce second test (il s'agit ici d'une structure IF... THEN... ELSE... END) n'est effectué que si la première condition est satisfaite ('i>2')

- CRC : #E20Ah / 119

Le programme ci-avant prend en argument un réel placé dans la pile avant le lancement du programme :

- si ce réel est supérieur à 2 et supérieur à 10, le programme renvoie la chaîne "TRES GRAND",
- si ce réel est supérieur à 2 mais inférieur ou égal à 10, le programme renvoie la chaîne "GRAND",
- si ce réel est inférieur ou égal à 2, le programme renvoie la chaîne "PETIT".



Il est très facile d'imbriquer des boucles, des tests ou des structures de variables locales les uns dans les autres. Néanmoins, le programmeur doit faire très attention aux positions des délimiteurs « et » ainsi qu'aux emplacements des instructions END utilisées pour marquer les fins des tests ou des boucles.

Adaptons maintenant le programme précédent aux spécificités de l'instruction IFTE :

```

«
→ i
«
  'i>2' →NUM
  «-----»
  'i>10' →NUM
  "TRES GRAND"
  "GRAND"
  IFTE
  »←-----»
  "PETIT"
  IFTE
»
»

```

Ce programme, délimité par « et » est placé dans la pile. Il n'est évalué que si $i > 2$. Lors de son évaluation, il procède à un autre test qui fait, lui aussi, intervenir une instruction IFTE.

- CRC : #1083h / 116.5

Le programme ci-dessus fonctionne exactement comme le précédent. Il prend en argument un réel placé dans la pile avant de lancer l'exécution du programme et renvoie les mêmes chaînes en résultat.

■ Batterie de tests

La structure CASE... (THEN... END...) _n END... permet d'effectuer toute une série de tests. CASE est suivi par une série de conditions à remplir, elles-mêmes suivies par les instructions à exécuter si ces conditions sont satisfaites.

L'instruction CASE ci-dessous est suivie par deux conditions (**i 10 >** puis **i 2 >**). A chacune de ces conditions correspond une portion de programme à exécuter si la condition est satisfaite.

```

«
→ i
«
CASE
  i 10 > ← Première condition
  THEN
    "TRES GRAND" ← Chaîne renvoyée si la première condition est satisfaite
  END
  i 2 > ← Deuxième condition
  THEN
    "GRAND" ← Chaîne renvoyée si la seconde condition est satisfaite
  END
  "PETIT" ← Chaîne renvoyée si aucune condition n'est satisfaite
END
»
»

```

- CRC : 3CBFh / 104

Ce programme prend en argument un réel placé dans la pile avant le lancement du programme. Il renvoie une chaîne de caractères commentant la valeur du réel donné en argument. Dans le cadre de cet exemple, le programme ne comporte que deux tests mais il est possible d'insérer autant de tests que vous le souhaitez à la suite de CASE. L'instruction à exécuter, si aucune condition n'a été satisfaite, est facultative.

Dans le cadre d'une structure CASE... (THEN... END...)_n END... la portion de programme exécutée de façon conditionnelle (ou la portion par défaut) peut être un véritable programme placé entre « et ». La structure CASE... (THEN... END...)_n END... autorise aussi l'imbrication d'autres tests. Ici encore la position des instructions END délimitant les tests est très importante.



Attention ! L'ordre des tests est ici très important ! Les instructions associées à la première condition satisfaite sont exécutées. Les autres sont ignorées même si la condition peut être satisfaite.

Par rapport au programme précédent, l'ordre des deux tests a été inversé dans le programme ci-dessous. Tous les cas qui devraient être interceptés par le second test sont en fait interceptés par le premier test. Les instructions associées au second test ne sont donc jamais exécutées.

```

«
→ i
«
CASE
  i 2 >
  THEN
    "GRAND"
  END
  i 10 >
  THEN
    "TRES GRAND"
  END
  "PETIT"
END
»
»

```

La condition $i 10 >$ peut être satisfaite mais le premier test intercepte prématurément tous les cas qui pourraient provoquer une exécution des instructions ci-contre. La chaîne "TRES GRAND" n'est donc jamais renvoyée.

Cette portion de programme, exécutée si aucune condition n'a été satisfaite, n'est pas obligatoire car une structure CASE n'est pas obligée de provoquer l'exécution d'une séquence d'instructions si aucune condition n'est satisfaite.

• CRC : 373h / 104

3

Gérer les erreurs



Une division par zéro, l'absence des arguments requis, l'incompatibilité des arguments ou le dépassement des capacités de la calculatrice sont des causes d'erreurs courantes. Il est possible de déclencher une erreur ou de l'intercepter. Les erreurs prévues par le système sont désignées par un entier binaire. Par exemple, l'erreur #201h correspond au message *Too few argument* renvoyé lorsque la pile ne contient pas assez d'arguments pour l'instruction en cours d'évaluation.

■ Rappel de la dernière erreur

Pour déclencher volontairement une erreur, introduisez :

```
"A"      [ENTER]
5        [ENTER]
*
```

La calculatrice renvoie ** Error : Bad Argument type*.



L'instruction `ERRM` renvoie le dernier message d'erreur. Dans le cadre de cet exemple, l'évaluation de `ERRM` renvoie donc la chaîne "Bad Argument type". L'instruction `ERRN` renvoie le numéro du dernier message d'erreur sous la forme d'un entier binaire. Toujours dans le cadre de cet exemple, l'évaluation de `ERRN` renvoie `#202h`. L'instruction `ERR0` supprime le message associé à la dernière erreur ainsi que son numéro afin d'interdire leurs rappels à l'aide de `ERRM` et de `ERRN`.

■ Déclencher une erreur

L'instruction `DOERR` déclenche une erreur et interrompt l'exécution du programme. `DOERR` peut prendre deux types d'arguments :

- une chaîne de caractères qui sera renvoyée sur la première ligne de l'écran,
- le numéro d'une erreur exprimé à l'aide d'un entier binaire.

```
<<
→ i
<<
IF
i 0 == ←————— Condition à vérifier (i=0)
THEN

  "DIVISION PAR 0" ←———— Ce message est affiché en haut de l'écran si i=0
  DOERR

ELSE
  1 i / ←————— Calcul de 1/i si i≠0
END
"FIN"
>>
>>
```

- CRC : `#D338h / 90.5`

Le programme ci-avant prend en argument un réel placé dans la pile avant de lancer le programme. Si ce réel est nul, le message placé dans la chaîne de caractères qui précède DOERR est affiché en haut de l'écran puis l'exécution du programme prend fin. Si le réel donné en argument n'est pas nul, la calculatrice renvoie l'inverse de ce réel suivi de "FIN".

Le programme ci-après fonctionne exactement comme le précédent. Cependant, lorsque l'argument est nul, le message n°#305h (autrement dit, «*Infinite Result*») est affiché en haut de l'écran. L'exécution du programme prend fin immédiatement après l'exécution de DOERR.

```

«
→ i
«
IF
i 0 ==
THEN
#305h
DOERR

ELSE
1 i /
END
"FIN"
»
»

```

Ici, DOERR prend en argument un entier binaire. Celui-ci correspond à un message d'erreur qui sera affiché en haut de l'écran.

- CRC : #849Bh /84.5

■ Détection d'une erreur



Le déclenchement volontaire d'une erreur à l'aide de DOERR ne présente de réel intérêt que dans quelques cas particuliers et sert surtout à précipiter l'interruption de l'exécution en cas de problème. Par contre, les structures d'interception et de gestion d'erreur permettent le déclenchement d'une erreur et la poursuite de l'exécution du programme. Par ailleurs, il est possible de gérer le déroulement du programme en fonction du déclenchement d'une erreur.

Les structures d'interception d'erreur sont :

- IFERR... THEN... END (apparenté à IF... THEN... END),
- IFERR... THEN... ELSE... END (apparenté à IF... THEN... ELSE... END).

Dans le cadre de ces deux structures, il faut placer entre `IFERR` et `THEN` la portion de programme susceptible de déclencher une erreur. Si cette portion de programme déclenche une erreur, les instructions placées entre `THEN` et `END` (structure `IFERR... THEN... END`) ou celles se trouvant entre `THEN` et `ELSE` (structure `IFERR... THEN... ELSE... END`) sont exécutées.

Uniquement dans le cadre d'une structure `IFERR... THEN... ELSE... END`, les instructions placées entre `ELSE` et `END` ne sont exécutées que si aucune erreur n'a été provoquée par l'exécution de la portion de programme placée entre `IFERR` et `THEN`.

Quelle que soit la structure d'interception d'erreur, l'exécution des instructions placées à la suite du `END` délimitant la structure d'interception sont exécutées normalement, et ce, qu'une erreur ait été détectée ou pas.

```
«
→ i
«
IFERR
```

```
  1 i /
←
```

Portion de programme susceptible de déclencher une erreur

```
THEN
```

```
  "DIVISION PAR 0"
←
```

Portion de programme exécutée si une erreur a été déclenchée par la portion de programme placée entre THEN et END

```
END
»
»
```

- CRC : DCBFh / 63

Ce programme prend en argument un réel placé dans la pile avant de lancer l'exécution du programme. Si ce réel est non nul, ce programme renvoie l'inverse de ce réel.



Attention ! Les objets renvoyés dans la pile par ce programme dépendent du déclenchement ou non d'une erreur par `1 i /`. Si aucune erreur n'est déclenchée par `1 i /`, la calculatrice calcule $1/i$ et renvoie ce réel dans la pile. Si `1 i /` provoque une erreur, cette erreur ne peut être liée qu'à `/`. La calculatrice place donc `1` puis la valeur associée à `i` dans la pile mais ne peut pas calculer la division. Cette erreur provoque l'affichage du message **"DIVISION PAR 0"**. En cas d'erreur la pile contient donc `1`, la valeur de `i` et la chaîne **"DIVISION PAR 0"**.

Le programme ci-dessous est une extension du précédent et fonctionne de la même manière en prenant un réel placé dans la pile comme argument. Si aucune erreur n'a été détectée, la portion de programme placée entre `ELSE` et `END` est évaluée. Les instructions qui s'y trouvent provoquent l'insertion de la chaîne "**INVERSE =**" dans la pile et l'inversion des niveaux 1 et 2 de la pile. Par exemple, si le réel donné en argument est 2, le programme renvoie "**INVERSE =**" au niveau 2 de la pile et `.5` au niveau 1 de la pile.

```
«
→ i
«
```

```
IFERR
```

```
1 i /
```

```
←
THEN
```

```
"DIVISION PAR 0"
```

```
←
ELSE
```

```
"INVERSE ="
SWAP
```

```
←
END
»
»
```

Portion de programme susceptible de déclencher une erreur

Portion de programme exécutée si une erreur a été déclenchée par la portion de programme placée entre `THEN` et `END`

Portion de programme exécutée si la portion de programme placée entre `THEN` et `END` n'a pas déclenché d'erreur

4

Sous-programmes



Un programme est stocké en mémoire sous la forme d'une variable, et ce, comme n'importe quel autre objet. Placer un nom de variable ou un nom de fonction intégrée dans un programme provoque l'évaluation de cette variable ou l'exécution de la fonction. De la même manière, placer un nom de programme dans un programme provoque l'exécution du programme dont le nom est cité.

Saisissez le programme ci-dessous et enregistrez-le sous le nom 'BIP'.

```

«
→ a b
  «
  a b BEEP
  »
»

```

Le sous-programme BIP crée deux variables locales et prend ainsi les niveaux 1 et 2 de la pile comme arguments de l'instruction BEEP.

- CRC : #517h / 38

Dans le répertoire où se trouve 'BIP', placez également le programme ci-dessous que vous enregistrerez sous le nom 'BOUCLE'.

```

«
1 10 START
  ┌───┐
  │ RAND 1000 *
  │ RAND 3 /
  │ BIP ←
  └───┘
NEXT
»

```

Il suffit de placer le nom d'un programme dans le listing pour provoquer l'exécution de ce programme

- CRC : #3DC1h / 57.5

Lancez l'exécution de 'BOUCLE'. Ce programme appelle 'BIP' qui est traité comme un sous-programme. Après chaque appel de 'BIP', l'exécution de 'BOUCLE' immédiatement après **BIP**.



Tout programme est susceptible d'être appelé par un autre programme. Il devient alors un sous-programme.

Entrées et sorties



Un programme doit communiquer avec son utilisateur. En effet, ce dernier doit pouvoir saisir des données et disposer des résultats. L'ensemble des moyens permettant au programme de communiquer avec son utilisateur est regroupé sous le nom d'interface-utilisateur.

La HP-48 offre plusieurs instructions permettant la saisie de données ainsi qu'un choix tout aussi étendu pour les instructions chargées de restituer des résultats.

1

Entrées

■ Affichage dans la zone d'état et saisie dans la pile

L'instruction `PROMPT` prend une chaîne de caractères en argument et l'affiche en haut de l'écran (dans la zone d'état). `PROMPT` interrompt l'exécution du programme et laisse l'utilisateur introduire les objets de son choix dans la pile (vous pouvez alors introduire autant d'objets que vous le souhaitez, chaque objet doit bien entendu être validé avec `[ENTER]`). Une fois les objets introduits, il est indispensable de saisir `[←] [ON]` (fonction `CONT`) pour que le programme poursuive normalement son exécution.

```

«
"SAISIR l et L" ← Cette chaîne est affichée par PROMPT en haut de
PROMPT ← l'écran

→ i j
  «
    i j * ← Les données introduites dans la pile sont
  » ← exploitées par une structure de variables locales
»

```

- CRC : B9A1h / 58.5

Le programme ci-dessus calcule la surface d'un rectangle. PROMPT affiche un message et interrompt l'exécution pour permettre la saisie des données dans la pile. L'utilisateur peut alors introduire, par exemple, [4] [ENTER] [5] [ENTER]. L'exécution du programme reprend dès que l'utilisateur introduit [↵] [ON]. La surface (ici, 20) se trouve dans la pile à l'issue de l'exécution.

■ Saisie d'un objet avec masquage de la pile



INPUT prend deux chaînes de caractères en arguments. Celle qui se trouve au niveau 2 est affichée en haut de l'écran (dans la zone d'état) alors que la chaîne placée au niveau 1 constitue le début de la ligne de commande à partir de laquelle l'utilisateur peut introduire un objet.

```

«
DEG ← Le degré devient l'unité de mesure d'angle

"SINUS & COS" ← Texte affiché en haut de l'écran
":X:" ← début de la ligne de commande
INPUT ← INPUT prend deux chaînes en arguments

OBJ→ ← OBJ→ restitue l'objet saisi

→ i
«
i sin
i cos
←
»
»

```

Les calculs sont effectués à l'aide d'une variable locale (i) qui prend pour valeur l'objet renvoyé par OBJ→. Cet objet n'est autre que celui introduit par l'utilisateur.

- CRC : #1994h / 71.5

Ce programme fait du degré l'unité de mesure d'angle courante. La chaîne "**SINUS & COS**" est affichée par `INPUT` en haut de l'écran alors que "**:X:**" constitue le début de la ligne de commande. `INPUT` interrompt l'exécution du programme tant que l'utilisateur n'a pas validé à l'aide de `[ENTER]` un objet introduit. `INPUT` renvoie ensuite dans la pile une chaîne de caractères qui n'est autre que la concaténation de la chaîne placée dans la ligne de commande avec l'objet saisi. `OBJ→` extrait ensuite de cette chaîne l'objet introduit par l'utilisateur et renvoie cet objet au niveau 1 de la pile. La suite du programme consiste en deux calculs effectués à l'aide d'une variable locale.



La seconde chaîne prise en argument par `INPUT` constitue, comme nous l'avons dit, le début de la ligne de commande. Cette chaîne peut éventuellement se résumer à une chaîne vide (""). Si cette chaîne n'est pas vide, son contenu doit impérativement être placé entre `:` et `:` afin de permettre à `OBJ→` de reconstituer correctement l'objet saisi à partir de la chaîne de caractères que renvoie `INPUT`.

■ Détection d'une pression sur une touche

«

DO

50 .1 BEEP ← Corps de la boucle

UNTIL

KEY ←

END

KEY renvoie 1 au niveau 1 si une touche est pressée. En plaçant 1 dans la pile, KEY agit comme si une condition était vérifiée ce qui provoque ici la sortie de la boucle DO... UNTIL... END. En effet, une condition est considérée comme vérifiée si le résultat du test est un réel non nul.

51 ==

"PRESSION SUR ENTER"

IFT

»

On exploite ici le numéro de la touche pressée renvoyé par KEY. Il s'agit ici d'une structure de test IFT.

- CRC : E47Ch / 82



Un test, par exemple `1 2 >`, renvoie 1 si la condition est vérifiée ou 0 si elle ne l'est pas. De même, `KEY` renvoie 0 si aucune n'est pressée ou alors le numéro de la touche suivi de 1 si une touche est pressée. `KEY` peut donc servir de condition d'entrée ou de sortie pour les boucles `DO... UNTIL... ou WHILE... REPEAT...`

Lors de son exécution **KEY** renvoie :

- 0 si aucune n'a été pressée,
- le numéro de la touche (voir **ASN**) suivi, au niveau suivant, de 1, si une touche a été pressée.

Le programme proposé ici émet un son tant qu'aucune touche n'a été pressée. Dès qu'une pression est détectée sur le clavier, le son cesse et une chaîne est renvoyée si la touche pressée est la touche **[ENTER]** (codée 51 puisqu'elle se trouve dans la cinquième ligne et la première colonne du clavier).

■ Création d'un écran de saisie

INFORM, indisponible sur HP-48 S/SX, permet de créer des écrans de saisie conviviaux. Ces écrans sont composés de différentes rubriques dotées de masques de saisie. **INFORM** prend cinq arguments.

```

«
"MON EQUATION" ← Titre de l'écran de saisie

{ ← Début de la liste définissant les rubriques
{ "X" "SAISIR X" 0 } ← Sous-liste définissant la première rubrique
{ "EQ" "SAISIR EQ" 9 } ← Sous-liste définissant la seconde rubrique
} ← Fin de la liste définissant les rubriques

{ 2 3 } ← Liste définissant le format d'affichage

{ NOVAL 'SIN(X)' } ← Valeurs de réinitialisation

{ π 'COS(X)' } ← Valeurs initiales

INFORM
»

```

- CRC : #3A5Bh / 144.5

Les arguments de **INFORM** occupent cinq niveaux de la pile. Certains de ces arguments sont facultatifs.

① Au niveau 5, une chaîne de caractères définit le nom de l'écran de saisie. Ici, **"MON EQUATION"** est le titre de l'écran de saisie. Ce titre peut éventuellement se résumer à une chaîne vide **"**.

② Au niveau 4, une liste définit les rubriques, chaque rubrique étant définie par une sous-liste. Par exemple, la liste { { **"X" "SAISIR X" 0** }

{ "EQ" "SAISIR EQ" 9 } } comprend deux sous-listes { "X" "SAISIR X" 0 } et { "EQ" "SAISIR EQ" 9 } définissant chacune une rubrique. Le nombre de sous-listes dépend du nombre de rubriques à mettre en place sur l'écran. Chaque sous-liste comprend :

- le titre de la rubrique sous la forme d'une chaîne de caractères, par exemple, "EQ",
- le texte d'aide est facultatif. Il est affiché en bas de l'écran quand la rubrique est sélectionnée et se présente sous la forme d'une chaîne de caractères, par exemple, "SAISIR EQ",
- le ou les types admis pour les objets saisis dans la rubrique peuvent ne pas être précisés. S'ils le sont, c'est au moyen d'un ou plusieurs entiers, par exemple 9, définissant le type ou les types admis (voir TYPE),

③ Au niveau 3, une liste contenant deux entiers précise le nombre de colonnes divisant l'écran et l'espace séparant les titres des masques qui leur sont associés. Le premier entier de la liste définit le nombre de colonnes, le second, l'espace à droite du titre de chaque rubrique. Par exemple, { 2 3 } répartit les rubriques sur deux colonnes et maintient l'espace par défaut (3) entre chaque titre et le masque qui lui est associé.

④ Au niveau 2, une liste donne pour chaque rubrique le contenu qui y sera placé en cas de réinitialisation de l'écran de saisie. Par exemple, { NOVAL 'SIN(X)' } donne les valeurs de réinitialisation des deux rubriques de l'écran. Il est possible ne pas préciser de valeur en remplaçant par NOVAL la valeur de réinitialisation de l'une des rubriques.

⑤ Au niveau 1, une liste donne pour chaque rubrique le contenu par défaut affiché initialement. Par exemple, { π 'COS(X)' } donne les valeurs placées dans les deux rubriques lorsque l'écran de saisie s'affiche. Il est possible ne pas préciser de valeur en remplaçant par NOVAL la valeur initiale de l'une des rubriques.



Bien qu'assez lourde à manipuler, l'instruction INFORM peut aisément être allégée en omettant ses paramètres facultatifs. Son emploi améliore grandement la convivialité des programmes qui l'emploient.

INFORM renvoie les données saisies dans la pile :

- si l'écran a été quitté avec (CANCL) ou [ON], INFORM place 0 dans la pile,
- si l'écran a été quitté avec (OK), INFORM place au niveau 2 de la pile une liste rassemblant les objets saisis en respectant l'ordre des rubriques puis insère 1 au niveau 1 de la pile. La récupération des données saisies est aisée : OBJ→ décompose la liste puis DROP supprime le réel représentant le nombre d'objets de la liste qui a été placé au niveau 1 par OBJ→.

L'écran créé avec `INFORM` étant affiché, les flèches de déplacement permettent de sélectionner une rubrique. Une rubrique étant sélectionnée (son champ de saisie est contrasté), une pression sur (`EDIT`) provoque l'édition de cette rubrique. Une pression sur (`OK`) valide l'objet en cours de saisie dans une rubrique ou valide l'ensemble de l'écran si aucune rubrique n'est en cours d'édition.

Le programme ci-dessous est une extension du précédent. Il place dans la pile les objets saisis grâce à l'écran créé avec `INFORM`.

«

```

"MON EQUATION"

{
{ "X" "SAISIR X" 0 }
{ "EQ" "SAISIR EQ" 9 }
}

{ 2 3 }

{ NOVAL 'SIN(X)' }

{ π 'COS(X)' }
    
```

Arguments de INFORM

INFORM

IF

1 ==

← *Test : INFORM a-t-il renvoyé 1 au niveau 1 ?*

THEN

```

OBJ→
DROP
    
```

Si INFORM a renvoyé 1 au niveau 1, OBJ→ renvoie dans la pile chaque élément de la liste fournie par INFORM. Le nombre d'objets renvoyés étant placé au niveau 1, DROP supprime ce nombre.

END

»

- CRC : #A890h / 167



Dans le cadre de cet exemple, l'instruction `LIST→` aurait pu être utilisée à la place de `OBJ→` mais cela n'aurait pas présenté le moindre intérêt.

■ **Menu vertical**

Contrairement à `INFORM` qui permet la saisie d'objets, `CHOOSE` permet la sélection d'un objet au moyen d'une liste déroulante. `CHOOSE` prend trois arguments et n'existe pas sur HP-48 S/SX.

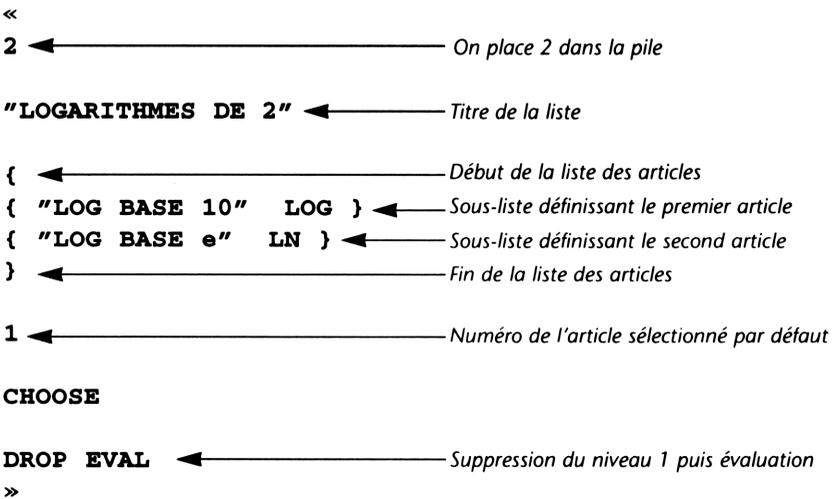
❶ Au niveau 3, une chaîne de caractères définit le nom de la liste. Ici, **"LOGARITHMES DE 2"** est le titre de la liste. Ce titre peut éventuellement se résumer à une chaîne vide "".

❷ Au niveau 2, une liste définit le contenu du menu vertical. Cette liste comprend des sous-listes, chaque sous-liste définissant un article du menu (vous pouvez en définir autant que vous le souhaitez). Une sous-liste comprend :

- une chaîne de caractères correspondant à l'article apparaissant dans le menu vertical,
- un objet (fonction intégrée, nom de variable, programme, objet quelconque...) renvoyé au niveau 2 de la pile si l'article correspondant est sélectionné.

Par exemple, `{ {"LOG BASE 10" LOG} {"LOG BASE e" LN} }` crée un menu comprenant deux articles `LOG BASE 10` et `LOG BASE e` plaçant respectivement `LOG` et `LN` au niveau 2 de la pile s'ils sont sélectionnés.

❸ Au niveau 1, un entier (ici, 1), détermine l'article du menu qui est sélectionné par défaut. Cet article apparaît donc contrasté à l'affichage du menu vertical.



• CRC : #111Ah / 97.5

L'emploi de la liste créée avec `CHOOSE` peut renvoyer deux types de résultats :

- si la liste a été quittée à l'aide de (`CANCL`) ou de [`ON`], `CHOOSE` renvoie 0 au niveau 1 de la pile,
- si un article de la liste a été sélectionné (avec [`↓`] et [`↑`]) puis validé avec (`OK`) ou [`ENTER`], `CHOOSE` renvoie au niveau 2 l'objet associé à l'article et place 1 au niveau 1.

Le programme proposé en exemple permet le calcul de *Log 2* ou *Ln 2* selon l'article sélectionné dans le menu.

```
«  
2  
"LOGARITHMES DE 2"  
{  
{ "LOG BASE 10" LOG }  
{ "LOG BASE e" LN }  
}  
1  
CHOOSE  
  
« EVAL »  
IFT  
»
```

- CRC : #9E4Fh / 110

Le programme ci-dessus n'évalue le résultat renvoyé par `CHOOSE` que si `CHOOSE` ne renvoie pas 0. En effet, `IFT` évalue l'objet placé au niveau 1, ici « `EVAL` », si le réel placé au niveau 2 est non nul.

2

Sorties

■ Affichage de la pile ou de l'écran graphique

Depuis un programme, il est très simple d'afficher la pile ou l'écran graphique (contenu de l'objet graphique `PICT`).



TEXT est l'instruction affichant la pile pendant l'exécution d'un programme. **PVIEW** (*Picture View*) affiche l'écran graphique. Dans le cas le plus simple, **PVIEW** prend une liste vide {} en argument. Cependant, il est aussi possible d'afficher **PICT** en plaçant dans l'angle supérieur gauche de l'écran un point (pixel) particulier de **PICT**. Les coordonnées de ce pixel doivent alors être données en argument à **PVIEW**. Ces coordonnées peuvent être exprimées en coordonnées utilisateur (*x, y*) ou en coordonnées système { #x #y }.

■ Affichage dans une zone de message



MSGBOX, indisponible sur HP-48 S/SX, affiche dans un cadre une chaîne de caractères donnée en argument. Une fois le message affiché, appuyez sur [ON], (OK) ou [ENTER] pour laisser le programme poursuivre son exécution.

```
«
"PROGRAMMEZ EN RPL !"
MSGBOX
»
```

- CRC : #D0ACh / 39.5

■ Affichage sur un niveau donné de la pile

DISP affiche dans la pile une chaîne de caractères ou un objet dont le nom est donné en argument. Cette chaîne ou cet objet doivent se trouver au niveau 2 de la pile alors qu'au niveau 1 doit se trouver le numéro de la ligne où l'objet doit être affiché.

```
«
"PROGRAMMEZ EN RPL !"
2 DISP ← Affichage sur la 2ème ligne
3 FREEZE ← «Gel» de la pile et de la zone d'état
```

```
WHILE ← Boucle sans fin attendant une pression sur
KEY 0 == une touche
REPEAT
END ←
»
```

- CRC : #D930h / 64

FREEZE (voir **FREEZE** dans la liste alphabétique des instructions RPL) fige l'une des zones de l'écran texte (zone d'état, pile, menu) et est couramment employé avec **DISP**.

```
«  
CLLCD  
"PROGRAMMEZ EN RPL !"  
2 DISP  
3 FREEZE  
HALT  
»
```

- CRC : #560Fh / 49

Ci-dessus, **HALT** interrompt l'exécution du programme et permet la saisie d'un ou plusieurs objets. Ces objets étant validés avec [ENTER], il faut taper [←] [ON] pour que le programme poursuive son exécution. Par ailleurs, **CLLCD** se charge de vider l'écran de son contenu.

■ Signal sonore

L'instruction **BEEP** émet un son. Elle prend deux arguments :

- au niveau 2, la fréquence du son exprimée en hertz,
- au niveau 1, la durée du son exprimée en secondes.

```
«  
50 2500  
FOR i  
  
i .2 BEEP  
  
50 STEP  
»
```

- CRC : #FFE7h / 68.5

Graphisme

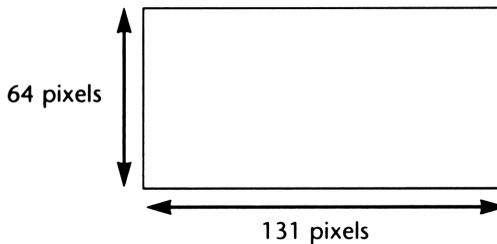
1

Commandes graphiques

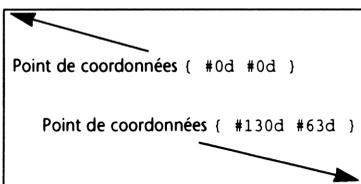
■ Les deux systèmes de coordonnées



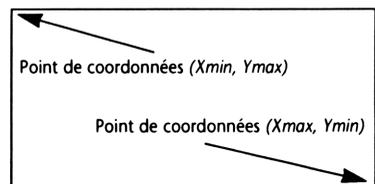
L'écran est une matrice de points appelés pixels. Physiquement, l'écran de la HP-48 est constitué de 131×64 pixels.



Chaque pixel peut être repéré à l'aide de ses coordonnées. Il existe deux systèmes de coordonnées : matérielles (système) et utilisateur.



Coordonnées matérielles



Coordonnées utilisateur



Les coordonnées matérielles (ou système) sont exprimées sous la forme d'une liste d'entiers binaires (en base 2, 8, 10 ou 16). Avec ce type de coordonnées, les coordonnées du point supérieur gauche d'un objet graphique sont toujours { #0h #0h } et toutes les coordonnées sont ici entières. Si un objet graphique comporte 5 lignes et 10 colonnes, les coordonnées matérielles de ses pixels varient entre { #0d #0d } pour le point supérieur gauche et { #9d #4d } pour le point inférieur droit.

Les coordonnées utilisateur sont directement liées au repère défini pour le tracé des courbes. Le contenu de la variable PPAR détermine les intervalles sur lesquels sont représentées les abscisses et les ordonnées :

- les abscisses sont représentées sur $[X_{\min}, X_{\max}]$,
- les ordonnées sont représentées sur $[Y_{\min}, Y_{\max}]$.

Avec ces deux systèmes de coordonnées, la première coordonnée repère horizontalement le point alors que la seconde le repère verticalement.



Les instructions $PX \rightarrow C$ et $C \rightarrow PX$ convertissent les coordonnées. $PX \rightarrow C$ transforme des coordonnées matérielles en coordonnées utilisateur. $C \rightarrow PX$ se charge de l'opération inverse.

■ Les instructions

Chaque pixel de l'écran pouvant être allumé ou éteint, des instructions permettent de traiter l'état de chaque pixel de l'écran graphique (contenu de PICT).



PVIEW prend en arguments les coordonnées matérielles du pixel à placer dans l'angle supérieur gauche puis provoque l'affichage de l'écran graphique. Autrement dit, PVIEW affiche l'objet graphique placé dans la variable réservée PICT.

Exceptées $PX \rightarrow C$ et $C \rightarrow PX$, toutes les instructions prenant un point en argument acceptent les deux systèmes de coordonnées. L'utilisateur peut donc choisir le système le mieux adapté à ses besoins.

PIXON, PIXOFF et PIX? prennent en argument les coordonnées d'un point. PIXON allume un point alors que PIXOFF l'éteint. PIX? renvoie 1 si le point dont les coordonnées sont fournies en arguments est allumé ou 0 s'il ne l'est pas.

BOX, LINE et TLINE prennent en arguments les coordonnées de deux points aux niveaux 1 et 2 de la pile. BOX trace un rectangle dont les côtés sont parallèles aux côtés de l'écran et dont la première diagonale a pour

extrémités les points dont les coordonnées ont été données en arguments. `LINE` trace un segment de droite dont les extrémités ont pour coordonnées celles données en arguments. `TLINE` agit comme `LINE` mais inverse l'état de tous les points constituant le segment de droite.

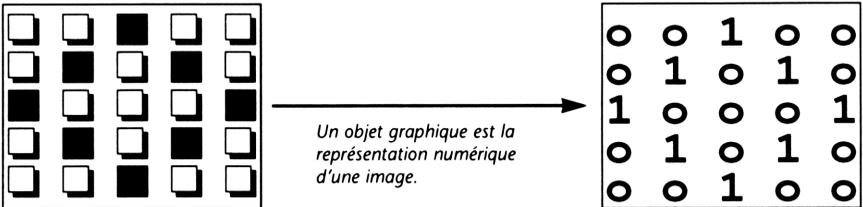
2

Les objets graphiques

■ Notions générales



Un objet graphique, ou GROB (*graphic object*) code une matrice de points rectangulaire. Un objet graphique est donc la représentation numérique d'une image composée de points noirs ou blancs. Un point allumé (noir) est représenté par un 1 alors qu'un point éteint (blanc) est représenté par un zéro.



Comme tous les autres objets, les objets graphiques (GROB) sont susceptibles d'être stockés dans des variables ou d'apparaître dans la pile. Placé dans la pile, un objet graphique s'affiche sous la forme :

Grob 1xh

où **1** représente la largeur et **h** la hauteur de l'objet graphique. **1** et **h** sont exprimées en pixels.

D'un point de vue logique la HP-48 dispose de deux écrans :

- l'écran texte affiche la pile,
- l'écran graphique correspond à l'affichage de PICT.

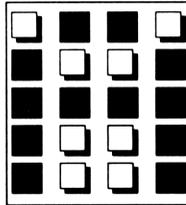
PICT est la variable réservée chargée de stocker l'objet graphique correspondant à l'écran graphique. Normalement, PICT contient un objet gra-



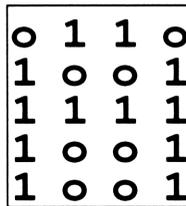
phique dont les dimensions sont celles de l'écran (131×64). Cependant, il est possible d'affecter à PICT un objet graphique beaucoup plus grand que l'écran physique. On parle alors d'écran graphique virtuel et la visualisation de l'intégralité de PICT n'est alors possible qu'en faisant défiler le contenu de l'écran graphique avec les flèches de déplacement.

■ Créer un objet graphique

Soit l'objet graphique défini par la matrice de points ci-dessous ;



Cet objet graphique comporte quatre colonnes et cinq lignes. Traduits en 0 et en 1, les états des points deviennent :



Cependant, un objet graphique doit être codé en employant un nombre de colonnes multiple de 8, et ce, même si l'objet graphique ne comporte qu'un nombre inférieur de colonnes utiles. Il convient donc de coder la matrice ci-dessous :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Un objet graphique est introduit sous la forme :

GROB 1 h d

1 et **h** représentent respectivement la largeur et la hauteur de l'objet graphique exprimées en pixels.

d correspond à la définition des pixels parcourus de gauche à droite et de bas en haut. **d** code donc **0110 0000 1001 0000 1111 0000 1001 0000 1001 0000**.

Ces 0 et ces 1 sont assimilables à des nombres binaires (nombre en base 2 couramment appelés *bits*). Pour augmenter la lisibilité de **d**, les nombres binaires sont remplacés par des nombres hexadécimaux (nombres en base 16).

Rappelons que les chiffres hexadécimaux sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, et F. Le tableau ci-dessous rappelle les équivalences entre les base 2, 10 et 16.

| Base 2 | Base 10 | Base 16 | Base 2 | Base 10 | Base 16 |
|--------|----------|---------|--------|-----------|---------|
| 0000 | 0 | 0 | 1000 | 8 | 8 |
| 0001 | 1 | 1 | 1001 | 9 | 9 |
| 0010 | 2 | 2 | 1010 | 10 | A |
| 0011 | 3 | 3 | 1011 | 11 | B |
| 0100 | 4 | 4 | 1100 | 12 | C |
| 0101 | 5 | 5 | 1101 | 13 | D |
| 0110 | 6 | 6 | 1110 | 14 | E |
| 0111 | 7 | 7 | 1111 | 15 | F |



Par rapport aux notations usuelles, le codage hexadécimal des objets graphiques suppose l'inversion de l'ordre des bits au sein d'un groupe de quatre bits. Ainsi devrait *a priori* être codé par 0111, qui, transformé en hexadécimal, devient 7. Compte tenu de l'inversion des bits, les quatre pixels deviennent 1110 qui est codé par E en hexadécimal. Attention ! Cette inversion ne s'applique qu'à des groupes de quatre pixels. Par exemple, est codé par 11100000 (E0) et non par 00001110 (0E).

Revenons à l'objet graphique que nous voulons introduire *via* la pile. Pour le saisir, il faut taper :

GROB 4 5 6090F09090



4 représente le nombre de colonnes. **5** est le nombre de lignes alors que **6090F09090** code les états des pixels. Attention ! La chaîne hexadécimale codant les états des pixels doit être saisie sans espace. Une fois l'objet graphique introduit et validé avec [ENTER], la calculatrice répond :

GROB 4×5

GROB 4x5 étant au niveau 1 de la pile, stockez-le dans une variable nommée GR1 en saisissant '**GR1**' [**STO**].

■ Afficher un objet graphique

```

«
ERASE ← Initialisation du contenu de PICT.

GROB 4 5 6090F09090 ← On crée un objet graphique 4x5 représentant
→ i ← la lettre A. Il est placé dans la pile et pris
comme valeur par la variable locale i.

«
PICT
{ #15h #10h }
i
GOR ← GOR superpose deux objets graphiques (avec
un effet de transparence). L'objet graphique
placé dans i est placé au-dessus de PICT.
L'angle supérieur gauche de i est placé au
niveau du point de coordonnées {#15h #10h}
de PICT.

{}
PVIEW ← PVIEW affiche PICT en plaçant le point dont les
coordonnées sont spécifiées dans l'angle supé-
rieur gauche de l'écran. Ici {} impose le centrage
de PICT sur l'écran et une pression sur [ENTER]
pour quitter l'affichage.
»
»

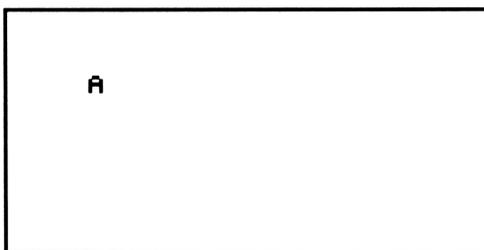
```

- CRC : #B5D5h / 87.5



Afin d'afficher **GROB 4 5 6090F09090**, le programme le place « au-dessus de PICT ». PICT est ensuite affiché à l'aide de PVIEW.

L'angle supérieur gauche de l'objet graphique défini par GROB 4 5 6090F09090 et stocké dans i se place à l'intersection de la 21e colonne (#15h) et de la 16e ligne (#10h) de PICT.



■ Superposition d'objets graphiques

Comme GOR, GXOR et REPL prennent trois arguments,

- au niveau 3, l'objet graphique d'arrière-plan,
- au niveau 2, les coordonnées dans l'objet graphique d'arrière-plan de l'angle supérieur gauche de l'objet graphique de premier plan,
- au niveau 1, l'objet graphique de premier plan.

GOR, GXOR et REPL gèrent la superposition de deux objets graphiques et renvoient un objet graphique dont les dimensions sont celles de l'objet graphique d'arrière-plan :

- GOR rend transparents les pixels blancs du premier plan,
- REPL rend opaques les pixels blancs du premier plan,
- GXOR n'allument que les pixels qui ne sont allumés que sur l'un des deux objets graphiques.



Le programme ci-dessous inverse l'état de chacun des pixels de PICT (fonction NEG) avant d'ajouter à PICT un objet graphique dont les points blancs ne sont pas transparents en raison de l'emploi de REPL à la place de GOR.

```

«
ERASE

GROB 4 5 6090F09090
→ i
    
```

```

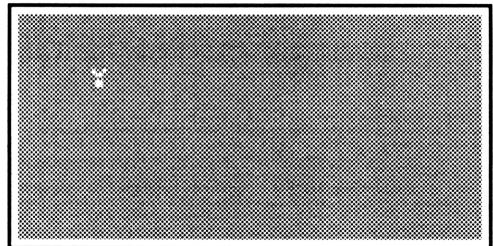
«
PICT NEG
PICT
{ #15h #10h }
i
REPL

{ }
PVIEW
»
    
```

On place PICT dans la pile puis on inverse l'état de ses pixels avec NEG. PICT étant un nom de variable, NEG ne renvoie rien, d'où l'obligation de réintroduire PICT dans la pile pour pouvoir l'utiliser comme argument de REPL.

- CRC : #1422h / 92.5

Le fond étant noir (à cause de ERASE PICT NEG PICT), la transparence provoquée par GOR aurait rendu l'objet graphique de premier plan totalement invisible. Ici, REPL permet donc de garder les points blancs visibles.



Après avoir mis en œuvre les instructions GOR et REPL, passons à GXOR, qui, finalement, inverse les états des points d'arrière-plan sous l'objet graphique de premier plan.

```

«
ERASE

GROB 4 5 6090F09090
→ i

```

```

«
PICT NEG

PICT
{ #15h #10h }
i
GXOR

```

```

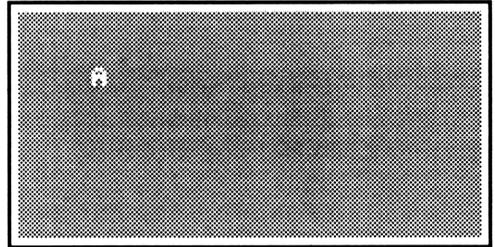
{} ←
PVIEW
»

```

{ } provoque le centrage de PICT sur l'écran et impose une pression sur [ENTER] pour quitter l'affichage graphique.

- CRC : #CF90h / 92.5

GXOR a inversé tous les points de l'arrière-plan placé sous l'objet de premier plan.



■ Autres fonctions graphiques



SIZE prend un objet graphique en argument et renvoie une liste de deux entiers binaires { #L #H } où #L code la largeur en pixels et #H la hauteur en pixels de l'objet graphique donné en argument.

BLANK renvoie un objet graphique blanc dont la largeur est définie par un entier binaire placé au niveau 2 et la hauteur, par un entier binaire placé au niveau 1.

SUB crée un objet graphique à partir d'un rectangle découpé dans un autre objet graphique. L'objet graphique père est placé au niveau 3 alors qu'aux niveaux 1 et 2 sont placées les coordonnées de la première diagonale du rectangle à découper.

3

Animation graphique

Le programme ci-dessous fait rouler une petite boule de gauche à droite. Il se fonde sur deux instructions GOR :

- la première place un point sur la boule afin de simuler un rayon tournant,
- la seconde place sur PICT la boule munie de son point.



Attention ! L'emploi de GOR (ou de GXOR ou de REPL) impose l'emploi de coordonnées relatives. Ainsi, l'angle supérieur gauche de l'objet graphique d'arrière-plan a toujours { #0 #0 } pour coordonnées.

```

«
GROB 7 7 C12214141422C1 ← Objet graphique représentant la boule.
GROB 1 1 10 ← Objet graphique représentant le point.

→ i j ← La boule est placée dans i, le point l'est dans j.
« ← Début de la visibilité de i et de j.
DO ← Début d'une boucle infinie.
      2 124 FOR k ← En-tête d'une boucle dont le compteur k varie
                    entre 2 et 124.
      ERASE ← Le contenu de PICT est effacé.
      PICT ← On place PICT dans la pile

      k R→B
      #20h 2
      2 →LIST ← On crée une liste { #k #20h } correspondant
                    aux coordonnées dans PICT de la boule munie
                    de son point.

      i ← On place la boule dans la pile.

      k 16 / FP → m ← Comme k varie par pas de 2 et que nous nous
                    intéressons à la détection de huitièmes de tour,
                    on extrait la partie décimale de k÷16 que l'on
                    affecte à la variable locale m.

«
CASE
m 0 == THEN 2 3 END
m .125 == THEN 2 4 END
m .25 == THEN 3 4 END
m .375 == THEN 4 4 END
m .5 == THEN 4 3 END
m .625 == THEN 4 2 END

```

Pour chaque huitième de tour, on place l'ordonnée puis l'abscisse du point à coller sur la boule.

```

m .75 == THEN 3 2 END
m .875 == THEN 2 2 END
END ← Fin de la structure CASE.
»

```

```

R→B
SWAP
R→B
2 →LIST ←

```

On crée la liste contenant les coordonnées du point sur la boule. Pour cela, on convertit l'abscisse en entier binaire, on inverse les deux premiers niveaux de la pile, on convertit l'ordonnée en entier binaire puis on construit une liste à partir de ces deux entiers binaires.

```

j ←

```

On insère j (le point) dans la pile.

```

GOR ←

```

On place le point sur la boule.

```

GOR ←

```

On place la boule munie d'un point sur PICT.

```

{#0h #0h} ←
PVIEW

```

On affiche PICT en plaçant son point de coordonnées { #0 #0 } dans l'angle supérieur gauche de l'écran.

```

.15 WAIT ←

```

Une petite pause de 0,15 seconde...

```

2 STEP ←

```

k augmente de 2. Fin de la boucle FOR.

```

UNTIL KEY ←
END

```

Attente d'une pression sur une touche pour sortir de la boucle infinie.

```

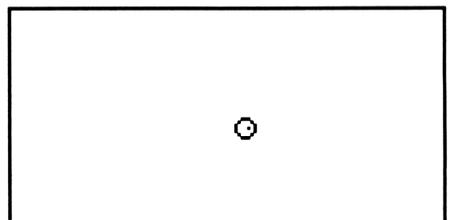
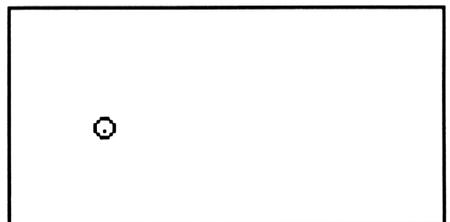
» ←
»

```

Fin de la visibilité de i et de j.

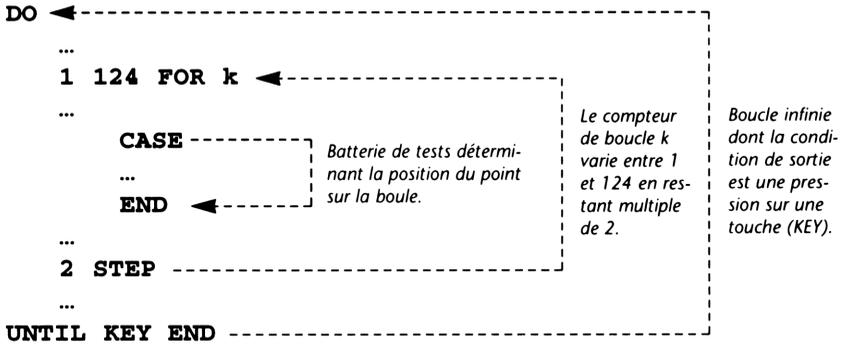
- CRC : #CD14h / 486.5

La petite boule tourne à l'infini en se déplaçant de gauche à droite. En fait, l'objet graphique définissant la boule reste inchangé. On se contente de lui ajouter un point pouvant prendre huit positions différentes afin de restituer un effet de rotation.





Ce programme repose sur deux boucles imbriquées contenant une batterie de tests (**CASE**) :



Par ailleurs, deux instructions **GOR** sont imbriquées. Rappelons que **GOR** prend trois arguments :

- au niveau 3, l'objet graphique d'arrière-plan,
- au niveau 2 les coordonnées de l'angle supérieur gauche de l'objet graphique de premier plan dans l'objet graphique d'arrière-plan,
- au niveau 1, l'objet graphique de premier plan.

Principe :

- 1 • On place **PICT** dans la pile.
- 2 • On place dans la pile les coordonnées de la boule dans **PICT**, ces coordonnées dépendent de k.
 - 3-1 • On place la boule (i) dans la pile.
 - 3-2 • On place dans la pile, en fonction de k et à l'aide d'une structure **CASE**, les coordonnées du point sur la boule.
 - 3-3 • On place le point (j) dans la pile.
 - 3-4 • On place le point sur la boule à l'aide **GOR**.
- 4 • On place la boule munie de son point sur **PICT** à l'aide de **GOR**.

Ces opérations graphiques sont répétées pour k variant de 2 à 124 (**2 124 FOR k**) par pas de 2 (**2 STEP**). Cette boucle est elle-même placée dans une boucle infinie (**DO... END**) dont la condition de sortie est une pression sur le clavier (**UNTIL KEY**).

Le RPL en prépa



Le programme présenté dans ce chapitre est issu du livre *HP-48 : Physique et chimie en prépa* (DUNOD). Il met en œuvre de nombreux objets de différents types et représente assez bien ce qu'il est possible de réaliser dans le cadre de la programmation parascolaire. Une nouvelle version de ce programme est maintenant disponible dans la deuxième édition du livre *HP-48 : Physique et chimie en prépa*.

■ Présentation du programme

Ce programme est consacré à l'étude des courants continus ou sinusoïdaux (régime permanent). Vous pouvez utiliser ce programme de deux façons différentes :

- en commençant par évaluer l'objet DEPART vous pourrez dessiner votre circuit puis vous obtiendrez automatiquement dans la pile les tensions entre chaque nœud du circuit,
- en commençant par évaluer l'objet ENTR, vous dessinerez uniquement votre circuit. Il vous faudra ensuite appeler un autre programme suivant les résultats souhaités.

Dans les deux cas, vous activez la fonction de tracé des circuits.

■ Utilisation du programme

Utilisons le programme en évaluant l'objet 'DEPART' :

- placez-vous dans le répertoire rassemblant les objets relatifs au programme,
- affichez les variables ([VAR]) et sélectionnez l'article du menu correspondant à l'objet 'DEPART',



- définissez les dimensions de la représentation graphique, l'écran « normal » possède 131×64 pixels, cela dit, vous pouvez saisir des valeurs supérieures (il vous faudra alors faire défiler l'objet pour le visualiser dans son intégralité),
- le circuit est maintenant représenté, sa seule partie toujours visible est la masse, placée en bas de l'écran. Le menu vous propose alors :
 - *VOIR* fait disparaître le menu afin de mieux visualiser le circuit, une pression sur [ON] réaffichera le menu,
 - *NOEUD* sert à la construction d'un nœud, pour cela :
 - sélectionnez l'article *NOEUD* du menu,
 - à l'aide des flèches de déplacement, placez l'index à l'endroit où vous souhaitez créer un nœud, marquez cette position en appuyant sur [ENTER], appuyez ensuite sur [ON],
 - le numéro 1 qui marque la position du nœud indique qu'il s'agit du premier nœud. Introduisez ensuite tous les nœuds nécessaires,
 - *BRAN* établit une branche de circuit entre deux nœuds préalablement définis, pour cela :
 - sélectionnez l'article *BRAN* du menu,
 - les extrémités d'une branche sont marquées par deux nœuds. A l'aide des flèches de déplacement, placez l'index sur le premier nœud, marquez cette position en appuyant sur [ENTER], de même, placez l'index sur le second nœud et marquez sa position en appuyant sur [ENTER], appuyez ensuite sur [ON],
 - il va maintenant vous falloir introduire les grandeurs caractérisant le contenu de la branche considérée. On passe d'un champ de saisie à l'autre en utilisant les flèches de déplacement ou en sélectionnant l'article *OK* du menu (selon le type de votre HP-48), on valide un écran en appuyant sur [ENTER]. Pour chaque branche, saisissez la capacité, l'inductance, la résistance et la f.e.m. des dipôles constituant la branche, une valeur nulle indique que le dipôle n'existe pas (par exemple, une capacité nulle indique que la branche ne comporte pas de condensateur, par suite, si une branche ne comporte pas de générateur, répondez 0 pour e, etc.). Notez que vous pouvez "inverser les pôles" d'un générateur en saisissant une f.e.m. négative. Les générateurs de Norton devront de toutes façons être modélisés à l'aide du modèle de Thévenin,



- le circuit étant terminé, sélectionnez l'article *EXIT* du menu, deux possibilités sont envisageables :
 - vous aviez évalué l'objet 'DEPART' du répertoire. Dans ce cas, l'affichage de la pile entraînera l'affichage des tensions entre chaque nœud (sous forme complexe si le circuit comporte une bobine ou un condensateur),
 - vous aviez évalué l'objet 'ENTR' Dans ce cas, vous quittez simplement l'affichage du circuit,
- ce que vous propose alors le menu :
 - *SMART* fait varier la fréquence des oscillations d'un circuit, validez votre choix par [ENTER],
 - *DRW* permet de tracer l'un des diagrammes de BODE. Après avoir précisé le nœud de sortie du quadripôle, il sélectionnera les deux nœuds de part et d'autre du générateur, puis, pour la sortie, le nœud 0 (masse) et celui que vous avez précisé,
 - avant de tracer le diagramme de Bode, vous pouvez choisir de tracer le gain en décibels.

■ Le principe du programme

Le programme forme n équations linéaires dont les n inconnues sont les tensions au n nœuds du circuit par rapport à la masse. On construit donc le circuit en indiquant les nœuds et les composants dans les branches reliant ces nœuds. Pour tracer le diagramme de Bode, on donne différentes valeurs à la pulsation ω et on trace le logarithme de la tension à la sortie en fonction de $\log(\omega)$.

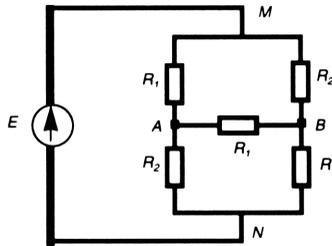
■ Objets associés au programme

- '**DEPART**' : demande la saisie des éléments nécessaires à la constitution d'un circuit et calcule les tensions aux différents nœuds,
- '**BODE**' : demande la saisie des éléments nécessaires à la constitution d'un circuit et trace le diagramme de Bode correspondant,
- '**SMART**' : calcule les tension aux nœuds et les présente dans la pile,
- '**DRW**' : trace le diagramme de Bode pour le dernier circuit introduit,
- '**CALC**' : calcule les tensions au nœuds,
- '**ENTR**' : demande la saisie des éléments nécessaires à la constitution d'un circuit,
- '**CIRCUIT**' : variable contenant le dernier circuit introduit,
- '**BODE.EQ**' : équation utilisée pour tracer le diagramme de Bode,
- '**PHI.EQ**' : équation utilisée pour le tracé du diagramme de Bode relatif à la phase,

- ' ω ' : pulsation du circuit,
- '**NS**' : contient le numéro du nœud correspondant à la sortie (pour le diagramme de Bode),
- '**MASSE**' : dessin de la masse du circuit,
- '**GR**' : trace une branche du circuit,
- '**CO**' : contient les coordonnées des nœuds,
- '**X**' : contient $\log(\omega)$ lors du tracé,
- '**EQ**' : contient '**BODE.EQ**' ou '**PHI.EQ**' selon le cas,
- '**PPAR**' : contient les paramètres de la fenêtre graphique.

■ Exemple d'après Mine de Douai, 1991

On considère un générateur de tension continue de f.e.m. E et de résistance interne négligeable. On réalise, avec ce générateur, trois résistances R_1 et deux résistances R_2 , le montage ci-dessus.



Déterminez numériquement le courant I_1 qui circule de A vers B.

Données numériques :

$$E = 12 \text{ V}$$

$$R_1 = 6 \ \Omega$$

$$R_2 = 3 \ \Omega$$

On construit tout d'abord un nouveau circuit en lançant '**DEPART**'. Le programme demande les dimensions de l'écran : le circuit étant de taille raisonnable on peut se contenter de 100 par 100, appuyez ensuite sur [ENTER]. Ensuite apparaît la masse en bas de l'écran ainsi que les menus. Placez le premier nœud correspondant à M dans l'énoncé (la masse correspondant ici à N) ; choisissez la touche de menu '**NOEUD**', vous vous trouvez alors dans l'environnement GRAPH (Attention ! le pointeur étant au milieu et la masse tout en bas, celle-ci peut ne pas être visible ; il suffit de descendre pour l'afficher) ; déplacez le pointeur à l'endroit souhaité puis appuyez sur [ENTER] puis sur [ON] pour revenir au programme.

Pour placer le générateur, appuyez sur 'BRANCHE'. Placez le pointeur près du 1 de la masse, appuyez sur [ENTER], placez le curseur près du 0, appuyez sur [ENTER] puis sur [ON] pour revenir au programme qui vous demande confirmation.

Appuyez sur «O» ([EVAL]), ou sur «N» ([STO]) selon le cas ; dans ce dernier cas, il faudra alors recommencer l'opération en sélectionnant l'article *BRANCHE*. Le programme demande alors les dipôles présents sur la branche. Il faut répondre 0 pour les trois dans le cas de l'exercice, puis appuyez sur [ENTER].

Ensuite, le programme demande la valeur de la f.e.m. du générateur sur la branche, répondez 12 pour cet exercice. Puis le programme revient au menu après avoir dessiné la branche. Ensuite placez le nœud 2 (pour B) et la résistance R_2 (mettre 0 pour E).

Placez de même le nœud 3 (pour A) et les résistances R_1 et R_2 restantes, pour obtenir le circuit. Appuyez alors sur 'EXIT' pour lancer les calculs (mettre 0 pour la fréquence). Vous obtiendrez alors les tensions aux différents nœuds dans la pile :

V1 : 12
V2 : 7.2
V3 : 4.8

L'intensité traversant AB est donc $i = (4,8 - 7,2)/6 = -0,4$ A. Le courant circule donc de B vers A ($V_B > V_A$) avec une intensité de 0,4 Ampères.

Dans le cadre de cet exemple, on a placé les nœuds, puis immédiatement après, les dipôles sur chaque branche. On peut aussi placer tout les nœuds d'abord, puis les branches dans un ordre quelconque. A tous moments, vous pouvez appuyer sur 'VOIR' pour visualiser l'ensemble du circuit si ses dimensions sont plus grandes que celles de l'écran de la machine (131 par 64).

Tous les objets doivent être réunis dans un répertoire particulier. Nommez-le par exemple ELEC.

- 'DEPART'

« ENTR SMART »

- 'BODE'

« ENTR DRW »

• 'SMART' (CRC : #B5AC)

```
«
"Fréquence des oscillations (0 si courant continu)"
{ V } INPUT OBJ→ 2
* π * 'ω' STO CALC
ARRAY→ LIST→
  FOR J "V" J +
→TAG CIRCUIT 1 GET
ROLLD -1
STEP
»
```

• 'DRW' (CRC : #BB48)

```
«
"Noeud de sortie ?"
{ V } INPUT OBJ→
'NS' STO 'BODE.EQ'
STEQ # 83h # 40h
PDIM # 4h RES 0 6
XRNG -5 5 YRNG
ERASE DRAX DRAW
GRAPH 33 MENU
»
```

• 'CALC' (CRC : #CE23)

```
«
  IF ω 0 ==
  THEN
.0000000001 'ω' STO
  END CIRCUIT
LIST→ DUP 1 - SWAP
1 + ROLL DUP DUP2 2
→LIST (0,0) CON
SWAP 1 →LIST (0,0)
CON → N M E
  « 1 SWAP
  START LIST→
DROP 4 ROLLD ω *
(0,1) * INV SWAP ω
* (0,1) * + +
  IF DUP 0
```

```

==
                THEN DROP
.00000001
                END SWAP
OVER / SWAP INV 4
ROLL 4 ROLL
                CASE DUP
0 ==
                THEN
DROP 'M' OVER DUP 2
→LIST DUP2 GET 5
ROLL + PUT 'E' SWAP
DUP2 GET 4 ROLL -
PUT
                END
OVER 0 ==
                THEN
SWAP DROP 'M' OVER
DUP 2 →LIST DUP2
GET 5 ROLL + PUT
'E' SWAP DUP2 GET 4
ROLL + PUT
                END → j
k
                « 'M' j
k 2 →LIST DUP2 GET
4 PICK - PUT 'M' k
j 2 →LIST DUP2 GET
4 PICK - PUT 'M' j
DUP 2 →LIST DUP2
GET 4 PICK + PUT
'M' k DUP 2 →LIST
DUP2 GET 4 ROLL +
PUT 'E' j DUP2 GET
4 PICK + PUT 'E' k
DUP2 GET 4 ROLL -
PUT
                »
                END
                NEXT E M /
                IF DUP IM
ABS .00000001 ≤
                THEN RE
                END
                »
                »

```

• 'ENTR' (CRC : #EABA)

```
      << RCLMENU -22
SF # 83h # 40h PDIM
0 130 XRNG 0 63
YRNG (0,0)
"Dimensions de l'écran"
{ ":X:
:Y:" V 4 }
INPUT OBJ→ 1 - 63
MAX SWAP 1 - 130
MAX SWAP R→C PDIM
PICT SIZE DUP2
# 40h - SWAP # 83h
- SWAP 2 →LIST DUP
PVIEW 3 ROLLD 15 -
SWAP 2 / 15 - SWAP
2 →LIST PICT OVER
MASSE REPL PX→C 7 +
1 →LIST 'C0' STO 0
→ m C N
      << { }
'CIRCUIT' STO {
"NOEUD" "BRANCHE"
"" "VOIR" "" "EXIT"
} TMENU
      WHILE -1
WAIT IP DUP 16 ≠
      REPEAT
      CASE DUP
11 ==
      THEN
DROP GRAPH 'C0'
OVER STO+ PICT SWAP
'N' INCR 1 →GROB
REPL C PVIEW
      END DUP
12 ==
      THEN
DROP GRAPH 0 0 MAXR
MAXR → C1 C2 N1 N2
M1 M2
      << 0 N
FOR J C0 J 1 + GET
      IF C1 OVER - ABS
DUP M1 ≤
```

```

    THEN J 'N1' STO
'M1' STO
    ELSE DROP
    END
    IF C2 - ABS DUP
M2 ≤
    THEN J 'N2' STO
'M2' STO
    ELSE DROP
    END
NEXT "" 4 DISP "" 5
DISP "" 6 DISP "" 7
DISP "Branche " N1
+ "-" + N2 +
" ? (O/N)" + 3 DISP
WHILE 0 WAIT IP {
32 33 } OVER POS
NOT
REPEAT DROP
END
IF 33 ==
THEN N1 N2
"Branche " N1 + "-"
+ N2 + DUP {
":R:
:L:
:C:" V 4 }
INPUT OBJ→ DTAG ROT
DTAG ROT DTAG ROT 4
ROLL " Générateur"
+ { ":E:" V 4 }
INPUT OBJ→ DTAG GR
    IF OVER 0 ==
    THEN SWAP INV
SWAP
    END 6 →LIST 1
→LIST 'CIRCUIT'
SWAP STO+
END C PVIEW
    >
    END DUP
14 ==
    THEN
DROP { } PVIEW C
PVIEW
    END

```

```
DROP 380 .1 BEEP
      END
      END DROP m
MENU N 'CIRCUIT'
STO+
  »
  »
```

- 'CIRCUIT'

```
{ }
```

- 'BODE.EQ' (CRC : #8D4E)

```
<< X ALOG 'ω'
STO CALC NS GET ABS
LOG
  »
```

- 'PHI.EQ' (CRC : #877C)

```
<< X ALOG
'ω'
STO CALC
NS GET ARG
  »
```

- 'ω'

```
0
```

- 'NS'

```
2
```

- 'MASSE' (CRC : #9FBF)

```
GROB 16 15 00100820082008200010000000100010001
000100010CFFF429429429421
```

- 'GR' (CRC : #E9C0)

```
<< 6 →LIST ""
SWAP 3
DO GETI
IF 0 ≠
```

```

      THEN
"EOORLC" OVER DUP
SUB 4 ROLL SWAP + 3
ROLLD
      END
      UNTIL -64 FS?
      END DROP C0
OVER 1 GET 1 + GET
(1,-2) + C0 3 PICK
2 GET 1 + GET
(1,-2) + DUP2 + 2 /
3 ROLLD DUP2 - SIGN
4 * ROT OVER - 3
ROLLD + LINE PICT 4
ROLL
      IF "" OVER ==
      THEN 3 DROPN
      ELSE 1 →GROB
DUP SIZE 1 + SWAP 1
+ SWAP BLANK { # 1h
# 1h } ROT
REPL DUP
SIZE 2 + SWAP 2 +
SWAP BLANK NEG {
# 1h # 1h } ROT
REPL DUP SIZE B→R 2
/ SWAP B→R -2 /
SWAP R→C 4 ROLL +
SWAP REPL
      END
LIST→
DROP
»

```

- 'C0'

```
{ }
```

- 'X'

```
6
```

- 'EQ'

```
PHI.EQ
```

- 'PPAR'

```
{  
  (0,-3.14159265359)  
  (6,3.14159265359) X  
  # 4h (0,0) FUNCTION  
  Y }  
END
```

2

Références pour le RPL



Outre un guide des instructions RPL, cette partie rassemble de nombreuses informations utiles au développeur :

- codes des messages-système,
- codes ASCII des caractères HP-48 (avec équivalences hexadécimales),
- rôles des indicateurs système,
- codes des menus des HP-48 G/GX.

Les codes des menus des HP-48 S/SX sont fournis par HP. Vous les trouverez dans les manuels livrés avec la machine.

Rappel des notations utilisées pour les objets :

- ' ' délimite un nom,
- { } délimite une liste,
- [] délimite un vecteur,
- [[]] délimite une matrice,
- « » délimite un programme RPL.

Un tableau peut être une matrice ou un vecteur.

Familles d'instructions



Ce chapitre vous présente plus de 500 instructions classées par familles. Nous avons distingué 25 familles d'instructions réparties en six groupes. Le programmeur peu aguerri est invité à parcourir ce chapitre afin de découvrir de nouvelles instructions. Les instructions sont ici classées afin de faciliter leur identification, ceci dit, l'explication détaillée de leur fonctionnement est l'objet du prochain chapitre.

1

Mathématiques et physique

1.1

Mathématiques générales



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--------------------------------|-------------------------|------------------------------|-----------------------------|
| * | Produit | • | • | |
| + | Addition ou concaténation | • | • | |
| - | Soustraction | • | • | |
| / | Quotient | • | • | |
| = | Création d'une équation | • | • | |
| ^ | Exponentiation (« puissance ») | • | • | |
| ! | Factorielle | • | • | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 5/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| % | Pourcentage | • | • | |
| ABS | Valeur absolue | • | • | |
| ACOS | Arc cosinus | • | • | |
| ACOSH | Arc cosinus hyperbolique | • | • | |
| ALOG | Fonction 10 puissance x | • | • | |
| ARG | Argument d'un nombre complexe | • | • | |
| ASIN | Arc sinus | • | • | |
| ASINH | Arc sinus hyperbolique | • | • | |
| ATAN | Arc tangente | • | • | |
| ATANH | Arc tangente hyperbolique | • | • | |
| CEIL | Recherche de l'entier immédiatement supérieur ou égal | • | • | |
| %CH | Différence exprimée en pourcentage | • | • | |
| COLCT | Rassemblement des termes dans une expression symbolique | • | | |
| COMB | Calcul de combinaisons (probabilités, etc.) | • | • | |
| CONJ | Conjugué d'un nombre complexe ou d'un tableau de complexes | • | • | |
| COS | Cosinus | • | • | |
| COSH | Cosinus hyperbolique | • | • | |
| C→R | Extraction des parties réelle et imaginaire d'un nombre complexe | • | | |
| CROSS | Produit vectoriel | • | | |
| CYLIN | Activation du mode de coordonnées cylindriques | | | • |
| DEG | Activation du mode degrés (unité de mesure d'angles) | | | |
| DOT | Produit scalaire | • | | |
| D→R | Conversion de degrés en radians | • | • | |
| e | Nombre de Népere | • | • | |
| EQ→ | Séparation des membres gauche et droit d'une équation | • | | |
| EXP | Fonction exponentielle | • | • | |
| EXPAN | Développement des multiplications et des exponentiations dans une expression | • | | |
| EXPM | Fonction exponentielle moins un | • | • | |
| FACT | Fonction factorielle (!) | • | • | |
| FLOOR | Plus grand entier inférieur ou égal au nombre réel spécifié | • | • | |
| FP | Extraction de la partie décimale | • | • | |
| GRAD | Utilisation du grade comme unité de mesure d'angle | • | | |
| i | Nombre imaginaire (constante symbolique i pour les nombres complexes) | • | • | |
| IM | Extraction de la partie entière d'un nombre décimal | • | • | |
| INV | Calcul de l'inverse (1/x) | • | • | |
| IP | Extraction de la partie réelle d'un nombre complexe | • | • | |
| ISOL | Isolation d'une variable dans une expression symbolique | • | • | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| LININ | Vérification de la linéarité d'une expression algébrique | • | | • |
| LN | Logarithme népérien | • | • | |
| LNPI | Logarithme népérien plus 1 | • | • | |
| LOG | Logarithme décimal | • | • | |
| MANT | Extraction de la mantisse | • | • | |
| MAX | Extraction de la plus grande valeur parmi celles se trouvant aux niveaux 1 et 2 de la pile | • | • | |
| MAXR | Plus grand nombre pouvant être traité par la HP-48 | • | • | |
| MCALC | Déclaration d'une variable comme valeur calculée par le solveur | • | | • |
| MIN | Extraction de la plus petite valeur parmi celles se trouvant au niveaux 1 et 2 de la pile | • | • | |
| MINIT | Initialisation du menu des équations multiples | | | • |
| MINR | Plus petit nombre réel manipulé par la HP-48 | • | • | |
| MITM | Modification du menu des équations multiples | • | | |
| MOD | Modulo | • | • | |
| MROOT | Résolution d'équations multiples | • | | • |
| MSOLVR | Utilisation des variables du solveur d'équations multiples | | | • |
| MUSER | Déclaration d'une variable utilisateur (EqLib) | • | | • |
| NDIST | Distribution normale (probabilités) | • | | • |
| NEG | Recherche de l'opposé | • | • | |
| PCOEF | Recherche des coefficients d'un polynôme à partir de ses racines | • | | • |
| PERM | Permutations (probabilités) | • | • | |
| PEVAL | Valeur d'un polynôme pour une valeur donnée de la variable indépendante | • | | • |
| PROOT | Racines d'un polynôme calculées à partir de ses coefficients | • | • | • |
| →Q | Transformation d'un nombre décimal en une fraction | • | • | |
| →Q π | Transformation d'un nombre décimal en une fraction rationnelle facteur π | • | • | |
| QUAD | Résolution d'équation quadratique | • | • | |
| QUOTE | Place entre quotes un argument symbolique pour une fonction de l'utilisateur | • | | |
| RAD | Utilisation du radian comme unité de mesure d'angle | | | |
| RATIO | Division (en position préfixe) | • | | |
| R→C | Création d'un complexe à partir de deux réels | • | | |
| R→D | Conversion de radians en degrés | • | • | |
| RE | Partie réelle d'un nombre complexe | • | • | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| RECT | Mode coordonnées rectangulaires | | | • |
| REPL | Remplacement d'une portion d'un tableau, d'une liste ou d'un objet graphique | • | | |
| RKF | Résolution d'équation différentielle par la méthode de Runge-Kutta-Fehlberg | • | | • |
| RKFERR | Estimation d'erreur lors de l'utilisation de la méthode de Runge-Kutta-Fehlberg | • | | • |
| RKFSTEP | Intervalle entre les solutions fournies par RKF | • | | • |
| RND | Arrondi | • | • | |
| ROOT | Valeur de la variable indépendante pour laquelle un programme ou une expression s'annule | • | | |
| RRK | Résolution d'équation différentielle avec dérivées partielles connues | • | | • |
| RRKSTEP | Intervalle entre deux valeurs utilisées par RRK | • | | • |
| RSBERR | Estimation d'erreur liée à la méthode de Rosenbrock | • | | • |
| SHOW | Remplacement dans une expression des variables par les valeurs qu'elles dénotent | • | • | |
| SIGN | Signe | • | • | |
| SIN | Sinus | • | • | |
| SINH | Sinus hyperbolique | • | • | |
| SOLVEQN | Exécution du solveur d'équations multiples | • | • | • |
| SQ | Carré | • | • | |
| %T | Pourcentage | • | • | |
| TAN | Tangente | • | • | |
| TANH | Tangente hyperbolique | • | • | |
| TAYLR | Polynôme de Taylor | • | | |
| UTPC | Distribution Khi carré à droite (probabilités) | • | | |
| UTPF | Distribution F de Snedecor à droite (probabilités) | • | | |
| UTPN | Distribution normale à droite (probabilités) | • | | |
| UTPT | Distribution T de Student à droite (probabilités) | • | | |
| →V2 | Création d'un vecteur du plan ou d'un nombre complexe à partir de deux réels | • | • | |
| →V3 | Création d'un vecteur de l'espace à partir de trois réels | • | • | |
| V→ | Séparation des éléments d'un vecteur ou d'un nombre complexe | • | • | |
| XPON | Extraction de l'exposant | • | • | |
| XROOT | Racine x-ième | • | • | |
| | Dans une expression, remplacement des noms par les valeurs qu'ils dénotent | • | • | |
| ∂ | Dérivation | • | • | |
| Σ | Somme d'une série finie | • | • | |
| π | Constante π | • | • | |
| ∫ | Intégration | • | • | |
| √ | Racine carrée | • | • | |

1.2

Opérations logiques

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|-----------------------------|-------------------------|------------------------------|-----------------------------|
| ≠ | Inégalité entre deux objets | • | • | |
| ≤ | Infériorité large | • | • | |
| ≥ | Supériorité large | • | • | |
| == | Egalité entre deux objets | • | • | |
| < | Infériorité stricte | • | • | |
| > | Supériorité stricte | • | • | |
| ADD | Somme de deux listes | • | | • |
| AND | Et logique | • | • | |
| OR | Ou logique | • | • | |
| XOR | Ou logique exclusif | • | • | |



Distinguez le signe « = » du connecteur « == ».

1.3

Traitement des entiers binaires

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| ASR | Décalage d'un bit à droite (entier binaire) | • | • | |
| BIN | Activation du mode binaire (base 2) | | • | |
| B→R | Conversion d'un entier binaire en nombre réel | • | • | |
| DEC | Activation du mode décimale (base 10) | | • | |
| HEX | Activation du mode hexadécimal (base 16) | | • | |
| OCT | Activation du mode de calcul octal (base 8) | | • | |
| R→B | Conversion d'un réel en entier binaire | • | • | |
| RCWS | Place dans la pile la taille du mot binaire utilisé | • | • | |
| RL | Rotation d'un bit vers la gauche | • | • | |
| RLB | Rotation d'un octet vers la gauche | • | • | |
| RR | Rotation d'un bit vers la droite | • | • | |
| RRB | Rotation d'un octet vers la droite | • | • | |
| SL | Décalage d'un bit vers la gauche | • | • | |
| SLB | Décalage d'un octet vers la gauche | • | • | |
| SR | Décalage d'un bit vers la droite | • | • | |
| SRB | Décalage d'un octet vers la droite | • | • | |
| STWS | Taille du mot binaire utilisé pour les calculs | • | • | |

1.4

Objets avec unités

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| CONVERT | Conversion de l'unité d'un objet-unité | • | | |
| UBASE | Conversion d'une unité quelconque en unité du système international | • | • | |
| UFACT | Factorisation d'objets avec unités | • | | |
| →UNIT | Création d'un objet unité à l'aide d'un réel et d'une unité | • | | |
| UVAL | Extraction de la partie numérique d'un objet avec unité | • | • | |
| - | Association d'une unité à un nombre réel | • | | |

1.5

Tableaux, matrices et vecteur

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| →ARRY | Création d'un tableau | • | | |
| ARRY→ | Copie d'un tableau dans la pile | • | | |
| CNRM | Norme de la colonne spécifiée | • | | |
| →COL | Conversion d'une matrice en vecteurs ou d'un vecteur en ses éléments | • | | • |
| COL+ | Insertion de colonnes dans une matrice ou d'un élément dans un vecteur | • | | |
| COL- | Suppression d'une colonne d'une matrice ou suppression d'un élément d'un vecteur | • | | |
| COL→ | Création d'une matrice à partir d'une colonne | • | | • |
| COLΣ | Détermination des variables dépendantes et indépendantes d'une matrice statistique | • | | |
| CON | Création d'un tableau (matrice ou vecteur) dont tous les éléments sont égaux à une constante donnée | • | • | |
| COND | Calcul du nombre de conditions de la norme de colonne d'une matrice | • | | • |
| CSWP | Permutation des colonnes d'une matrice | • | | • |
| DET | Déterminant d'une matrice carrée | • | • | |
| →DIAG | Création d'un vecteur fondé sur les éléments diagonaux d'une matrice | • | | • |
| DIAG→ | Création d'une matrice à partir de dimensions et d'éléments diagonaux | • | | • |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| EGV | Valeurs propres et vecteurs propres d'une matrice carrée | • | | • |
| EGVL | Valeurs propres d'une matrice carrée | • | | • |
| FFT | Transformée de Fourier | • | | • |
| GET | Extraction d'un élément d'une liste ou d'une matrice | • | | |
| GET | Extraction d'un élément d'une liste ou d'une matrice | • | | |
| GETI | Extraction d'un élément d'une liste ou d'une matrice puis incrémentation du pointeur dans la liste ou dans le tableau | • | • | |
| IDN | Création d'une matrice identité | • | | |
| IFFT | Inverse de la transformée de Fourier | • | | • |
| LQ | Factorisation (L) d'une matrice | • | | • |
| LSQ | Norme minimale d'un système d'équations linéaires | • | • | • |
| LU | Décomposition LU d'une matrice carrée | • | | • |
| PUT | Remplacement d'un élément d'un tableau ou d'une liste | • | | |
| PUTI | Remplacement d'un élément d'un tableau ou d'une liste puis incrémentation du pointeur | • | • | |
| QR | Factorisation QR | • | | • |
| RANK | Rang d'une matrice | • | • | • |
| RANM | Matrice aléatoire | • | | • |
| RCI | Produit d'une ligne d'un tableau et d'une constante | • | | • |
| RCIJ | Produit d'une ligne d'un tableau avec une constante puis somme de cette ligne avec la ligne suivante | • | | • |
| RDM | Changement des dimensions d'un tableau | • | | |
| REPL | Remplacement d'une portion d'un tableau, d'une liste ou d'un objet graphique | • | | |
| RNRM | Norme de ligne d'un tableau | • | | |
| →ROW | Conversion d'une matrice en vecteurs ou d'un vecteur en ses éléments | • | | • |
| ROW+ | Insertion d'un élément, d'un vecteur ou d'une matrice dans un tableau | • | | |
| ROW- | Suppression d'une ligne dans un tableau | • | | |
| ROW→ | Création d'une matrice à partir de vecteurs ou d'un vecteur à partir de ses éléments | • | | • |
| RREF | Forme réduite échelonnée | • | • | • |
| RSD | Calcul de résidus | • | | |
| RSWP | Permutation de deux lignes d'une matrice | • | | • |
| SCHUR | Décomposition (Schur) d'une matrice carrée | • | | • |
| SIZE | Taille | • | | |
| SNRM | Norme spectrale | • | | • |
| SRAD | Rayon spectrale d'une matrice carrée | • | | • |
| SVD | Décomposition en valeurs singulières | • | | • |
| SVL | Valeurs singulières | • | | • |
| TRN | Transposition | • | | |

1.6

Nombres aléatoires

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| RAND RDZ | Nombre aléatoire Définition du nombre générateur de nombres aléatoires pour RAND | • • | | |

1.7

Statistiques

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|---------------|--|-------------------------|------------------------------|-----------------------------|
| BESTFIT | Sélection automatique du mode de régression | • | | |
| BINS | Tri par blocs | | | |
| CL Σ | Suppression des données statistiques | | | |
| CORR | Coefficient de corrélation de la matrice statistique courante | • | | |
| COV | Covariance | • | | |
| EXPFIT | Modèle d'ajustement exponentiel pour les calculs de régression | | | |
| Σ Line | Expression définissant la droite d'ajustement | • | | |
| LINFIT | Utilisation du modèle d'ajustement linéaire pour les calculs de régression | | | |
| LOGFIT | Utilisation du modèle d'ajustement logarithmique pour les calculs de régression | | | |
| LR | Régression linéaire | • | | |
| MAX Σ | Valeurs maximales des différentes colonnes de la matrice statistique courante | • | | |
| MEAN | Moyenne de chaque colonne de la matrice statistique courante | • | | |
| MIN Σ | Extraction des valeurs minimales des différentes colonnes de la matrice statistique courante | • | | |
| N Σ | Nombre de lignes de la matrice statistique courante | • | | • |
| PCOV | Covariance | • | | • |
| PREDV | Valeur de la variable dépendante associée à une valeur donnée de la variable indépendante | • | | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|-----------------|---|-------------------------|------------------------------|-----------------------------|
| PREDX | Valeur de la variable indépendante associée à une valeur donnée de la variable dépendante | • | | |
| PREDY | Valeur de la variable dépendante associée à une valeur donnée de la variable indépendante | • | | |
| PSDEV | Ecart de population | • | | • |
| PVAR | Variance d'une population | • | | • |
| PWRFIT | Modèle d'ajustement de puissance pour les calculs de régression | • | | |
| RCLΣ | Place la matrice statistique courante dans la pile | • | | |
| SCATRPLOT | Nuage de points | | | |
| SDEV | Ecart standard sur échantillon | • | | |
| STOΣ | Définition de la valeur de ΣDAT | • | | |
| TOT | Somme du contenu des colonnes de la matrice statistique | • | | |
| VAR | Variance | • | | |
| ΣX | Somme des variables indépendantes | • | | |
| ΣX ² | Somme des carrés des variables indépendantes | • | | |
| XCOL | Définition de la colonne de la matrice statistique contenant les variables indépendantes | • | | |
| ΣX*Y | Somme des produits de la variable indépendante et de la variable dépendante | • | | |
| ΣY | Somme des variables dépendantes | • | | |
| ΣY ² | Somme des carrés de la variable dépendante | • | | |
| YCOL | Définition de la colonne de la matrice statistique contenant la variable dépendante | • | | |
| Σ+ | Ajout de données à la matrice statistique courante | • | | |
| Σ- | Renvoie les données relatives à la dernière entrée effectuée par Σ+ | • | | |

1.8

Physique

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| DARCY | Facteur Darcy d'un fluide en mouvement | • | | • |
| F0λ | Fraction du pouvoir d'émission (corps noir) | • | • | • |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| FANNING | Facteur Fanning de friction d'un fluide en mouvement | • | • | • |
| SIDENS | Densité intrinsèque du silicium | • | • | • |
| TDELTA | Différence de température | • | • | • |
| TINC | Ajout d'une différence de température | • | • | • |
| ZFACTOR | Facteur de correction de compressibilité d'un gaz (comportement non idéal d'un hydrocarbure) | • | • | • |

2

Mémoire, variables et bibliothèques

2.1

Variables, mémoire et répertoires



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| → | Introduction de variables locales | • | | |
| ARCHIVE | Sauvegarde du contenu de Home | • | • | |
| CLTEACH | Suppression du répertoire EXAMPLES et de son contenu | | | • |
| CLUSR | Suppression des variables et sous-répertoires du répertoire courant | | | |
| CLVAR | Suppression des variables et sous-répertoires du répertoire courant | | | |
| CONST | Place dans la pile la valeur d'une constante | • | • | • |
| CRDIR | Création d'un répertoire (dossier) | • | | |
| DECR | Décrémenter (par pas de un) d'une variable | • | | |
| DEFINE | Définition d'une variable ou d'une fonction | • | • | |
| HOME | Fait du répertoire racine (Home) le répertoire courant | | | |
| INCR | Incrémenter (pas de 1) d'une variable | • | | |
| MEM | Mémoire disponible | • | | |
| ORDER | Classement des variables d'un répertoire | • | | |
| PATH | Chemin d'accès au répertoire courant depuis la racine | • | | |
| PGDIR | Suppression d'un répertoire et de son contenu | • | | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| PURGE | Suppression de variables ou de répertoires vides | • | | |
| RCL | Place dans la pile un objet associé à une variable sans évaluation | • | | |
| RESTORE | Remplacement de Home par une de ses sauvegardes | • | | |
| CONJ | Remplace l'objet stocké dans une variable par son conjugué | • | | |
| SINV | Remplace l'objet stocké dans une variable par son inverse | • | | |
| SNEG | Remplace l'objet stocké dans une variable par son opposé | • | | |
| STO | Création d'une variable par liaison d'un objet à un identificateur (nom) | • | | |
| STO* | Produit d'un objet et d'un objet associé à une variable puis enregistrement dudit produit dans ladite variable | • | | |
| STO+ | Addition d'un objet à l'objet associée à une variable spécifiée | • | | |
| STO- | Calcul de la différence entre un objet donné et celui associé à une variable puis stockage de la différence dans ladite variable | • | | |
| STO/ | Quotient d'un objet et d'un objet associé à une variable puis stockage dans ladite variable dudit quotient | • | | |
| TEACH | Création du répertoire EXAMPLES | | | • |
| TVARS | Liste des variables du répertoire courant contenant des objets d'un type spécifié | • | | |
| UPDIR | Retour vers le répertoire-père | | | |
| VARS | Liste des variables du répertoire courant | • | | |
| VTYPE | Type de l'objet associé à la variable spécifiée | • | | |

2.2

Bibliothèques



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| ATTACH | Association d'une bibliothèque (library) à un répertoire | • | | |
| CONLIB | Ouverture du catalogue d'une bibliothèque de constantes | | | • |
| DETACH | Dissociation d'une bibliothèque et du répertoire auquel elle est attachée | • | | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|---------------------------|--|-------------------------|------------------------------|-----------------------------|
| EQNLIB LIBEVAL LIBS | Accès à la bibliothèque d'équations (EQLIB) Evaluation d'une fonction de bibliothèque Informations relatives aux bibliothèques associées au répertoire courant | • • | | • • |

2.3

Ports et cartes-mémoire



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|----------------|---|-------------------------|------------------------------|-----------------------------|
| FREE | Désolidarisation de la carte mémoire (RAM) du port 1 utilisée en mémoire fusionnée | • | | |
| FREE1 | Désolidarisation de la carte mémoire (RAM) du port 1 utilisée en mémoire fusionnée | • | | • |
| MERGE | Fusion de la mémoire vive interne et de la carte RAM du port 1 | • | | |
| MERGE1 | Fusion de la mémoire vive interne et de la carte RAM du port 1 | | | • |
| PINIT PVARs | Initialisation des ports ouverts Liste des sauvegardes (objets-sauvegardes) et des bibliothèques | • | | • |

3

Interface-utilisateur et graphisme

3.1

Gestion de la pile



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|----------------|---|-------------------------|------------------------------|-----------------------------|
| CLEAR DEPTH | Suppression du contenu de la pile Nombre de niveaux occupés dans la pile | • • | | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| DROP | Suppression d'un objet dans la pile | • | | |
| DROP2 | Suppression des objets des niveaux 1 et 2 de la pile | • | | |
| DROPN | Suppression de n objets dans la pile | • | | |
| DTAG | Suppression des libellés ou identifications d'un objet | • | | |
| DUP | Place au niveau 2 de la pile une copie de l'objet se trouvant au niveau 1 | • | | |
| DUP2 | Place au niveau 3 de la pile une copie du niveau 1, et au niveau 4, une copie du niveau 2 | • | | |
| DUPN | Place sur n niveaux de la pile une copie de l'objet se trouvant au niveau 1 | • | | |
| OVER | Copie du contenu du niveau 2 de la pile | • | | |
| PICK | Copie du contenu d'un niveau spécifié de la pile au niveau 1 | • | | |
| ROLL | Place le contenu de l'un des niveaux de la pile au niveau 1 et refoule les autres niveaux vers le haut | • | | |
| ROLLD | Place le contenu du niveau 1 de la pile à un niveau spécifié et décale les niveaux intermédiaires vers le bas | • | | |
| ROT | Fait passer le contenu du niveau 3 de la pile au niveau 1 et refoule les autres niveaux | • | | |
| SWAP | Inverse les contenus des niveaux 1 et 2 | • | | |

3.2

Gestion des objets



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| BYTES | Taille et somme de contrôle d'un objet | • | | |
| EVAL | Evaluation d'un objet | • | • | |
| IFT | Evaluation d'un objet en fonction de la valeur se trouvant au niveau 2 de la pile | • | | |
| IFTE | Evaluation de l'objet se trouvant au niveau 1 de la pile ou de celui se trouvant au niveau 2 en fonction de la valeur se trouvant au niveau 3 | • | | |
| NEWOB | Place la copie d'un objet dans la pile et supprime l'original | • | • | |
| →NUM | Evaluation fournissant un résultat numérique | • | | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| OBJ→ | Dissociation des éléments élémentaires composant un objet ou évaluation d'un objet élémentaire | • | | |
| SAME | Test de l'identité de deux objets | • | | |
| SIZE | Taille | • | | |
| SYSEVAL | Evaluation d'un objet dont l'adresse est connue | • | | |
| →TAG | Création d'un objet « avec étiquette » | • | | |
| TYPE | Type d'un objet (numéro du type) | • | | |

3.3

Interface-utilisateur, saisie et format des données



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| ASN | Affectation d'un objet à une touche | • | • | |
| CHOOSE | Création d'un menu vertical | • | | • |
| DELKEYS | Annulation des fonctionnalités associées aux touches redéfinies | • | • | |
| ENG | Mode d'affichage « ingénieur » | • | | |
| FIX | Détermination du nombre de décimales (mode « Fix ») | • | | |
| INFORM | Création d'une zone de dialogue | • | | • |
| INPUT | Introduction de données via la ligne de commande | • | | |
| KEY | Détection d'une pression sur une touche | • | | |
| MATCH↓ | Ecriture du haut vers le bas | • | | |
| MATCH↑ | Ecriture du bas vers le haut | • | | |
| MENU | Affichage d'un menu intégré ou d'un menu personnalisé | • | | |
| MSGBOX | Création d'une zone de dialogue | • | | • |
| NOVAL | Absence de valeur par défaut dans une rubrique d'une zone de dialogue créée avec INFORM | | | • |
| PROMPT | Affichage d'une chaîne de caractères et interruption temporaire de l'exécution | • | | |
| RCLKEYS | Touches redéfinies par l'utilisateur | • | • | |
| RCLMENU | Numéro du menu courant | • | | |
| SCI | Mode d'affichage dit scientifique | • | | |
| STD | Mode standard pour l'affichage des nombres | | | |
| STOKEYS | Affectation des touches du clavier | • | • | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|---------------|---|-------------------------|------------------------------|-----------------------------|
| TMENU TRNC | Menu temporaire Troncation du nombre de décimales ou du nombre de chiffres significatifs | • • | • | |

3.4

Manipulations des drapeaux

Les drapeaux (aussi appelés *flags*, *system flags* ou indicateurs) dont il est question ici correspondent à ceux accessibles par [←I] [CST] (FLAG) sur HP-48 G/GX. Les instructions qui suivent permettent, entre autres, l'armement (activation) ou le désarmement (désactivation) d'un drapeau pendant l'exécution d'un programme.



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| CF | Désarmement d'un indicateur (drapeau) | • | | |
| FC? | Test de l'état d'un indicateur (drapeau) | • | | |
| FC?C | Test de l'état d'un indicateur (drapeau) et désarmement de l'indicateur | • | | |
| FS? | Test de l'état d'un indicateur (drapeau) | • | | |
| FS?C | Test de l'état d'un indicateur (drapeau) et désarmement de l'indicateur | • | | |
| RCLF | Etats des 128 indicateurs (drapeaux) | • | • | |
| SF | Armement d'un indicateur (drapeau) | • | • | |
| STOF | Définition de l'état d'un indicateur (drapeau) | • | • | |

3.5

Graphisme

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| ANIMATE | Animation d'objets graphiques | • | | • |
| ARC | Dessin d'un arc | • | • | |
| ATICK | Intervalles entre les graduations sur les axes du repère | • | | • |
| AUTO | Calcul automatique des paramètres de l'axe des ordonnées | | | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| AXES | Paramètres relatifs aux axes du repère | • | | |
| BLANK | Création d'un objet graphique « blanc » | • | | |
| BOX | Dessin d'un rectangle | • | | |
| CENTR | Définition des coordonnées du centre d'un tracé | • | | |
| CLLCD | Supprime tout le contenu de l'écran sauf la barre des menus | | | |
| CONIC | Active le mode « Conic » pour la représentation graphique de sections coniques | • | | |
| C→PX | Conversion de coordonnées de l'utilisateur en coordonnées exprimées en pixels sur l'écran | • | | |
| DEPND | Définition de la variable dépendante | • | | |
| DIFFEQ | Activation du mode graphique permettant la représentation d'équations différentielles | | | • |
| DISP | Affichage d'un objet sur une ligne spécifiée de l'écran | • | | |
| DRAW | Tracé de la représentation graphique | | • | |
| DRAX | Représentation des axes du repère | | | |
| ERASE | Initialisation de l'affichage graphique | | | |
| EYEPT | Définition de la position de l'observateur lors d'un tracé en perspective | • | | • |
| FREEZE | Blocage temporaire de l'affichage | • | | |
| FUNCTION | Activation du mode Fonction (fonctions cartésiennes) | | • | |
| GOR | Ou logique appliqué à des objets graphiques | • | | |
| GRAPH | Activation de l'environnement PICT (picture) | | | |
| GRIDMAP | Activation du mode de tracé Gridmap pour la représentation d'une fonction de deux vecteurs à deux variables | | | • |
| →GROB | Création d'un objet graphique à partir du contenu d'un niveau de la pile | • | | |
| GXOR | Création d'un objet graphique par « ou logique exclusif » entre deux autres objets graphiques | • | | |
| *H | Zoom ne concernant que l'axe des ordonnées | • | | |
| INDEP | Définition de la variable indépendante et de ses paramètres lors d'un tracé | • | | |
| LABEL | Ajout du noms des axes sur ceux-ci | | | |
| →LCD | Insertion d'un objet graphique sur l'affichage de la pile | • | | |
| LCD→ | Création d'un objet graphique (GROB) fondé sur l'affichage courant | • | | |
| LINE | Tracé d'un segment de droite à partir des coordonnées de ses extrémités | • | | |
| NUMX | Nombre de points représentés associés à la variable indépendante pour chaque point représenté de la variable dépendante | • | | • |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| NUMY | Nombre de points représentés associés à la variable dépendante • | | • | |
| PARAMETRIC | Activation du mode de tracé paramétrique • | | • | |
| PARSURFACE | Activation de mode de tracé de trois vecteurs à deux variables • | | | • |
| PCONTOUR | Activation du mode de tracé d'une fonction scalaire à deux variables • | | | • |
| PDIM | Remplacement du contenu de pict par un objet graphique vierge de dimensions spécifiées • | • | | |
| PICT | Place PICT au niveau 1 de la pile • | • | | |
| PICTURE | Activation de l'environnement graphique • | | | • |
| PIX? | Test de l'état d'un pixel • | • | | |
| PIXOFF | Extinction d'un pixel • | • | | |
| PIXON | Allumage d'un pixel • | • | | |
| PMAX | Définition des coordonnées de l'angle supérieur droit de PICT • | • | | |
| PMIN | Définition des coordonnées de l'angle inférieur gauche de PICT • | • | | |
| POLAR | Activation du mode de tracé de fonctions polaires • | | • | |
| PVIEW | Visualisation de PICT • | • | | |
| PX→C | Conversion de coordonnées exprimées en pixels en coordonnées exprimées en coordonnées-utilisateur • | • | | |
| RCEQ | Rappel de l'expression EQ • | • | | |
| REPL | Remplacement d'une portion d'un tableau, d'une liste ou d'un objet graphique • | • | | |
| RES | Intervalle entre deux valeurs utilisées de la variable indépendante • | • | | |
| SCALE | Définition de l'intervalle entre les graduations de la représentation graphique • | • | | |
| SIMU | Tracés simultanés • | | • | • |
| SLOPEFIELD | Activation du mode SLOPEFIELD pour les tracés • | | | • |
| SPHERE | Activation du mode de coordonnées sphériques • | | | • |
| STEQ | Stockage dans EQ • | • | | |
| SUB | Extraction d'une portion d'une chaîne de caractères, d'une liste ou d'un objet graphique • | • | | |
| TEXT | Affichage de la pile • | | | |
| TLINE | Tracé d'un segment de droite par inversion de l'état de chaque pixel du segment • | • | | |
| TRUTH | Mode de tracé Truth • | | | |
| *W | Zoom horizontal • | • | | |
| WIREFRAME | Activation du mode de tracé d'une fonction scalaire à deux variables avec vue en perspective • | | | • |
| XRNG | Etendue représentée sur l'axe des abscisses • | • | | |
| XVOL | Valeurs extrêmes de la variable indépendante x (coordonnées volumiques) • | • | | • |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| XXRNG | Valeurs extrêmes de la variable indépendante x (représentation graphique de deux ou trois vecteurs à deux variables) | • | | • |
| YRNG | Etendue représentée sur l'axe des ordonnées | • | | |
| YSLICE | Activation du mode de tracé d'une fonction scalaire à deux variables | | | • |
| YVOL | Profondeur de la vue volumique | • | | • |
| YRNG | Valeurs extrêmes de la variable d'une zone d'entrée | • | | • |
| ZVOL | Hauteur de la vue volumique | • | | • |

3.6

Graphisme lié aux statistiques

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| BAR | Paramètres pour un tracé de type « Bar » | | | |
| BARPLOT | Exécution d'un tracé de type « BAR » | | | |
| HISTOGRAM | Activation du mode de tracé d'histogramme | | | |
| HISTPLOT | Tracé d'un histogramme fondé sur le contenu d'une colonne de la matrice statistique courante | | | |
| SCATTER | Mode graphique SCATTER | | | |
| SCLΣ | Ajustement des paramètres de pict pour qu'un tracé en nuage remplisse pict | | | • |

4

Listes et chaînes de caractères

4.1

Gestion des listes

Certaines fonctions relatives aux tableaux s'appliquent aussi aux listes. Consultez donc également le paragraphe consacré aux tableaux.



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| DOLIST | Permet d'utiliser une liste comme argument d'une fonction, d'un programme, etc. | • | | • |
| DOSUBS | Utilisation de groupes d'éléments d'une liste comme arguments d'une fonction, d'un programme | • | | • |
| ENDSUB | Nombre de blocs dans la liste traitée par DOSUBS | | | • |
| GET | Extraction d'un élément d'une liste ou d'une matrice | • | | |
| GETI | Extraction d'un élément d'une liste ou d'une matrice puis incrémentation du pointeur dans la liste ou dans le tableau | • | • | |
| HEAD | Extraction du premier élément d'une liste | • | | • |
| →LIST | Création d'une liste à partir des éléments se trouvant dans la pile | • | | |
| ΔLIST | Différences entre les éléments d'une liste | • | | • |
| πLIST | Produit des éléments d'une liste | • | | • |
| ΣLIST) | Somme des éléments d'une liste | • | | • |
| LIST→ | Place chaque élément d'une liste sur un niveau distinct de la pile | • | | |
| NSUB | Obtention du numéro de l'élément d'une liste utilisé par le programme | | | • |
| POS | Position d'un fragment de chaîne dans une chaîne de caractères ou position d'un élément d'une liste dans celle-ci | • | | |
| PUT | Remplacement d'un élément d'un tableau ou d'une liste | • | | |
| PUTI | Remplacement d'un élément d'un tableau ou d'une liste puis incrémentation du pointeur | • | • | |
| REPL | Remplacement d'une portion d'un tableau, d'une liste ou d'un objet graphique | • | | |
| REVLIST | Inversion d'une liste | • | | • |
| SEQ | Calcul séquentiel | • | | • |
| SIZE | Taille | • | | |
| SORT | Tri selon un ordre croissant | • | | • |
| STREAM | Exécution récursive | • | | • |
| SUB | Extraction d'une portion d'une chaîne de caractères, d'une liste ou d'un objet graphique | • | | |
| TAIL | Liste privée de son premier élément ou chaîne privée de son premier caractère | • | | • |



Comme nous l'avons dit, certaines fonctions prenant des tableaux comme arguments peuvent aussi être utilisées avec des listes. Ceci étant dit, l'inverse est aussi vrai. En effet, de nombreuses fonctions plutôt destinées aux listes sont utilisables avec des tableaux (vecteurs, matrices, etc.), des chaînes de caractères ou même, avec des objets graphiques (GROB). Les instructions GET, GETI, POS, PUT, PUTI, REPL et SUB sont des instructions s'appliquant autant aux listes qu'aux chaînes de caractères.

4.2

Manipulation des chaînes de caractères

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| + | Concaténation | • | • | |
| CHR | Affichage du caractère correspondant au code spécifié | • | | |
| NUM | Code ASCII (ASCII HP-48) du premier caractère d'une chaîne | • | | |
| POS | Position d'un fragment de chaîne dans une chaîne de caractères ou position d'un élément d'une liste dans celle-ci | • | | |
| SIZE | Taille | • | | |
| →STR | Place un objet dans une chaîne de caractères | • | • | |
| STR→ | Evaluation du contenu d'une chaîne de caractères | • | | |
| SUB | Extraction d'une portion d'une chaîne de caractères, d'une liste ou d'un objet graphique | • | | |
| TAIL | Liste privée de son premier élément ou chaîne privée de son premier caractère | • | | • |
| TSTR | Création d'une chaîne à partir d'une date et d'une heure | • | • | |

5

Programmation et gestion des erreurs

5.1

Structures de programmation et contrôle de l'exécution

De nombreuses instructions sont indissociables d'un groupe d'instructions, par exemple, If... Then... Else ou Do... Until... End.



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| CASE | Structure de test | | | |
| CONT | Reprise de l'exécution d'un programme interrompu | | | |
| DEBUG | Arrêt du programme immédiatement après son lancement | • | | |
| DO | Début d'une boucle indéfinie Do... Until... End | • | | |
| ELSE | En-tête de l'action devant être exécutée si une condition n'est pas remplie | | | |
| END | Délimite une boucle, une procédure d'interception d'erreur ou une structure de test | | | |
| FOR | Début d'une boucle For... Next ou For... Step | • | | |
| HALT | Interruption de l'exécution d'un programme | | | |
| IF | Début de structure conditionnelle If... Then... (Else...) End | • | | |
| KILL | Annulation des programmes stoppés | | | |
| NEXT | Fin d'une boucle commencée avec For... ou Start... | | | |
| NEXT | Etape suivante d'un programme | | | |
| NOT | Complément à 1 (ou inverse logique) | • | • | |
| REPEAT | Structure de boucle While... Repeat... End | | | |
| SST | Exécution pas à pas | | | |
| SST↓ | Exécution de la prochaine instruction dans un programme en cours d'exécution pas à pas | | | |
| START | Début de boucle Start... Next ou Start... Step | • | | |
| STEP | Pas d'incréméntation (incrément) de la variable contrôlant une boucle | | | |
| THEN | En-tête de la procédure à exécuter si la condition est remplie dans une structure If... Then, Case... Then ou Iferr... Then | | | |
| UNTIL | Test dans une structure de boucle de type Do... Until... End | | | |
| WAIT | Interruption temporaire de l'exécution | • | | |
| WHILE | Test dans une boucle While... Repeat... End | • | | |

5.2

Gestion des erreurs



Les instructions de traitement des erreurs permettent de modifier l'exécution du programme en cas d'erreur (Iferr) mais aussi de rappeler le dernier message d'erreur ou son numéro (Errm, Errn).



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|---------------|---|-------------------------|------------------------------|-----------------------------|
| DOERR ERRO | Simulation d'un type d'erreur spécifié Suppression du dernier numéro d'erreur enregistré | • | | |
| ERRM | Place au niveau 1 de la pile le dernier message d'erreur renvoyé | • | | |
| ERRN | Place au niveau 1 de la pile le numéro de la dernière erreur | • | | |
| IFERR | Début de structure conditionnelle de traitement d'une éventuelle erreur | | • | |

6

Autres fonctions

6.1

Heure, date et alarme



On appelle « alarme » une sonnerie ou un événement programmé. Les alarmes (ou sonneries programmées) sont mémorisées dans une liste.



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| ACK | Accusé de réception d'une sonnerie programmée | | • | |
| ACKALL | Accusé de réception de toutes les sonneries | | • | |
| CLKADJ | Modification de l'horloge système | • | | |
| DATE | Place la date-système au niveau 1 de la pile | • | • | |
| →DATE | Modification de la date-système | • | • | |
| DATE+ | Ajout d'un nombre de jours à une date | • | • | |
| DDAYS | Intervalle entre deux dates | • | • | |
| DELALARM | Annulation d'une sonnerie programmée | • | | |
| FINDALARM | Index de la sonnerie programmée déclenchée après un délai spécifié | • | • | |
| →HMS | Conversion d'un nombre décimal en son équivalent au format Heures-minutes-secondes | • | | |
| HMS+ | Somme de deux données de type Heures-minutes-secondes | • | | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| HMS- | Différence entre deux données de type Heures-minutes-secondes | • | | |
| HMS→ | Conversion d'une donnée de type Heures-Minutes-secondes en son équivalent décimal | • | | |
| RCLALARM | Rappel d'une sonnerie programmée | • | | |
| STOALARM | Ajout d'une sonnerie programmée (alarme) dans la liste des alarmes | • | • | |
| TICKS | Heure système | • | | |
| TIME | Heure système au format Heures-Minutes-Secondes | • | | |
| →TIME | Définition de l'heure système | • | | |
| TSTR | Création d'une chaîne à partie d'une date et d'une heure | • | • | |

6.2

Finance et gestion

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|------------------------------|-------------------------|------------------------------|-----------------------------|
| AMORT | Amortissement | • | • | • |
| TVM | Menu Solveur TVM | | | • |
| TVMBEG | Paiement en début de période | | | • |
| TVMEND | Paiement en fin de période | | | • |
| TVMROOT | Solution TVM | • | | • |

6.3

Communication série

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|--|-------------------------|------------------------------|-----------------------------|
| BAUD | Définition de la vitesse de transmission | • | | |
| BUFLEN | Nombre de caractères dans le tampon RS 232 | • | | |
| CKSM | Définition du type de contrôle utilisé | • | | |
| CLOSEIO | Fermeture des ports infrarouge et série | | | |
| CR | Impression du contenu du tampon | | • | |



| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| DELAY | Intervalle de temps entre chaque ligne envoyée par la HP-48 à son imprimante | • | • | |
| FINISH | Quitte le mode serveur Kermit | | • | |
| IR | Activation du port série (câble) ou du port infrarouge | • | • | |
| KERRM | Affichage des messages d'erreur liées au protocole Kermit | • | | |
| KGET | Téléchargement d'un objet spécifié depuis un serveur Kermit distant | • | • | |
| OLDPRT | Compatibilité entre les jeu de caractères de la HP-48 et celui de l'imprimante HP-82240 | | | |
| OPENIO | Ouverture du port IR ou du port série | | • | |
| PARITY | Définition du mode de contrôle de la parité | • | | |
| PKT | Emission de paquets (Kermit) | • | • | |
| PR1 | Impression de l'objet placé au niveau 1 | • | • | |
| PRLCD | Impression de l'écran courant | | • | |
| PRST | Impression du contenu de la pile | | • | |
| PRSTC | Impression du contenu de la pile, les objets plus longs qu'une ligne d'affichage étant tronqués | | • | |
| PRVAR | Impression d'une variable | • | • | |
| RECN | Réception de données via Kermit puis stockage de ces données dans une variable | • | • | |
| RECV | Réception de données via Kermit | | • | |
| SBRK | Interruption de la transmission | | • | |
| SEND | Emission d'une variable | • | • | |
| SERVER | Place la HP-48 en mode serveur Kermit | | • | |
| SRECV | Lecture de la mémoire tampon d'entrée du port série | • | • | |
| STIME | Temporisation | • | | |
| TRANSIO | Mode de traduction | • | | |
| XMIT | Emission d'une chaîne (sans Kermit) | • | • | |
| XRECV | Mode réception (protocole XModem) | • | • | • |
| XSEND | Emission d'un objet en utilisant (XModem) | • | • | • |

6.4

Divers

| Instructions | Descriptions | Modification de la pile | Modification des indicateurs | Indisponible sur HP-48 S/SX |
|--------------|---|-------------------------|------------------------------|-----------------------------|
| APPLY | Association d'une fonction à ses arguments | • | | |
| BEEP | Emission d'un son | • | • | |
| LAST | Arguments de la dernière instruction exécutée | • | • | |
| LASTARG | Arguments de la dernière instruction exécutée | • | • | |
| MINEHUNT | Jeu du démineur | | | • |
| OFF | Extinction | | | |
| VERSION | Version de la calculatrice | • | | • |
| WSLOG | Informations relatives au redémarrage | • | • | • |

Index alphabétique des instructions RPL



Vous trouverez dans ce chapitre environ 500 instructions classées par ordre alphabétique. Les instructions représentées par des symboles non alphabétiques (→, par exemple) sont classées selon l'ordre des codes ASCII croissant à la suite de la liste alphabétique. Les instructions composées de caractères alphabétiques mêlés à des symboles non alphabétiques se trouvent dans la liste alphabétique, les symboles non alphabétiques n'étant pas pris en compte pour leur classement.

Toutes les instructions ne sont pas compatibles avec toutes les versions de HP-48. Un panneau de « sens interdit » signale les versions incompatibles avec une instruction donnée. Par exemple...



...signale les instructions inexistantes sur HP-48 S et SX.



Les instructions indisponibles sur HP-48 S et SX et accessibles à l'aide de [↵] [3]... sur HP-48 G et GX, sont utilisables sur HP-48 SX à condition de disposer de la carte HP-Solve.

Chaque instruction est présentée dans un tableau dans lequel vous trouverez successivement le nom de la fonction, les séquences de touches permettant d'accéder à ladite fonction, la liste des drapeaux (indicateurs ou *flags*) modifiant l'effet de l'instruction ou modifiés par celle-ci, l'état de la pile avant utilisation et contenant les éventuels arguments, l'état de la pile après utilisation et contenant les éventuels résultats, enfin, quelques commentaires qui vous aideront à mettre en œuvre les fonctions et instructions proposées. Dans les commentaires, les instructions sont notées avec des caractères particuliers, par exemple : `instruction`.

ABS**Valeur absolue, norme d'un vecteur, norme de Frobenius**

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] (ABS) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) (ABS)

1: → 1:

L'utilisation la plus simple de fonction permet la recherche de la valeur absolue d'un nombre réel ou d'un objet-unité (qui n'est autre que l'association d'un nombre réel à une unité). Notez que ABS permet aussi le calcul de la norme d'un vecteur du plan (ce qui revient au calcul du module d'un nombre complexe). Pour cela, placez (x, y) dans la pile. ABS renverra le résultat de $\sqrt{x^2+y^2}$ ce qui correspond à la norme du vecteur (x, y) et au module du complexe $x+iy$. Plus généralement, ABS renvoie la norme de Frobenius de son argument. En admettant que celui-ci soit un tableau, ABS, renvoie la racine carrée de la somme des carrés de ces éléments. Vous pouvez aussi utiliser une expression symbolique 'expr_symb' comme argument de ABS qui renvoie alors une nouvelle expression symbolique 'ABS(expr_symb)'. En cas de dérivation, SYMB est renvoyée comme dérivée de ABS.

ACK**Accusé de réception d'une alarme arrivée à échéance**

Accès (HP-48 G/GX) : [◀] [4] (ALRM) (ACK) Drapeaux : -43, -44

Accès (HP-48 S/SX) : [◀] [4] (ACK)

ACK supprime de la liste des alarmes la plus ancienne alarme échue (une alarme est une sonnerie programmée). Attention ! Le témoin « alarme » ne disparaît après activation de ACK que s'il n'y a plus aucune autre alarme échue ni aucune cause matérielle d'alarme (alimentation faible, etc.).

**ACKALL****Accusé de réception de toutes les alarmes arrivées à échéance**

Accès (HP-48 G/GX) : [◀] [4] (ALRM) (ACKA) Drapeaux : -43, -44

Accès (HP-48 S/SX) : [◀] [4] (ACKA)

ACKALL supprime de la liste des alarmes toutes les alarmes échues (une alarme est une sonnerie programmée). Attention ! Le témoin « alarme » ne disparaît après activation de ACKALL que s'il n'y a plus de cause matérielle d'alarme (alimentation faible, etc.).



ACOS

Arc cosinus

Accès : [↵] [COS]

Drapeaux : -1, -3, -17, -18

1: réel ou complexe



1: réel ou complexe

A une valeur comprise entre -1 et 1, ACOS fait correspondre l'angle (exprimé à l'aide de l'unité courante) dont le cosinus correspond à cette valeur. Cet angle renvoyé par ACOS est la solution principale puisque qu'il existe plusieurs angles correspondant à un cosinus donné. Si l'argument a de ACOS n'est pas compris dans l'intervalle $[-1,1]$, il est transformé en un complexe $a + 0.i$ et le résultat renvoyé est alors complexe. Appliqué à une expression symbolique 'expr_symb', ACOS renvoie une nouvelle expression symbolique 'ACOS (expr_symb)'.

ACOSH

Arc cosinus hyperbolique

Accès : [MTH] (HYP) (ACOSH)

Drapeaux : -1, -3

1: réel ou complexe



1: réel ou complexe



Attention ! ACOSH n'est pas exactement l'inverse de COSH, ACOSH est l'inverse de COSH sur une partie de son domaine. Si l'argument a de ACOSH est réel et appartient à l'intervalle $[-1,1]$, il sera automatiquement remplacé par un argument complexe $a + 0.i$. Appliqué à une expression symbolique 'expr_symb', ACOSH renvoie une nouvelle expression symbolique 'ACOSH (expr_symb)'.

ADD

Somme commutative de deux listes



Accès : [MTH] (LIST) (ADD)

Drapeaux : -

 2: liste
 1: liste

 2:
 1: liste

 2: objet
 1: liste

 2:
 1: liste

ADD crée une liste dont les éléments sont les sommes des éléments de mêmes rangs des deux listes fournies comme arguments. On peut aussi placer dans la pile une liste et un objet de même type que ceux contenus dans la liste, cet objet sera alors ajouté à ceux de la liste. Si l'objet ajouté est une chaîne, celle-ci est simplement ajoutée en bout de liste.

ALOGDix puissance x (10^x)Accès : $[^{\leftarrow}] [y^x]$

Drapeaux : -3

1: réel ou complexe



1: réel ou complexe

ALOG correspond à l'antilogarithme décimal, autrement dit, à la fonction 10 puissance x. L'argument de cette fonction peut être réel ou complexe, son résultat pouvant, lui aussi, être réel ou complexe. Si l'argument de ALOG est une expression symbolique 'expr_symb', le résultat renvoyé est 'ALOG(expr_symb)'.

AMORT

Amortissement d'un prêt ou d'un investissement

Accès (HP-48 G/GX) : $[^{\leftarrow}] [7]$ (TVM) (AMOR) Drapeaux : -3, -5, -10, -14
 3:
 2:
 1: nombre de versements

 3: capital
 2: intérêts
 1: solde

A partir du nombre de versements et des variables I%YR, PV, PMT et PYR (variables TVM), AMORT fournit capitaux, intérêts et solde pour un prêt ou un investissement.

AND

Fonction « et » logique

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (LOGIC) (AND) Drapeaux : -3, -5 à -10

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] [NXT] [NXT] (AND)

 2: Entier_binaire
 1: Entier_binaire

 2: Entier_binaire
 1: Entier_binaire

La fonction « et » logique est aussi appelée produit logique. Normalement, la fonction « et » logique prend deux entiers binaires comme argument et renvoie un nouvel entier binaire correspondant aux bits armés communs aux deux entiers binaires donnés en arguments. Les entiers binaires (format #nb où b représente la base) peuvent être représentés en binaire, octal, décimal ou hexadécimal. Le résultat est exprimé à l'aide de la base courante (déterminée avec HEX, DEC, OCT ou BIN). AND peut aussi prendre comme arguments des variables booléennes (vrai/faux) et même des chaînes de caractères, dans ce cas, le résultat est une chaîne générée par comparaison bit à bit des codages des chaînes.





ANIMATE



Affichages successifs d'objets graphiques

Accès (HP-48 G/GX) : [PRG] (GROB) [NXT] (ANIM) Drapeaux : -

```

n+1:  Objet graphique n°n
...
3:    Objet graphique n°2
2:    Objet graphique n°1
1:    Nombre d'objets graph.

```



```

n+1:  Objet graphique n°n
...
3:    Objet graphique n°2
2:    Objet graphique n°1
1:    Nombre d'objets graph.

```

```

n+1:  Objet graphique n°n
...
3:    Objet graphique n°2
2:    Objet graphique n°1
1:    {nb_ob {#x #y} T R}

```



```

n+1:  Objet graphique n°n
...
3:    Objet graphique n°2
2:    Objet graphique n°1
1:    {nb_ob {#x #y} T R}

```

ANIMATE permet d'afficher successivement plusieurs objets graphiques placés dans la pile (il peut s'agir de GROB placés dans la pile ou de variable contenant un objet graphique). Dans le cas le plus simple, on se contente de placer au niveau 1 de la pile le nombre d'objets graphiques à afficher. Cependant, il est aussi possible de placer au niveau 1 de la pile une liste {nb_ob {#x #y} T R} où nb_ob correspond au nombre d'objets graphiques à afficher, {#x #y}, aux coordonnées (en pixels) de l'angle supérieur gauche, T, au temps d'affichage de chaque objet graphique, et R, au nombre de répétitions de la série d'objets graphiques.



APPLY

Association d'arguments évalués à une expression non évaluée

Accès : [⌈] [9] [NXT] (APPLY)

Drapeaux : -

```

2:  {arg_1 arg_2 ... arg_n}
1:  'fct'

```



```

2:
1: 'fct(arg_1 arg_2 ... arg_n)'

```

APPLY associe une liste d'arguments à une fonction. La fonction est une fonction utilisateur dont le nom est placé entre apostrophes au niveau 1 de la pile. Au niveau 2 de la pile est placée une liste d'arguments délimitée par { et }. Les arguments arg_1, arg_2... arg_n peuvent être des expressions symboliques expr_symb_1, expr_symb_2... expr_symb_n. Le rôle de APPLY consiste à placer au niveau 1 de la pile une expression évaluable 'fct(arg_1 arg_2 ... arg_n)' contenant la fonction fct qui prend ici comme arguments les éléments de la liste précédemment placée au niveau 2 de la pile. Lors de l'évaluation de 'fct(arg_1 arg_2 ... arg_n)', les arguments arg_1 à arg_n seront évalués avant fct. APPLY peut aussi être utilisée dans une expression algébrique sous la forme 'APPLY(fct, arg_1, arg_2... arg_n)'.



ARC

Tracé d'un arc ou d'un cercle

Accès (HP-48 G/GX) : [PRG] (PICT) (ARC)

Drapeaux : -17, -18

Accès (HP-48 S/SX) : [PRG] (DSPL) (ARC)

| | | | | |
|----|-----------|---|----|--|
| 4: | {#x #y} | → | 4: | |
| 3: | xr | | 3: | |
| 2: | angle_dep | | 2: | |
| 1: | angle_fin | | 1: | |

ARC trace un arc de cercle dont le centre est défini par {#x #y} qui sont ses coordonnées exprimées en pixels. xr est l'abscisse de l'extrémité du premier rayon (sens trigonométrique) définie sans équivoque par angle_dep qui est l'angle de départ de l'arc de cercle, l'arc étant tracé jusqu'à angle_fin. angle_dep et angle_fin sont exprimés à l'aide de l'unité de mesure d'angle courante. ARC permet aussi de tracer des cercles, ainsi, si l'unité de mesure d'angle est le degré, il suffit d'affecter les valeurs 0 et 360 à angle_dep et angle_fin (respectivement). Avec ARC, vous pouvez aussi utiliser les coordonnées-utilisateur de type (x, y) au lieu de {#x #y}. Pour bien utiliser ARC, ayez le cercle trigonométrique à l'esprit. Notez que le rayon de l'arc est constant.



ARCHIVE



Création d'une sauvegarde de Home et de différents paramètres

Accès (HP-48 G/GX) : [↵] [VAR] [NXT] (ARCHI) Drapeaux : -33, -39

Accès (HP-48 S/SX) : [↵] [VAR] [NXT] [NXT] (ARCHI)

| | | | | |
|----|----------|---|----|--|
| 1: | :n_p:nom | → | 1: | |
| 1: | :ES:nom | → | 1: | |

ARCHIVE crée une sauvegarde (objet-sauvegarde) du contenu du répertoire HOME, des touches redéfinies et de la liste des alarmes. Cette sauvegarde est placée en mémoire vive indépendante, c'est-à-dire non fusionnée avec la mémoire principale. Rappelons que la mémoire vive est segmentée en partitions aussi appelées ports. Il existe trois ports sur HP-48 SX (ports 0, 1 et 2) et 34 ports sur HP-48 GX (ports 0 à 33). Utilisez FREE ou FREE1 pour désolidariser un port de la mémoire centrale et n'utilisez pas le port 0 avec ARC. Si vous donnez :n_p:nom en argument à ARCHIVE, vous obtiendrez sur le port n° n_p une variable (objet-sauvegarde) nommé :n_p:nom. Sur HP-48 G/GX, vous pouvez utiliser :ES:nom comme argument de ARCHIVE. Vous créez alors un objet-sauvegarde directement transmis en binaire via le port série actif et sauvegardé sur le périphérique relié via le port série (PC, etc.).

ARG

Argument d'un nombre complexe

Accès (HP-48 G/GX) : [MTH] [NXT] (CMPL) (ARG) **Drapeaux** : -17, -18

Accès (HP-48 S/SX) : [MTH] (PARTS) (ARG)

| | | | | |
|----|-----------------|---|----|-------------|
| 1: | complexe | → | 1: | réel |
|----|-----------------|---|----|-------------|

A partir d'un nombre complexe $x + i.y$ placé dans la pile sous la forme (x, y) , ARG fournit l'argument θ de ce nombre complexe. θ est un réel, il est exprimé à l'aide de l'unité de mesure d'angle courante. Si l'argument de ARG est un nombre réel x , il est considéré comme un nombre complexe $x + i.0$. Si l'argument de ARG est une expression symbolique 'expr_symb', ARG renvoie une nouvelle expression symbolique 'ARG(expr_symb)'.



→ARRY

Création d'un tableau

Accès (HP-48 G/GX) : [PRG] (TYPE) (→ARR) **Drapeaux** : -

Accès (HP-48 S/SX) : [PRG] (OBJ) (→ARR)

| | | |
|---|---|---|
| $n+1$: réel ou complexe $n^{\circ}1$ n : réel ou complexe $n^{\circ}2$... 2: réel ou complexe $n^{\circ}n$ 1: n | → | 5: 4: 3: 2: 1: [[vecteur]] |
|---|---|---|

| | | |
|--|---|---|
| $n+1$: réel ou complexe $n^{\circ}1,1$ n : réel ou complexe $n^{\circ}1,2$... 2: réel ou complexe $n^{\circ}1,c$ 1: $\{1\ c\}$ | → | 5: 4: 3: 2: 1: [[matrice]] |
|--|---|---|

A partir de nombres réels ou complexes placés dans la pile, →ARRY crée un tableau ou un vecteur. Ce tableau (matrice) ou ce vecteur sera complexe si l'un au moins de ses éléments est un nombre complexe. Pour créer un vecteur contenant n éléments, il suffit d'introduire ces n éléments du premier au dernier et de placer au niveau 1 de la pile le nombre n d'éléments de la pile à réunir en un vecteur. Pour créer un tableau (matrice), on placera au niveau 1 une liste $\{1\ c\}$ où 1 est le nombre de lignes de la matrice, et c , son nombre de colonnes. Si vous souhaitez créer une matrice, introduisez les éléments de chaque ligne dans l'ordre des colonnes. Ainsi, s'il s'agit d'une matrice de 3 lignes et de 2 colonnes, vous introduirez successivement les éléments dont les positions sont (1,1), (1,2), (2,1), (2,2), (3,1) et (3,2).

ARRY →
Séparation des éléments d'un tableau

Accès : à saisir

| | |
|----|-----------|
| 5: | |
| 4: | |
| 3: | |
| 2: | |
| 1: | [vecteur] |

→

| | |
|---------|-------------------------------|
| $n+1$: | réel ou complexe $n^{\circ}1$ |
| n : | réel ou complexe $n^{\circ}2$ |
| ... | |
| 2: | réel ou complexe $n^{\circ}n$ |
| 1: | { n } |

| | |
|----|-------------|
| 5: | |
| 4: | |
| 3: | |
| 2: | |
| 1: | [[matrice]] |

→

| | |
|---------|---------------------------------|
| $n+1$: | réel ou complexe $n^{\circ}1,1$ |
| n : | réel ou complexe $n^{\circ}1,2$ |
| ... | |
| 2: | réel ou complexe $n^{\circ}1,c$ |
| 1: | { $1\ c$ } |

ARRY → sépare les éléments (nombres réels ou complexes) d'un tableau (matrice ou vecteur). Les différents éléments sont renvoyés dans la pile les uns à la suite des autres. Au niveau 1 de la pile se trouve une liste contenant, soit le nombre d'éléments d'un vecteur, soit le nombre de lignes et le nombre de colonnes d'une matrice. ARRY → a le même effet sur un tableau que OBJ →. Voir aussi →ARRY.

ASIN
Arc sinus

Accès : [↵] [SIN]

Drapeaux : -1, -3, -17, -18

| | |
|----|------------------|
| 1: | réel ou complexe |
|----|------------------|

→

| | |
|----|------------------|
| 1: | réel ou complexe |
|----|------------------|

A un sinus donné, ASIN fait correspondre l'angle dont le sinus correspond à cette valeur. Ce résultat constitue la solution principale et utilise l'unité de mesure d'angle courante. Si l'argument a de ASIN n'est pas compris dans l'intervalle $[-1,1]$, il est transformé en un complexe $a + 0.i$ (le résultat est alors complexe). Appliqué à une expression symbolique 'expr_symb', ASIN renvoie 'ASIN(expr_symb)'.

ASINH
Arc sinus hyperbolique

Accès : [MTH] (HYP) (ASINH)

Drapeaux : -1, -3,

| | |
|----|------------------|
| 1: | réel ou complexe |
|----|------------------|

→

| | |
|----|------------------|
| 1: | réel ou complexe |
|----|------------------|

ASINH n'est l'inverse de SINH que sur une partie de son domaine. Si l'argument a est réel et appartient à l'intervalle $[-1,1]$, il sera automatiquement remplacé par le complexe $a + 0.i$. Voir aussi ASIN et SINH.



ASN

Personnalisation d'une touche du clavier

Accès (HP-48 G/GX) : [←] [CST] (KEYS) (ASN) Drapeaux : -61, -62

Accès (HP-48 S/SX) : [→] [CST] (ASN)

| | | | | |
|----|----------------|---|----|--|
| 2: | obj_à_associer | → | 2: | |
| 1: | N1Nc.B | | 1: | |
| 2: | 'SKEY' | → | 2: | |
| 1: | N1Nc.B | | 1: | |



ASN personnalise une touche du clavier en lui affectant un objet, une pression sur la touche personnalisée entraînant ensuite l'évaluation dudit objet. On place au niveau 2 l'objet à associer (un programme, etc.) et au niveau 1 N1Nc.B qui détermine la touche associée à l'objet ainsi que l'éventuelle touche de bascule nécessaire ([α], [←], [→], etc.). N1 est le numéro de la ligne où se trouve la touche (on compte les lignes de haut en bas), Nc est le numéro de la colonne (on compte les colonnes de gauche à droite). B détermine les touches de bascule nécessaires pour accéder à la fonction personnalisée de la touche (voir tableau).

| Valeur de B | Touches de bascule nécessaires | Valeur de B | Touches de bascule nécessaires |
|-------------|--------------------------------|-------------|--------------------------------|
| 0 | aucune | 4 | [α] |
| 1 | aucune | 5 | [α] [←] |
| 2 | [←] | 6 | [α] [→] |
| 3 | [→] | | |

Par exemple, pour associer le programme ESSAI à la combinaison de touches [←]-[NXT], on placera ESSAI au niveau 2 de la pile et 26.2 au niveau 1 de la pile avant d'activer ASN. Dans le cadre de cet exemple, N1 vaut 2 ([NXT] se trouve sur la deuxième ligne de touches du clavier), Nc vaut 6 ([NXT] se trouve sur la sixième colonne de touches) et B vaut 2 puisque nous voulons associer ESSAI à la combinaison [←]-[NXT]. Si nous avions voulu associer ESSAI à la touche [NXT] et non à la combinaison de touches [←]-[NXT], il aurait suffi de donner à B la valeur 0, NcN1.B devient alors 26.0 ou 26.1. Pour rendre à une touche sa fonction normale, placez 'SKEY' au niveau 2 de la pile puis placez le code NcN1.B de la touche concernée au niveau 1 de la pile. Activez ensuite ASN.



Attention ! Ne confondez pas ASN (personnalisation d'une touche du clavier) et ASIN (Arc sinus). Si les touches redéfinies interdisent une utilisation normale de la HP-48, annulez les redéfinitions de touches en maintenant enfoncée [ON] puis en appuyant sur [C].

ASR**Décalage d'un bit vers la droite (division par 2)**

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BIT) (ASR) **Drapeaux** : -5 à -12

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] [NXT] (ASR)

1: entier binaire



1: entier binaire

Les entiers binaires sont présentés sous forme binaire (#nb), octale (#no), décimale (#nd) ou hexadécimale (#nh) mais sont en fait des mots binaires de 64 bits (ou moins en cas de redéfinition). ASR décale vers la droite les 63 bits de plus faible poids entraînant la perte du bit de plus faible poids. Le bit de plus fort poids n'est pas modifié par ASR ce qui permet de l'utiliser comme bit de signe. On remarquera qu'un décalage d'un bit vers la droite correspond à une division entière par deux de l'entier binaire (sur ses 63 bits de poids faible).

**ATAN****Arc tangente**

Accès : [\leftarrow] [TAN]

Drapeaux : -1, -3, -17, -18

1: réel ou complexe



1: réel ou complexe

ATAN renvoie l'angle (compris entre -90° et 90°) auquel correspond la valeur réelle de la fonction tangente fournie en argument. La solution renvoyer est la solution principale, en effet, il existe une infinité d'angle dont la tangente est égale à une valeur donnée. Si l'argument ATAN est une expression symbolique 'expr_symb', le résultat est une expression symbolique 'ATAN(expr_symb)'.

ATANH**Arc tangente hyperbolique**

Accès : [MTH] (HYP) (ATAN)

Drapeaux : -1, -3, -22

1: réel ou complexe



1: réel ou complexe

ATANH correspond à arc tangente hyperbolique, autrement dit, à la tangente hyperbolique inverse. Attention ! ATANH n'est l'inverse de TANH que sur une partie de son domaine de définition. Soit a , l'argument réel de ATANH. Si $|a|=1$, un résultat infini est signalé (message et drapeau 22). Si $|a|>1$, l'argument réel a est transformé en argument complexe $a + i.0$. Si l'argument ATANH est une expression symbolique 'expr_symb', le résultat est une expression symbolique 'ATANH(expr_symb)'.



ATICK



Intervalles entre les graduations des axes du repère

Accès (HP-48 G/GX) : [←] [8] (PPAR) [NXT] (ATICK) Drapeaux : -

| | | | |
|----|---------|---|----|
| 1: | a | → | 1: |
| 1: | { a b } | → | 1: |



ATICK détermine les intervalles entre deux graduations des axes du repère. a et b étant des réels, si l'argument de ATICK est a , un intervalle équivalent à a unités séparera les graduations des axes. Si l'argument de ATICK est une liste { a b }, a unités sépareront les graduations de l'axe des abscisses alors que b unités sépareront les graduations de l'axe des ordonnées. Ces paramètres sont incorporés dans la variable réservée PPAR. Vous pouvez aussi exprimer l'intervalle entre deux graduations en pixels (et non en unités). Il suffit pour cela d'utiliser des entiers binaires # ad et # bd à la place des réels a et b .

ATTACH

Association d'une bibliothèque à un répertoire

Accès (HP-48 G/GX) : [←] [2] [NXT] (ATTAC) Drapeaux : -

Accès (HP-48 S/SX) : [←] [VAR] [NXT] (ATTAC)

| | | | |
|----|-------------------|---|----|
| 1: | Num_bib | → | 1: |
| 1: | :Num_port:Num_bib | → | 1: |



Une bibliothèque (library) doit, pour être utilisable, être associée à un répertoire. ATTACH associe au répertoire courant la bibliothèque de numéro Num_lib. Si vous indiquez un numéro de partition (port) Num_port, celui-ci sera ignoré. Vous pouvez associer un nombre illimité de bibliothèques au répertoire-racine (HOME) mais vous ne pouvez associer qu'une seule bibliothèque à un autre répertoire.

AUTO

Calcul automatique des valeurs extrêmes des axes du repère

Accès (HP-48 G/GX) : [←] [8] [NXT] (AUTO) Drapeaux : -

Accès (HP-48 S/SX) : [←] [8] (AUTO)



AUTO détermine les valeurs extrêmes des axes du repère. AUTO n'a d'effet que sur l'axe des ordonnées en mode *Function*, mais ajuste les deux axes pour les modes *Histogram*, *Scatter*, *Bar*, *Parametric* et *Polar*. En mode *Conic*, les deux plages sont rendues équivalentes.

AXES

Choix des paramètres relatifs aux axes du repère

Accès (HP-48 G/GX) : [↵] [8] (PPAR) [NXT] (AXES) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [8] [NXT] [NXT] (AXES)

1: (x_or,y_or)



1:

1: {(x_or,y_or) g "x" "y"}



1:

AXES détermine le point d'intersection des deux axes du repère, les graduations des axes ainsi que les noms des axes du repère. (x_or,y_or) sont les coordonnées du point d'intersection des deux axes du repère. Normalement, (x_or,y_or) correspond à (0,0), autrement dit, les axes se croisent à l'origine.

Dans le cas le plus simple, AXES prend (x_or,y_or) comme argument mais vous pouvez aussi utiliser la liste {(x_or,y_or) g "x" "y"} comme argument. "x" et "y" sont alors, respectivement, le nom de l'axe des abscisses et le nom de l'axe des ordonnées. Vous pouvez personnaliser ces chaînes de caractères et placer entre les guillemets le nom de votre choix. g détermine l'intervalle séparant les graduations sur les axes. g peut prendre les mêmes formes que les paramètres de ATICK. g correspond donc à a ou à {a b}. a et b étant des réels, si g correspond à a, un intervalle équivalent à a unités séparera les graduations des axes. Si g est une liste { a b }, a unités sépareront les graduations de l'axe des abscisses alors que b unités sépareront les graduations de l'axe des ordonnées.



Vous pouvez aussi exprimer l'intervalle entre deux graduations en pixels (et non en unités). Il suffit pour cela d'utiliser des entiers binaires #ad et #bd à la place des réels a et b. Ces paramètres sont incorporés dans la variable PPAR.

BAR

Activation du mode de tracé « BAR »

Accès (HP-48 G/GX) : [↵] [8] [NXT] (STAT) (PTYPE) (BAR) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [8] [NXT] (PTYPE) (BAR)

BAR active le mode de tracé « BAR » (histogramme). L'histogramme est ensuite construit avec BARPLOT à partir d'une colonne (déterminée avec XCOL) de la matrice statistique courante.

BARPLOT

Tracé d'un histogramme

Accès (HP-48 G/GX) : [←] [5] (PLOT) (BARPL) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] (BARPL)

Le mode « BAR » étant activé (voir BAR), BARPLOT trace un histogramme fondé sur l'une des colonnes (déterminée par XCOL) de la matrice statistique courante (Σ DAT).



Les valeurs extrêmes des axes du repère sont déterminées automatiquement (mode de tracé « BAR »). Si certains éléments de la colonne de la matrice servant de base au tracé sont négatifs, certaines colonnes seront tracées « au-dessous » de l'axe des abscisses.



Au sein d'un programme RPL, FREEZE doit être exécuté immédiatement après BARPLOT pour que le tracé soit conservé.

BAUD

Définition de la vitesse de transfert (câble série ou infrarouge)

Accès (HP-48 G/GX) : [←] [1] (IOPAR) (BAUD) Drapeaux : -

Accès (HP-48 S/SX) : [←] [PRG] (SETUP) (BAUD)

1: → 1:

Bien que le baud soit une unité quantifiant la modulation et non le débit d'un canal de données, BAUD est l'instruction qui détermine la vitesse de transfert par le câble série ou le port infrarouge. vitesse est un réel valant 1200, 2400, 4800 ou 9600 bits par seconde.

BEEP

Emission d'un son

Accès (HP-48 G/GX) : [PRG] [NXT] (OUT) [NXT] (BEEP) Drapeaux : -56

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] [NXT] (BEEP)

2: → 2:
1: → 1:

BEEP produit un son (vibreur) dont la durée et la fréquence (limitée à 4,4 kHz) sont déterminées à l'aide des nombres réels Fréquence et Durée placées respectivement aux niveaux 2 et 1 de la pile. Faites appel à des listes pour jouer des morceaux de musique...



BESTFIT

Choix automatique du meilleur modèle de régression

Accès (HP-48 G/GX) : [←] [5] (ΣPAR) (MODL) (BESTF) **Drapeaux** : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] (MODL) (BEST)



BESTFIT, permet choisit le modèle de régression le mieux adapté parmi les quatre disponibles pour LR. Le modèle choisi est celui dont le coefficient de corrélation est le plus faible.

BIN

Représentation binaire des entiers binaires

Accès : [MTH] (BASE) (BIN)

Drapeaux : -5 à -12

Les entiers binaires sont codés par des mots de 64 bits, ils peuvent être représentés en binaire, octal, décimal ou hexadécimal. Après exécution de BIN, les entiers binaires sont représentés en binaire (base 2).

BINS

Tri par blocs (tri par intervalles)

Accès (HP-48 G/GX) : [←] [5] (1VAR) (BINS)

Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] (BINS)



Dans la colonne XCOL (variable indépendante) de la matrice statistique (ΣDAT), BINS quantifie le nombre d'éléments se trouvant dans chaque intervalle. Les intervalles sont déterminés par Dep, Larg_interv et n. La borne inférieure du premier intervalle est fixée par le réel Dep qui ne correspond pas forcément à la plus petite valeur de la variable. La largeur d'un intervalle est déterminée par Larg_interv. Ainsi, le premier intervalle correspond à [Dep, Dep+Larg_interv[. Un élément valant Dep+Larg_interv appartient donc au second intervalle et non au premier. Le nombre d'intervalles est déterminé par l'entier n. BINS renvoie une matrice et un vecteur. La matrice [[Nb_i1] [Nb_i2]...[Nb_in]] contient n tableaux de 1 élément, chacun de ces n tableaux contenant le nombre de valeurs de la variable indépendante appartenant à l'intervalle de même rang que le tableau correspondant. Le vecteur [Inf Sup] contient le nombre de valeurs de la variable qui sont inférieures (Inf) et supérieures (Sup) à tous les intervalles.

BLANK

Création d'un objet graphique « blanc »

Accès (HP-48 G/GX) : [PRG] (GROB) (BLAN) **Drapeaux** : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] (BLAN)

| | | | | |
|----|----------|---|----|-----------------|
| 2: | #Largeur | → | 2: | |
| 1: | #Hauteur | | 1: | Objet_graphique |



BLANK crée un objet graphique « blanc ». Il vous suffit de préciser la largeur et la hauteur du nouvel objet graphique à l'aide de #Largeur et de #hauteur qui doivent être des entiers binaires.

BOX

Dessin d'un rectangle

Accès (HP-48 G/GX) : [PRG] (PICT) (BOX) **Drapeaux** : -

Accès (HP-48 S/SX) : [PRG] (DSPL) (BOX)

| | | | | |
|----|-------------|---|----|--|
| 2: | (x_a, y_a) | → | 2: | |
| 1: | (x_b, y_b) | | 1: | |
| 2: | {#x_a #y_a} | → | 2: | |
| 1: | {#x_b #y_b} | | 1: | |



Sur l'écran de l'environnement graphique, BOX trace un rectangle dont les côtés sont parallèles à ceux de l'écran. Il vous suffit de préciser les coordonnées des extrémités de l'une des diagonales du rectangle. Saisissez les coordonnées (x_a, y_a) et (x_b, y_b) comme des nombres complexes si vous utilisez les coordonnées-utilisateur. Saisissez les coordonnées sous forme de listes si celles-ci sont exprimées en pixels.

B→R

Conversion d'un entier binaire en nombre réel

Accès (HP-48 G/GX) : [MTH] (BASE) (B→R) **Drapeaux** : -5 à -12

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] (B→R)

| | | | | |
|----|----------------|---|----|------|
| 1: | entier_binaire | → | 1: | réel |
|----|----------------|---|----|------|



Les entiers binaires sont codés sur 64 bits et peuvent être représentés en binaire, octal, décimal ou hexadécimal. B→R convertit un entier binaire (quelle que soit la base utilisée pour sa représentation) en réel (toujours exprimé en base 10). Si l'entier binaire excède 10^{12} , seul les douze chiffres décimaux les plus significatifs seront retenus.

BUFLEN

Nombre de caractères dans la mémoire tampon du port série

Accès (HP-48 G/GX) : [←] [1] [NXT] (SERIA) (BUFLE)

Drapeaux : -

Accès (HP-48 S/SX) : [←] [PRG] [NXT] [NXT] (BUFLE)

| | | | | |
|----|--|---|----|-------------------|
| 2: | | → | 2: | Nb_car |
| 1: | | | 1: | Drapeau_intégrité |



BUFLEN renvoie dans la pile le nombre Nb_car de caractères stockés dans la mémoire tampon (buffer) du port série (ou port infrarouge). Nb_car est un entier. Drapeau_intégrité vaut 1 si aucune erreur de transmission n'a été détectée, par contre, Drapeau_intégrité vaut 0 si des données reçues ont été perdues à la suite de la saturation de la mémoire tampon ou si une erreur de transmission a été détectée.

BYTES

Taille et somme de contrôle (CRC) d'un objet

Accès : [←] [VAR] (BYTES)

Drapeaux : -

| | | | | |
|----|-------|---|----|-------------------|
| 2: | | → | 2: | Somme_de_contrôle |
| 1: | Objet | | 1: | Taille |



Placez, sans l'évaluer, un objet dans la pile, BYTES renvoie sa somme de contrôle et sa taille en octets. On obtient pas toujours un nombre entier d'octets car les données sont codées grâce à des quartets par la HP-48. Ne confondez pas un objet et son nom ! Si l'argument est un objet intégré, sa somme de contrôle est nulle et sa taille correspond à 2,5 octets, autrement à la taille de son adresse en mémoire (codée sur 5 quartets).

CASE

Structure de test CASE... THEN... END

Accès (HP-48 G/GX) : [PRG] (BRCH) (CASE) (CASE)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) (CASE)

CASE est l'en-tête d'une structure de test CASE... THEN... END de la forme :

```

CASE
  test_1 THEN à_exécuter_si_test_1_vrai END
...
  test_n THEN à_exécuter_si_test_n_vrai END
à_exécuter_si_tous_les_tests_sont_faux
END
    
```

Voir les structures de test dans les chapitres d'initiation au RPL

CEIL

Entier immédiatement supérieur ou égal à un réel

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] [NXT] (CEIL) **Drapeaux** : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] (CEIL)

| | | | | |
|----|------|---|----|--------|
| 1: | Réel | → | 1: | entier |
|----|------|---|----|--------|

CEIL renvoie l'entier immédiatement supérieur ou égal au réel donné en argument. Ainsi, 1,54 CEIL renvoie 2. Notons que l'argument peut aussi être un objet-unité, en ce cas, c'est le nombre réel entrant dans la composition de l'objet-unité qui est modifié par CEIL. Si l'argument est une expression symbolique 'expr_symb', CEIL renvoie une nouvelle expression symbolique 'CEIL(expr_symb)'.

CENTR

Coordonnées du centre de l'écran graphique

Accès (HP-48 G/GX) : [↵] [8] (PPAR) [NXT] (CENT) **Drapeaux** : -

Accès (HP-48 S/SX) : [↵] [8] [NXT] (CENT)

| | | | | |
|----|-------------|---|----|--|
| 1: | (absc, ord) | → | 1: | |
|----|-------------|---|----|--|

| | | | | |
|----|------|---|----|--|
| 1: | absc | → | 1: | |
|----|------|---|----|--|



absc et ord étant des réels, CENTR fait en sorte que le pixel central de l'écran graphique ait les coordonnées (absc, ord). Si seul absc est précisé, la valeur par défaut de ord est zéro. CENTR modifie les données xmin, xmax, ymin et ymax stockés dans la variable réservée PPAR.

CF

Désarmement d'un indicateur (drapeau)

Accès : [PRG] (TEST) [NXT] [NXT] (CF) **Drapeaux** : -

| | | | | |
|----|--------|---|----|--|
| 1: | n_flag | → | 1: | |
|----|--------|---|----|--|



Les indicateurs, aussi appelés drapeaux ou flags peuvent être armés (et donc, actifs) ou désarmés (et donc, inactifs). Les indicateurs sont en fait des bits et peuvent donc prendre deux valeurs : 0 et 1.

CF désarme un indicateur donné, autrement dit, CF désactive un indicateur. Le numéro n_flag de l'indicateur à désactiver doit être placé dans la pile. n_flag est obligatoirement un entier et doit impérativement être compris dans l'intervalle $[-64, -1] \cup [1, 64]$ car il y a 128 indicateurs.

%CH

Différence exprimée en pourcentage

Accès (HP-48 G/GX) : [MTH] (REAL) (%CH) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] (%CH)

| | | | | |
|----|------|---|----|------|
| 2: | réel | → | 2: | réel |
| 1: | réel | | 1: | réel |

%CH exprime en pourcentage la différence entre le contenu du niveau 2 et celui du niveau 1 de la pile. Par exemple, si 25 se trouve au niveau 2 et si 30 se trouve au niveau 1, %CH renvoie 20. En effet $25+20\%$ (autrement dit, $25 \times 1,20$) égale 30. On notera que %CH peut aussi être utilisé avec des objets-unités. Si l'un au moins des argument est une expression symbolique 'expr_symb', le résultat est une expression symbolique '%CH(expr_symb, autre_argument)'.

CHOOSE



Création d'un menu vertical

Accès (HP-48 G/GX) : [PRG] [NXT] (IN) (CHOOS) Drapeaux : -

| | | | | |
|----|-------------------------|---|----|------------|
| 3: | "ch_titre" | → | 3: | obj_select |
| 2: | {{a_1 o_1}...{a_n o_n}} | | 2: | 1 |
| 1: | art_def | | 1: | |

| | | | | |
|----|-------------------------|---|----|---|
| 3: | "ch_titre" | → | 3: | |
| 2: | {{a_1 o_1}...{a_n o_n}} | | 2: | 0 |
| 1: | art_def | | 1: | |



CHOOSE affiche un menu vertical (une liste d'articles). Le choix d'un article doit être validé par la sélection de (OK) dans le menu horizontal (en bas de l'écran), ou être annulé avec (CANCL). (OK) et (CANCL) apparaissent automatiquement en bas de l'écran. "Ch_titre" est une courte chaîne de caractères correspondant au titre du menu. Dans le cas le plus simple, les articles du menu peuvent être présentés sous la forme d'une liste d'objets {o_1 o_2 ... o_n}. Dans ce cas, l'objet apparaît dans le menu et il est, en cas de sélection, renvoyé au niveau 2 de la pile (obj_select). Il est aussi possible d'associer un titre à chaque objet, en ce cas, on utilise une liste de listes {{a_1 o_1} {a_2 o_2} ... {a_n o_n}}. Avec ce second type de liste, l'article a_n (a_n étant une chaîne de caractères) apparaît dans le menu vertical et sa sélection entraîne le renvoi de l'objet o_n (un programme RPL, le plus souvent) au niveau 2. art_def est la position de l'article par défaut (en générale art_def vaut 1). Si art_def est nul, aucun article ne peut être sélectionné et 0 est renvoyé comme si (CANCL) avait été pressée.

CHR

Caractère correspondant à un code donné

Accès (HP-48 G/GX) : [PRG] (TYPE) [NXT] (CHR) **Drapeaux : -**

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] (CHR)

1: → 1:



CHR prend en argument un entier compris entre 0 et 255 et renvoie le symbole correspondant dans une chaîne de caractères ne comportant qu'un seul caractère. Les codes 0 à 30 correspondent à des caractères de contrôle et sont représentés par des petits carrés. Le code 0 correspond au caractère nul (utilisé en fin de ligne de commande). Dans une chaîne de caractères, ce code ne peut pas être modifié. Les codes 0 à 127 correspondent à l'ASCII standard alors que les codes 128 à 255 sont propres à la HP-48.

CKSM

Définition du type de contrôle utilisé pour la transmission série

Accès (HP-48 G/GX) : [←] [1] (IOPAR) (CKSM) **Drapeaux : -**

Accès (HP-48 S/SX) : [←] [PRG] (IOPAR) (CKSM)

1: → 1:



CKSM détermine le type de contrôle utilisé pour détecter les erreurs de transmission lors d'une transmission impliquant le protocole Kermit. Utilisé dans un programme, CKSM prend en argument un entier compris entre 1 et 3 inclus. Avec 1 et 2, on effectue un contrôle arithmétique à 1 ou 2 chiffres. Avec 3, on provoque un contrôle par redondance cyclique sur trois chiffres. Par défaut, l'argument de CKSM est 3.

CLEAR

Suppression du contenu de la pile

Accès (HP-48 G/GX) : [←] [DEL] **Drapeaux : -**

Accès (HP-48 S/SX) : [←] [⇐]

→ 1:



CLEAR supprime le contenu de la pile et rend une pile vide. Sur HP-48 G/GX et hors d'un programme, le contenu de la pile est récupérable avec [←] [EVAL] (fonction d'annulation UNDO). CLEAR ne supprime pas une variable dont le nom se trouve dans la pile.

CLKADJ**Ajustement de l'horloge**

Accès (HP-48 G/GX) : [↵] [4] [NXT] [NXT] (CLKA)

Drapeaux : -

Accès (HP-48 S/SX) : [↵] [4] (ADJST) [NXT] (CLKA)

1:

nb_tops



1:



nb_tops est un entier relatif exprimant en 8192^e de seconde la variation de l'heure. Ainsi, pour faire « avancer » l'horloge de 2 secondes, nb_tops vaut 16384. Bien entendu, pour retarder l'horloge nb_tops peut être négatif.

CLLCD**Supprime le contenu de l'écran affichant la pile**

Accès (HP-48 G/GX) : [PRG] [NXT] (OUT) (CLLCD)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] [NXT] (CLLCD)



CLLCD supprime le contenu de l'affichage de la pile (à opposer à l'affichage graphique « PICT »). Afin de figer l'affichage, CLLCD est souvent associé à FREEZE.

CLOSEIO**Fermeture du port série (câble) et du port infrarouge**

Accès (HP-48 G/GX) : [↵] [1] [NXT] (CLOSE)

Drapeaux : -

Accès (HP-48 S/SX) : [↵] [PRG] [NXT] (CLOSE)



CLOSEIO ferme les ports de communication série (câble et infrarouge). CLOSEIO efface le contenu de la mémoire tampon d'entrée (buffer série) et les messages d'erreur (KERRM).

CLTEACH**Suppression du répertoire EXAMPLES et de son contenu**

Accès (HP-48 G/GX) : [α] [α] [C] [NXT] [COS] [E] [A] [C] [PRG]

Drapeaux : -

CLTEACH supprime le répertoire EXAMPLES et son contenu. Le répertoire EXAMPLES est créé par la commande TEACH et contient différents exemples du didactiel intégré à la HP-48. TEACH et CLTEACH n'existent pas sur HP-48 S/SX. CLTEACH n'est pas accessible *via* un quelconque menu et doit être introduit caractère par caractère.

CLUSR

Suppression de toutes les variables du répertoire courant

Accès : $[\alpha]$ $[\alpha]$ $[C]$ $[NXT]$ $[TAN]$ $[SIN]$ $[\Rightarrow]$ Drapeaux : -



CLUSR efface toutes les variables et tous les répertoires vides placés dans le répertoire courant. CLUSR est équivalent à CLVAR.

CLVAR

Suppression de toutes les variables du répertoire courant

Accès (HP-48 G/GX) : $[\alpha]$ $[\alpha]$ $[C]$ $[NXT]$ $[\sqrt{x}]$ $[A]$ $[\Rightarrow]$ Drapeaux : -

Accès (HP-48 S/SX) : $[\leftarrow]$ $[DEL]$



CLVAR efface toutes les variables et tous les répertoires vides se trouvant dans le répertoire courant.

CL Σ

Suppression des données statistiques

Accès (HP-48 G/GX) : $[\leftarrow]$ $[S]$ $(DATA)$ $(CL\Sigma)$ Drapeaux : -

Accès (HP-48 S/SX) : $[\leftarrow]$ $[S]$ $(CL\Sigma)$



CL Σ supprime les données statistiques placées dans la variable réservée ΣDAT (données statistiques).

CNRM

Norme de colonne d'un tableau

Accès (HP-48 G/GX) : $[MTH]$ $(MATR)$ $(NORM)$ $(CNRM)$ Drapeaux : -

Accès (HP-48 S/SX) : $[MTH]$ $(MATR)$ $[NXT]$ $(CNRM)$

| | | | | |
|----|---------|---------------|----|-----|
| 1: | tableau | \rightarrow | 1: | n_c |
|----|---------|---------------|----|-----|

CNRM (column norm) prend un tableau (vecteur, matrice, etc.) en argument et renvoie la norme de colonne dudit tableau. Considérons les sommes des valeurs absolues des éléments de chacune des colonnes d'un tableau. La norme de colonne du tableau est la plus grande des sommes obtenues. Un vecteur constitue un cas particulier puisqu'il ne comporte qu'une seule colonne. Lorsqu'un tableau contient des éléments complexes, on calcule la somme des modules de ces nombres complexes.

→COL
S/SX

Conversion d'un tableau en colonnes (vecteurs)

Accès (HP-48 G/GX) : [MTH] (MATR) (COL) (→COL) **Drapeaux : -**

| | | |
|---|---|--|
| 4: 3: 2: 1: [vecteur] | → | n: e1_n ... 2: e1_2 1: e1_1 |
| 4: 3: 2: 1: [[matrice]] | → | n: col_n ... 2: col_2 1: col_1 |

→COL décompose un tableau. Si le tableau est une matrice (tableau à deux dimensions), →COL renvoie chaque colonne du tableau sous la forme d'un vecteur. Si le tableau est un vecteur (tableau à une dimension), →COL renvoie les n éléments $e1_1$ à $e1_n$ du tableau. Ces éléments peuvent être des nombres réels ou complexes.



COL+
S/SX

Insertion d'un tableau dans un autre tableau

Accès (HP-48 G/GX) : [MTH] (MATR) (COL) (COL+) **Drapeaux : -**

| | | |
|---|---|---|
| 3: [vecteur] 2: e1 1: col_ins | → | 3: 2: 1: [vecteur] |
| 3: [[matrice]] 2: [vecteur] 1: col_ins | → | 3: 2: 1: [[matrice]] |
| 3: [[matrice]] 2: [[matrice]] 1: col_ins | → | 3: 2: 1: [[matrice]] |

COL+ insère une ou plusieurs colonnes dans une matrice ou insère un élément dans un vecteur. col_ins est le numéro d'ordre de la première colonne devant être refoulée vers la droite pour permettre l'insertion des nouveaux éléments. Dans le cas d'un vecteur [1 2], si l'élément à insérer $e1$ est 3 et si col_ins vaut 1, les éléments du vecteur sont refoulés à partir du premier, COL+ renvoie donc dans ce cas [3 1 2]. Pour insérer une nouvelle colonne dans une matrice, il suffit de placer les éléments de cette nouvelle colonne dans un vecteur, col_ins correspond alors au numéro de la première colonne devant être refoulée vers la droite. COL+ permet aussi d'insérer une matrice dans une autre matrice. →



→ Soit la matrice $\begin{bmatrix} [1 & 2] \\ [3 & 4] \end{bmatrix}$ dans laquelle nous voulons insérer la matrice $\begin{bmatrix} [5 & 6] \\ [7 & 8] \end{bmatrix}$. Si `col_ins` vaut 1, l'insertion commence dès la première colonne et on obtient la matrice $\begin{bmatrix} [5 & 6 & 1 & 2] \\ [7 & 8 & 3 & 4] \end{bmatrix}$. Attention ! Les dimensions des éléments insérés ne sont pas quelconques. Ainsi, une nouvelle colonne doit impérativement comporter autant de lignes que la matrice dans laquelle elle s'insère. La même remarque s'applique à l'insertion d'une matrice dans une autre matrice. Le tableau (vecteur ou matrice) dans lequel doivent être insérés de nouveaux éléments est celui placé au niveau 3 de la pile avant appel de `COL+`.

COL-

5/5X

Suppression d'une colonne d'un tableau (vecteur ou matrice)

Accès (HP-48 G/GX) : `[MTH] (MATR) (COL) (COL-)` Drapeaux : -

2: [vecteur]
1: num_el_suppr



2: [vecteur]
1: el_suppr

2: [[matrice]]
1: num_col_suppr



2: [[matrice]]
1: [vecteur]

`COL-` supprime un élément d'un vecteur ou une colonne d'une matrice. `num_el_suppr` et `num_col_suppr` sont les numéros d'ordre de l'élément ou de la colonne à supprimer (le premier élément d'un vecteur ou la première colonne d'une matrice porte le numéro 1). `COL-` renvoie au niveau 2 le tableau (vecteur ou matrice) modifié et les éléments supprimés au niveau 1.

COL→

5/5X

Création d'un tableau à partir de colonnes

Accès (HP-48 G/GX) : `[MTH] (MATR) (COL) (COL→)` Drapeaux : -

n+1: [vecteur]
...
3: [vecteur]
2: [vecteur]
1: n



5:
4:
3:
2:
1: [[matrice]]

n+1: el_1
...
3: el_2
2: el_n
1: n



5:
4:
3:
2:
1: [vecteur]

`COL→` construit une matrice ou un vecteur à partir de ses éléments →

→ Pour construire une matrice, on place les éléments de chaque colonne dans un vecteur. Le nombre total de colonnes n est placé au niveau 1 de la pile. Le vecteur contenant les éléments de la première colonne est placé au niveau $n+1$, les vecteurs contenant les éléments des autres colonnes étant placés à sa suite. Pour construire un vecteur à partir d'éléments réels ou complexes, on place au niveau 1 le nombre total n d'éléments du vecteur. Au niveau $n+1$ est placé le premier élément du vecteur, les autres éléments étant placés à sa suite.

COLCT

Rassemblement des termes identiques d'une expression symbolique

Accès : [\leftarrow] [9] (COLCT)

Drapeaux : -

1: → 1:



COLCT rassemble les termes identiques d'une expression symbolique et renvoie une expression symbolique simplifiée. Les objets-unités comportant des unités compatibles sont aussi regroupés par COLCT. COLCT agit indépendamment sur chaque membre d'une équation (expression symbolique comportant deux membres séparés par le signe « \Rightarrow »).

COL Σ

Détermination des variables dépendantes et indépendantes

Accès : [α] [α] [C] [EVAL] [NXT] [α] [\rightarrow] [TAN]

Drapeaux : -

2: → 2:
 1: → 1:

Dans la matrice de données statistiques Σ_{DAT} , COL Σ détermine la colonne contenant les valeurs de la variable indépendante et la colonne contenant les valeurs de la variable dépendante. `num_col_indep` est le numéro de la colonne de Σ_{DAT} contenant les valeurs de la variable indépendante alors que `num_col_dep` est le numéro de la colonne de Σ_{DAT} contenant les valeurs de la variable dépendante. COL Σ réunit les fonctions des instructions XCOL et YCOL. COL Σ modifie le contenu de la variable réservée Σ_{PAR} .

COMB

Combinaisons (probabilités)

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) (COMB) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PROB) (COMB)



→

| | | |
|---|---|-------------------------------------|
| 2: nb_total 1: nb_par_comb | → | 2: 1: nb_comb |
|---|---|-------------------------------------|

COMB renvoie le nombre nb_comb de combinaisons possibles à partir d'un nombre total d'éléments nb_total et du nombre d'éléments par combinaison nb_par_comb, on a bien sûr nb_par_comb < nb_total.

CON

Création d'un tableau dont tous les éléments égalent une constante

Accès (HP-48 G/GX) : [MTH] (MATR) (MAKE) (CON) Drapeaux : -

Accès (HP-48 S/SX) : [MTH] (MATR) (CON)

| | | |
|---|---|---|
| 2: {nb_col} 1: cste | → | 2: 1: [vecteur] |
| 2: {nb_ligne nb_col} 1: cste | → | 2: 1: [[matrice]] |
| 2: [[matrice]] 1: cste | → | 2: 1: [[matrice]] |

CON crée un tableau (matrice ou vecteur) dont tous les éléments sont égaux à une constante cste, cette constante pouvant être réelle ou complexe. Les dimensions du nouveau tableau sont placées dans une liste. Celle-ci ne contient que le nombre de colonnes nb_col en cas de création d'un vecteur. Si vous souhaitez créer une matrice, la liste doit non seulement contenir nb_col mais aussi le nombre de lignes nb_ligne. Appliquée à un tableau existant (ou au nom d'un tableau existant), CON remplace tous les éléments de ce tableau par la constante cste. Dans ce cas, cste doit être de même type (réel ou complexe) que les éléments remplacés du tableau donné en argument à CON.

COND

Nombre de condition de la norme de colonne (matrice carrée)

Accès (HP-48 G/GX) : [MTH] (MATR) (NORM) (COND) Drapeaux : -

| | | |
|-----------------------------------|---|-------------------------------|
| 1: [[matrice]] | → | 1: nb_cond |
|-----------------------------------|---|-------------------------------|

Attention ! COND ne s'applique qu'à une matrice carrée. COND prend donc en argument une matrice carrée et renvoie nb_cond qui correspond au nombre de condition. Le nombre de condition d'une matrice carrée est le produit de la norme de ladite matrice avec la norme de colonne de la matrice inverse.



CONIC

Activation du mode de tracé « CONIC »

Accès : [↵] [8] (PTYPE) (CONIC)

Drapeaux : -

CONIC active le mode tracé « CONIC » permettant la représentation graphique de sections coniques. CONIC modifie la variable réservée PPAR qui spécifie alors le type de tracé « CONIC ». L'effet de l'instruction DRAW ainsi que le rôle de la variable réservée EQ sont aussi modifiés. EQ contient alors l'équation de la section conique (polynôme du second degré où interviennent deux variables réelles).

CONJ

Conjugué d'un nombre complexe ou d'un tableau de complexes

Accès (HP-48 G/GX) : [MTH] [NXT] (CML) [NXT] (CONJ) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) (CONJ)

| | | | | |
|----|-------------|---|----|-------------|
| 1: | complexe | → | 1: | complexe |
| 1: | [[vecteur]] | → | 1: | [[vecteur]] |
| 1: | [[matrice]] | → | 1: | [[matrice]] |

Si l'argument de CONJ est un nombre complexe $a+ib$ saisi sous la forme (a, b) , CONJ renvoie son conjugué $a-ib$ sous la forme $(a, -b)$.

Si l'argument de CONJ est un tableau (il peut s'agir d'un vecteur ou d'une matrice) contenant des nombres complexes, CONJ renvoie un tableau de complexes dont tous les éléments ont été remplacés par leurs conjugués.

Appliquée à une expression symbolique 'expr_symb', l'instruction CONJ renvoie l'expression symbolique 'CONJ(expr_symb)'. Appliquée à un nombre réel ou à un tableau de réels, l'instruction CONJ renvoie comme résultat son argument réel non modifié.

CONLIB

Accès à la bibliothèque de constantes



Accès (HP-48 G/GX) : [↵] [3] (COLIB) (CONLI)

Drapeaux : -

CONLIB n'est disponible que sur HP-48 G et GX et donne accès à la bibliothèque de constantes. L'instruction CONST permet d'évaluer la valeur d'une constante.



CONST



Valeur d'une constante de la bibliothèque de constantes

Accès (HP-48 G/GX) : [←] [3] (COLIB) (CONS) Drapeaux : -60, -61

1: [] 'nom' → 1: [] objet_unité

Sur HP-48 G/GX, dans la bibliothèque de constantes ouverte par CONLIB (voir CONLIB), CONST renvoie la valeur d'une constante dont le nom 'nom' est donné en argument. L'argument de CONST est donc un nom placé entre apostrophes alors que son résultat est un objet-unité.

CONT

Poursuite de l'exécution d'un programme interrompu

Accès : [←] [ON] Drapeaux : -

CONT relance l'exécution d'un programme interrompu (voir les instructions PROMPT, HALT et KILL).

CONVERT

Conversion de l'unité d'un objet unité

Accès (HP-48 G/GX) : [←] [6] (CONV) Drapeaux : -

Accès (HP-48 S/SX) : [→] [6] (CONV)

2: [] objet_unité → 2: []
1: [] objet_unité 1: [] objet_unité

Placez au niveau 2 un objet-unité dont l'unité doit être convertie en l'unité de l'objet-unité se trouvant au niveau 1. L'objet-unité renvoyé par CONVERT utilise donc l'unité de l'objet-unité qui se trouvait au niveau 1. La partie réelle (numérique) de l'objet-unité placé au niveau 1 est ignorée. Bien entendu, les unités des deux objets-unités donnés en argument doivent être compatibles.



CORR

Coefficient de corrélation de la matrice statistique

Accès (HP-48 G/GX) : [←] [5] (FIT) (CORR) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] (CORR)

1: [] → 1: [] coeff_correl

CORR renvoie le coefficient de corrélation de la matrice statistique.

COS
Cosinus

Accès : [COS] Drapeaux : -3, -17, -18

| | | | | |
|----|-------------|---|----|----------|
| 1: | réel | → | 1: | réel |
| 1: | objet-unité | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |

Par défaut, la HP-48 considère que l'argument (l'angle) est exprimé à l'aide de l'unité de mesure d'angles courante. Cependant, il est aussi possible de donner comme argument à COS un objet-unité dont la partie unité spécifie l'unité de mesure d'angle utilisée. Si l'argument est une expression symbolique 'expr_symb', le résultat une expression symbolique 'COS(expr_symb)'. Attention ! Le radian est l'unité de mesure d'angle par défaut à activer si vous souhaitez travailler à partir d'expressions symboliques contenant des fonctions circulaires.



COSH
Cosinus hyperbolique

Accès : [MTH] (HYP) (COSH) Drapeaux : -3

| | | | | |
|----|------------------|---|----|------------------|
| 1: | réel ou complexe | → | 1: | réel ou complexe |
|----|------------------|---|----|------------------|

L'instruction COSH calcule le cosinus hyperbolique de son argument qui peut être réel ou complexe. Appliquée à une expression symbolique 'expr_symb', COSH renvoie une nouvelle expression symbolique 'COSH(expr_symb)'.

COV
Covariance

Accès (HP-48 G/GX) : [↵] [5] (FIT) (COV) Drapeaux : -
 Accès (HP-48 S/SX) : [↵] [5] [NXT] [NXT] [NXT] (COV)

| | | | | |
|----|--|---|----|-----|
| 1: | | → | 1: | cov |
|----|--|---|----|-----|

COV calcule la covariance cov et la place dans la pile. La covariance est calculée à partir des données de la matrice Σ_{DAT} de données statistiques, et, plus précisément, à partir de la colonne contenant les valeurs de la variable indépendante (par défaut, colonne 1) et de la colonne contenant les valeurs de la variable dépendante (par défaut, colonne 2). Il est possible de spécifier d'autres colonnes pour les valeurs des variables dépendantes et indépendantes (voir COLΣ, XCOL et YCOL).

C→PX

Conversion de coordonnées-utilisateur en coordonnées-pixels

Accès (HP-48 G/GX) : [PRG] (PICT) [NXT] (C→PX) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] (C→PX)

| | | |
|--------------------------------------|---|-----------------------------------|
| 1: (absc, ordo) | → | 1: {#absc_eb #ordo_eb} |
|--------------------------------------|---|-----------------------------------|



C→PX convertit des coordonnées-utilisateur (coordonnées dans le repère défini par l'utilisateur) en coordonnées exprimées en pixels sur l'écran graphique (dont les dimensions par défaut sont 131×64). Les coordonnées-utilisateurs sont introduites comme un nombre complexe sous la forme d'un couple de réels (absc, ordo) où absc et ordo représentent respectivement l'abscisse et l'ordonnée. Le résultat est une liste de deux entiers binaires #absc_eb et #ordo_eb qui correspondent respectivement à l'abscisse et à l'ordonnée exprimées en pixels.

CR

Impression du contenu du tampon d'impression

Accès (HP-48 G/GX) : [←] [1] (PRINT) (CR) Drapeaux : -33, -34, -37

Accès (HP-48 S/SX) : [←] [MTH] (CR)



Avec l'imprimante infrarouge HP 82240, l'impression de chaque ligne s'achève normalement par un passage automatique à la ligne suivante (CR/LF). Si le drapeau 38 (ajout de passage à la ligne) est armé, il est nécessaire d'exécuter CR pour imprimer les données accumulées dans le tampon d'impression et passer à la ligne suivante.

C→R

Eclatement d'un complexe en deux réels

Accès (HP-48 G/GX) : [MTH] [NXT] (CMPL) (C→R) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] (C→R)

| | | |
|----------------------------------|---|------------------------------|
| 2: complexe | → | 2: réel |
| 1: | | 1: réel |

| | | |
|-----------------------------------|---|-----------------------------------|
| 2: [tableau] | → | 2: [tableau] |
| 1: [tableau] | | 1: [tableau] |



C→R sépare un nombre complexe en deux réels, l'un correspond à la partie réelle (renvoyée au niveau 2), l'autre, à la partie imaginaire (renvoyée au niveau 1). Si l'argument de C→R est un tableau (vecteur ou matrice) dont les éléments sont des nombres complexes, →

→ $C \rightarrow R$ renvoie deux tableaux de réels ayant les mêmes dimensions que le tableau donné en argument. Les éléments du tableau renvoyé au niveau 2 sont les parties réelles des complexes du tableau donné en argument à $C \rightarrow R$. De même le tableau renvoyé au niveau 1 contient les parties imaginaires des éléments du tableau donné en argument.

CRDIR

Création d'un répertoire

Accès (HP-48 G/GX) : [\leftarrow] [VAR] (DIR) (CRDIR)

Drapeaux : -

Accès (HP-48 S/SX) : [\leftarrow] [VAR] (CRDIR)

1: → 1:

CRDIR crée dans le répertoire courant un nouveau (sous-)répertoire dont le nom est précisé entre apostrophes au niveau 1 de la pile. Lors de sa création, le nouveau répertoire est vide. CRDIR ne vous fait pas changer de répertoire le répertoire courant reste donc celui qui accueille le nouveau (sous-)répertoire.



Rappelons qu'il suffit dévaluer le nom d'un sous-répertoire pour en faire le répertoire courant.

CROSS

Produit vectoriel

Accès : [MTH] (VECTR) (CROSS)

Drapeaux : -

2: → 2:
1: → 1:

CROSS calcule le produit vectoriel des deux vecteurs donnés en arguments. Rappelons que le produit vectoriel de deux vecteurs est un autre vecteur. CROSS s'applique aux vecteurs du plan (deux coordonnées, donc, tableau à deux éléments) et aux vecteurs de l'espace (trois coordonnées, donc, tableau à trois éléments). Le produit $\vec{v}_1 \times \vec{v}_2$ s'obtient en plaçant \vec{v}_1 au niveau 2 et \vec{v}_2 au niveau 1.

Le résultat renvoyé est un vecteur du plan ou un vecteur de l'espace si l'un au moins des vecteurs donnés en arguments est un vecteur de l'espace. Dans ce cas, le vecteur du plan se voit ajouter une troisième coordonnée nulle et devient un vecteur de l'espace.



Ne confondez pas produit vectoriel (CROSS) et produit scalaire (DOT).



CSWP

S/SX

Permutation de colonnes d'un tableau

Accès : [MTH] (MATR) (COL) (CSWP) Drapeaux : -

| | | |
|---|---|---|
| 3: [vecteur] 2: num_el_1 1: num_el_2 | → | 3: 2: 1: [vecteur] |
| 3: [[matrice]] 2: num_col_1 1: num_col_2 | → | 3: 2: 1: [[matrice]] |

CSWP effectue une permutation (swap) de deux éléments d'un vecteur ou de deux colonnes d'une matrice. Les numéros des éléments ou des colonnes à permuter sont spécifiés aux niveaux 1 et 2 de la pile.

Le premier élément d'un vecteur et la colonne la plus à gauche d'une matrice portent le numéro 1.

CYLIN

S/SX

Activation du mode de coordonnées cylindriques

Accès (HP-48 G/GX) : [MTH] (VECTR) [NXT] (CYLIN) Drapeaux : -15, -16

Dans le cadre de la géométrie analytique en trois dimensions, CYLIN active le mode d'affichage cylindrique pour les coordonnées et entraîne donc l'affichage de l'indicateur RZZ en haut de l'écran. Les trois coordonnées d'un vecteur de l'espace correspondent à [R θ Z] quand le mode cylindrique est actif pour les coordonnées.

DARCY

S/SX

Calcul du facteur de friction Darcy d'un fluide en mouvement

Accès (HP-48 G/GX) : [↵] [3] (UTILS) (DARCY) Drapeaux : -

| | | |
|--|---|--|
| 2: rug_rel 1: nomb_Reynolds | → | 2: 1: Fact_Darcy |
|--|---|--|

Les deux arguments de DARCY sont des nombres réels non nuls (ou des objets-unités). Le premier, rug_rel, correspond à la rugosité relative, autrement dit, à la division de rugosité du canal d'écoulement à section circulaire par le diamètre dudit canal. Le second, nomb_Reynolds, correspond tout simplement au nombre de Reynolds. En fonction de ce second paramètre qui détermine le type d'écoulement, des techniques de calcul différentes sont employées pour obtenir le facteur de friction Darcy Fact_Darcy (équivalent au quadruple du facteur de Fanning).

DATE
Date

Accès (HP-48 G/GX) : [←] [4] (DATE) Drapeaux : -42
 Accès (HP-48 S/SX) : [←] [4] [NXT] (DATE)

1: → 1:

La date affichée dépend du drapeau 42 (format de date actif). Si vous utilisez la notation française habituelle, le 12 novembre 1995 sera représenté par 11.121995 (format JJ.MMAAAA).

→**DATE**
Modification de la date

Accès (HP-48 G/GX) : [←] [4] (→DAT) Drapeaux : -42
 Accès (HP-48 S/SX) : [←] [4] (SET) (→DAT)

1: → 1:

En fonction du drapeau 42, l'argument `date` doit être exprimé à l'aide du format JJ.MMAAAA ou à l'aide du format MM.JJAAAA. Il est aussi possible de n'ajuster que le jour et le mois, `date` prend alors la forme JJ.MM ou la forme MM.JJ.

DATE+
Ajout d'un nombre de jours à une date

Accès : [←] [4] [NXT] (DATE+) Drapeaux : -42

2:
 1: → 2:
 1:

A partir d'une date `date` placée au niveau 2 et d'un nombre de jours positif ou négatif `nb_jours` placé au niveau 1, DATE+ renvoie la nouvelle date calculée à partir de la date donnée en argument à laquelle a été ajouté le nombre de jours `nb_jours`. Ce dernier peut être négatif, en ce cas, la date renvoyée comme résultat est antérieure à la date donnée en argument.

 Les dates-système manipulées par DATE et →DATE doivent impérativement être comprise entre le 1/1/1991 et le 31/12/2090 (l'intervalle de dates est légèrement différent sur HP-48 S/SX). Cependant, DATE+ peut manipuler des dates comprises entre le 15/101582 et le 31/12/9999. Le calcul sur les dates est donc indépendant de la date-système.

DEBUG

Exécution pas à pas d'un programme

Accès (HP-48 G/GX) : [PRG] [NXT] (RUN) (DEBUG) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) (DEBUG)



'nom_prog' est le nom d'une variable contenant un programme RPL. «programme RPL» est un programme RPL placé dans la pile. DEBUG débute l'exécution puis la bloque (exécution pas à pas du programme). Reportez-vous aux instructions NEXT, SST↓ et SST.



DEBUG est une commande d'exécution et non une instruction. DEBUG ne peut donc pas être insérée dans un programme.

DDAYS

Intervalle entre deux dates

Accès : [←] [4] [NXT] (DDAYS) Drapeaux : -42



DDAYS renvoie un entier correspondant au nombre de jours séparant deux dates. L'entier obtenu est positif dans le cas normal où la date placée au niveau 2 de la pile est antérieure à celle placée au niveau 1. Le format usuel pour les dates est JJ.MMAAAA.



L'intervalle de dates toléré est le même que celui utilisé avec DATE+ (voir DATE+).

DEC

Activation du mode décimal pour les entiers binaires

Accès : [MTH] (BASE) (DEC) Drapeaux : -5, -10, -11, -12

DEC active la représentation décimale (base 10) pour les entiers binaires. Les entiers binaires sont représentés en mémoire par des mots binaires de 64 bits (ou moins, en cas de personnalisation). DEC, BIN, OCT et HEX n'affectent pas ce codage interne. Les entiers binaires sont toujours précédés du symbole # et sont suivis d'une lettre indiquant la base de représentation (d, dans le cas de base 10).

DECR

Décrémentation du contenu d'une variable

Accès (HP-48 G/GX) : [\leftarrow] [VAR] (ARITH) (DECR) Drapeaux : -

Accès (HP-48 S/SX) : [\rightarrow] [VAR] (DECR)

| | | | | |
|----|-----------|---|----|------|
| 1: | 'nom_var' | → | 1: | réel |
|----|-----------|---|----|------|

'nom_var' est le nom d'une variable contenant un nombre réel. DECR (décrément) évalue 'nom_var' (place le réel que contient la variable dans la pile), soustrait 1 à ce nombre, puis remplace la nouvelle valeur dans la variable 'nom_var' comme on le ferait avec [STO]. Après son exécution, DECR laisse au niveau 1 de la pile la nouvelle valeur de la variable 'nom_var', il s'agit bien sûr d'un nombre réel.

DEFINE

Création d'une fonction-utilisateur ou d'une variable

Accès : [\leftarrow] [STO] Drapeaux : -3

| | | | | |
|----|--|---|----|--|
| 1: | 'n=expr' | → | 1: | |
| 1: | 'n(n ₁ ,... n _n)=expr(n ₁ ,... n _n)' | → | 1: | |

DEFINE associe à un nom n une expression $expr$. Dans le cas le plus simple, et donc, à partir de 'n=expr', DEFINE affecte à la variable nommée n la valeur obtenue en évaluant $expr$ (Attention à la valeur du drapeau 3 si $expr$ contient une variable symbolique). Si l'argument de DEFINE est 'n(n₁,... n_n)=expr(n₁,... n_n)', DEFINE crée une fonction-utilisateur. Par exemple, avec 'DBLESOMM(A,B)=2*(A+B)', DEFINE crée le programme $\leftarrow A B '2*(A+B)'$ stocké dans la variable nommée DBLESOMM.



DEG

Activation du mode « degrés » (unité de mesure d'angles)

Accès (HP-48 G/GX) : [\leftarrow] [CST] (ANGLE) (DEG) Drapeaux : -17, -18

Accès (HP-48 S/SX) : [\leftarrow] [CST] [NXT] [NXT] (DEG)

DEG fait du degré l'unité de mesure d'angle courante. Après exécution de DEG, et en l'absence de précision, la HP-48 considérera que tous les angles sont exprimés en degrés. Par ailleurs, lorsqu'un angle doit être renvoyé comme résultat, il est exprimé en degrés. DEG désarme les drapeaux 17 et 18.

DELALARM

Annulation d'une sonnerie programmée

Accès (HP-48 G/GX) : [←] [4] (ALRM) (DELAL) Drapeaux : -

Accès (HP-48 S/SX) : [←] [4] (ALRM) [NXT] (DELAL)

1: → 1:

DELALARM supprime de la liste des alarmes l'alarme dont le numéro num_alm est fourni en argument.



Attention ! Si num_alm vaut 0, toute la liste des alarmes système est effacée.

DELAY

Intervalle de temps entre deux envois à l'imprimante

Accès (HP-48 G/GX) : [←] [1] (PRINT) (PRTPA) (DELAY) Drapeaux : -33, -34

Accès (HP-48 S/SX) : [←] [MTH] [NXT] (DELAY)

1: → 1:

DELAY impose un temps d'attente entre l'émission de deux lignes à imprimer vers l'imprimante. Le réel positif placé au niveau 1 correspond au délai exprimé en secondes. En l'absence de précisions, 1,8 secondes séparent deux envois à l'imprimante. Le délai maximal ne doit pas excéder 6,9 secondes. Attention ! Vous risquez de perdre des données à imprimer si vous imposez un délai inférieur à 1,8 secondes car ce temps est celui dont à besoin l'imprimante infrarouge pour imprimer une ligne.



DELKEYS

Annulation des fonctionnalités associées aux touches redéfinies

Accès (HP-48 G/GX) : [←] [CST] (KEYS) (DELK) Drapeaux : -61, -62

Accès (HP-48 S/SX) : [←] [CST] (DELK)

| | | |
|---|---|-------------------------|
| 1: <input type="text" value="0"/> | → | 1: <input type="text"/> |
| 1: <input type="text" value="'S'"/> | → | 1: <input type="text"/> |
| 1: <input type="text" value="NcN1.B"/> | → | 1: <input type="text"/> |
| 1: <input type="text" value="{NcN1.B1 ... NcN1.Bn}"/> | → | 1: <input type="text"/> |

DELKEYS annule les fonctions associées par l'utilisateur à certaines touches du clavier (voir l'instruction ASN). →

→ Si l'argument de DELKEYS est 0, toutes les redéfinitions de touches faites par l'utilisateur sont annulées (annulation du clavier-utilisateur). Au contraire, si l'argument de DELKEYS est 'S', ce sont toutes les affectations standard des touches qui sont désactivées. Seules les touches redéfinies par l'utilisateur sont alors actives. Pour retrouver le clavier standard, il faut avoir recours à la combinaison [ON]-[C].



Bien entendu, il est aussi possible de n'annuler que les redéfinitions de certaines touches faites par l'utilisateur. Dans ce cas on place dans la pile le code $NcN1.B$ de la touche dont la redéfinition est à annuler. On peut aussi placer dans la pile une liste $\{NcN1.B_1 \dots NcN1.B_n\}$ de touches dont les redéfinitions sont à annuler. Reportez-vous à l'instruction ASN pour les explications relatives au codage $NcN1.B$ des touches.

DEPND

Variable dépendante pour les représentations graphiques

Accès (HP-48 G/GX) : [←] [8] (PPAR) (DEPN)

Drapeaux : -

Accès (HP-48 S/SX) : [→] [8] [NXT] (DEPN)

| | | | | |
|----|---------------|---|----|--|
| 1: | 'nom' | → | 1: | |
| 1: | {nom} | → | 1: | |
| 1: | {nom min max} | → | 1: | |
| 2: | min | → | 2: | |
| 1: | max | → | 1: | |



DEPND remplace dans PPAR le nom de la variable indépendante (par défaut (y)) par nom. nom est un nom de variable globale introduit en tant que nom ('nom') ou dans une liste({nom}). Sur HP-48 G/GX et pour les tracés de type TRUTH, il est possible de spécifier les limites min et max de l'intervalle sur lequel varie la variable indépendante.

DEPTH

Nombre de niveaux occupés dans la pile

Accès (HP-48 G/GX) : [←] [▲] (DEPTH)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (STK) (DEPTH)

| | | | | |
|----|--|---|----|--------|
| 1: | | → | 1: | entier |
|----|--|---|----|--------|

DEPTH place dans la pile un entier correspondant au nombre de niveaux occupés dans la pile (nombre d'objets dans la pile).

DET**Calcul du déterminant d'une matrice carrée**Accès (HP-48 G/GX) : [MTH] (MATR) (NORM) [NXT] (DET) **Drapeaux** : -54

Accès (HP-48 S/SX) : [MTH] (MATR) (DET)

1: → 1:

Appliquée à une matrice carrée (autant de lignes que de colonnes), l'instruction **DET** renvoie le déterminant `det` de cette matrice. Le déterminant d'une matrice carrée réelle est un réel. Le déterminant d'une matrice carrée réelle et entière est un entier. Par exemple, le déterminant de la matrice $\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$ est $\text{det} = 2 \times 5 - 4 \times 3 = -2$ (application de la formule $\text{det} = ad - bc$).

DETACH**La bibliothèque spécifiée est détachée de son répertoire**Accès (HP-48 G/GX) : [←] [2] (DETAC) **Drapeaux** : -

Accès (HP-48 S/SX) : [←] [VAR] [NXT] (DETAC)

1: → 1: 1: → 1:

La bibliothèque (library) dont le numéro `num_lib` est placé dans la pile est détachée du répertoire auquel elle était attachée. S'il est spécifié, le numéro de port `num_port` est ignoré. La suppression d'une bibliothèque attachée à Home n'est possible qu'après son détachement.

**→DIAG****Extraction des éléments diagonaux d'une matrice**Accès (HP-48 G/GX) : [MTH] (MATR) [NXT] (→DIAG) **Drapeaux** : -1: → 1:

La fonction **→DIAG** crée un vecteur à partir d'une matrice. Ce vecteur contient les éléments de la première diagonale de la matrice. **→DIAG** peut être appliquée à des matrices carrées ou non. Si la matrice est carrée, le nombre d'éléments du vecteur correspond au nombre de lignes de la matrice. Appliquée à la matrice $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$, **→DIAG** renvoie le vecteur $\begin{bmatrix} 1 \\ 4 \end{bmatrix}$. Lorsque la matrice n'est pas carrée, la plus petite dimension de la matrice détermine le nombre d'éléments du vecteur.

DIAG→

S/SX

Création d'une matrice à partir de sa diagonaleAccès (HP-48 G/GX) : [MTH] (MATR) [NXT] (DIAG→) **Drapeaux** : -

| | | | | |
|----|------------------------|---|----|-------------|
| 2: | [vecteur] | → | 2: | |
| 1: | {nombr_lign nombr_col} | | 1: | [[matrice]] |

DIAG→ crée une matrice à partir d'un vecteur et d'une liste. Le vecteur contient les éléments diagonaux de la matrice alors que la liste contient le nombre de ligne `nombr_lign` et le nombre de colonnes `nombr_col` de la matrice à créer. A partir du vecteur [1 4] et de la liste {2 3}, DIAG→ crée la matrice $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \end{bmatrix}$. Les éléments non définis sont remplacés par des zéros et la matrice peut ne pas être carrée.

**DIFFEQ****Mode graphique consacré aux équations différentielles**Accès (HP-48 G/GX) : [←] [8] (PTYPE) (DIFFE) **Drapeaux** : -

DIFFEQ active le mode graphique permettant la représentation graphique d'équations différentielles. Le contenu de la liste PPAR détermine les paramètres de tracé (intervalles de représentation des axes du repère, variable indépendante, intervalle de valeurs de la variable indépendante, informations relatives à la variable dépendante, etc.). L'équation à représenter est placée dans la variable EQ. Si EQ est une liste, la résolution s'opère en exploitant les algorithmes de Runge-Kutta-Fehlberg et de Rosenbrock. Si EQ n'est pas une liste, seule la méthode de Runge-Kutta-Fehlberg est utilisée.

**DISP****Affichage d'un objet**Accès (HP-48 G/GX) : [PRG] [NXT] (OUT) (DISP) **Drapeaux** : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] [NXT] (DISP) ou [PRG] (CTRL) [NXT] (DISP)

| | | | | |
|----|------------------|---|----|--|
| 2: | objet_à_afficher | → | 2: | |
| 1: | num_ligne | | 1: | |

Placez au niveau 2 de la pile l'objet à afficher (il suffit de placer dans le programme son nom sans apostrophe) puis le numéro `num_ligne` de ligne de l'écran où doit être affiché l'objet. La ligne 1 est la plus haute ligne de l'écran alors que la ligne 7 est la plus basse. FREEZE permet de bloquer l'affichage (et l'exécution) en attendant une pression sur une touche.



DO

Structure de boucle DO... UNTIL... END

Accès (HP-48 G/GX) : [PRG] (BRCH) (DO) (DO) **Drapeaux : -**

Accès (HP-48 S/SX) : [PRG] (BRCH) (DO)



DO constitue l'en-tête d'une boucle indéfinie DO... UNTIL... END de la forme :

```
DO
  instructions de la boucle...
UNTIL
  condition à tester
END
```

La condition à tester peut, par exemple, être 'I>0'.

DOERR

Provoque une erreur pendant l'exécution du programme

Accès (HP-48 G/GX) : [PRG] [NXT] (ERROR) (DOERR) **Drapeaux : -**

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] [NXT] (DOERR)

| | | | | |
|----|---------|---|----|--|
| 1: | 0 | → | 1: | |
| 1: | num_err | → | 1: | |
| 1: | "texte" | → | 1: | |



Si 0 est dans la pile, le programme est quitté. Si num_err ou "texte" sont dans la pile, le programme est non seulement quitté mais en plus, le message "texte" ou le texte associé à l'erreur n° num_err est affiché.

DOLIST



Emploi d'une liste d'arguments pour une fonction ou un programme

Accès (HP-48 G/GX) : [PRG] (LIST) (PROC) (DOLIS) **Drapeaux : -**

| | | | | |
|------|--------------|---|----|--------------|
| n+2: | {liste n° n} | → | 4: | |
| ... | | | 3: | |
| 3: | {liste n° 1} | | 2: | |
| 2: | n | | 1: | selon le cas |
| 1: | Programme | | | |
| n+2: | {liste n° n} | → | 4: | |
| ... | | | 3: | |
| 3: | {liste n° 1} | | 2: | |
| 2: | n | | 1: | selon le cas |
| 1: | Fonction | | | |



DOLIST applique la fonction ou le programme placé au niveau 1 aux n listes placées aux niveaux 3 à $n+2$ de la pile. Si le programme placé au niveau 1 est une fonction-utilisateur, une fonction intégrée ou un programme RPL ne contenant qu'une fonction, alors vous pouvez ne pas préciser n au niveau 2 de la pile. Dans ce cas, les n listes utilisées se placent entre les niveaux 2 et $n+1$ de la pile.



Toutes les listes devant comporter le même nombre m d'élément, le programme ou la fonction placé au niveau 1 sont évalués m fois en prenant simultanément pour arguments tous les éléments de même rang de toutes les listes. Toutes les listes sont donc parcourues simultanément depuis leur élément 1 jusqu'à leur élément m .

DOSUBS

S/SX

Emploi d'une liste d'arguments pour une fonction ou un programme

Accès (HP-48 G/GX) : [PRG] (LIST) (PROC) (DOSUB)

Drapeaux : -

| | |
|----|-------------|
| 3: | {liste_arg} |
| 2: | n |
| 1: | Programme |



| | |
|----|-------------|
| 3: | |
| 2: | |
| 1: | {liste_res} |

| | |
|----|-------------|
| 3: | {liste_arg} |
| 2: | n |
| 1: | Fonction |



| | |
|----|-------------|
| 3: | |
| 2: | |
| 1: | {liste_res} |



DOSUBS utilise le contenu de la liste {liste_arg} comme argument de la fonction ou du programme placé au niveau 1 de la pile. La liste des résultats, {liste_res}, est ensuite introduite dans la pile. Le contenu du niveau 1 de la pile est évalué tant que la liste contient des arguments non encore utilisés. Lors de la première itération, les éléments 1 à n de la liste sont utilisés comme arguments de la fonction ou du programme. Lors de l'itération suivante, c'est au tour des éléments 2 à $n+1$ d'être utilisés. Lors d'une itération quelconque, les éléments de la liste {liste_arg} utilisés comme arguments sont les éléments i à $i+n-1$ avec $i \geq 1$ et $i+n-1$ inférieur ou égal au nombre total d'éléments placés dans la liste. Toujours lors d'une itération quelconque, NSUB renvoie la valeur de i (numéro dans la liste du premier élément du groupe d'éléments actuellement utilisés comme arguments) alors que ENDSUB renvoie le nombre de groupes d'éléments pouvant être créés à partir de la liste {liste_arg}. Si le programme placé au niveau 1 est une fonction-utilisateur, une fonction intégrée ou un programme RPL ne contenant qu'une fonction, alors vous pouvez ne pas préciser n au niveau 2 de la pile. Dans ce cas, {liste_arg} est placée au niveau 2 de la pile.

DOT

Produit scalaire

Accès : [MTH] (VECTR) (DOT)

Drapeaux : -

| | | | | |
|----|-------------|---|----|------|
| 2: | [tableau_1] | → | 2: | |
| 1: | [tableau_2] | | 1: | réel |

DOT renvoie un nombre réel correspondant au produit scalaire de deux tableaux (un tableau est un vecteur ou une matrice). Ces deux tableaux doivent impérativement avoir les mêmes dimensions. Lorsque les tableaux sont des vecteurs, DOT permet de savoir si ces vecteurs ont des directions perpendiculaires. Ainsi, appliquée aux vecteurs $[2 \ 0]$ et $[0 \ 1]$, la fonction DOT renvoie 0, prouvant ainsi l'orthogonalité des deux vecteurs.



D→R

Conversion en radians d'une mesure exprimée en degrés

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] [NXT] (D→R)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (VECTR) [NXT] (D→R)

| | | | | |
|----|-------------|---|----|------------------|
| 1: | réel | → | 1: | réel |
| 1: | 'expr_symb' | → | 1: | 'D→R(expr_symb)' |

D→R multiplie le réel placé dans la pile par $\pi/180$. Ainsi, si le réel donné en argument est une mesure d'angle exprimée en degrés, D→R renvoie cette mesure d'angle exprimée en radians. Si l'argument est une expression symbolique 'expr_symb', D→R renvoie dans le pile l'expression symbolique 'D→R(expr_symb)' non évaluée. Voir aussi R→D.

DRAW

Représentation graphique de EQ ou de ΣDAT

Accès (HP-48 G/GX) : [↵] [8] (DRAW)

Drapeaux : -28, -31

Accès (HP-48 S/SX) : [↵] [8] (DRAW)

En fonction du mode graphique actif, DRAW utilise soit le contenu de la variable réservée EQ (représentation de fonctions cartésiennes, d'équations différentielles, de coniques, de fonctions polaires, etc.), soit le contenu de la variable réservée ΣDAT (représentation graphique de données statistiques). DRAW ne supprime pas le contenu de PICT (utilisez ERASE à cette fin puis utilisez DRAX pour tracer les axes du repère). pour figer l'affichage après le tracé, placez FREEZE à la suite de DRAW.



DRAX

Tracé des axes du repère

Accès (HP-48 G/GX) : [←] [8] (DRAX)

Drapeaux : -

Accès (HP-48 S/SX) : [↵] [8] [NXT] [NXT] (DRAX)

DRAX trace les axes du repère cartésien. Dans le cas général, DRAX est exécuté après ERASE mais avant DRAW (voir DRAW). AXES détermine les coordonnées de l'intersection des axes, ATICK fixe l'intervalle entre les graduations et LABEL nomme les axes. En fait, DRAX ajoute des éléments graphiques à l'objet graphique PICT (variable réservée contenant les représentations graphiques) et peut très bien mettre en place une nouvelle paire d'axes par dessus des axes plus anciens si ERASE n'a pas été activé.

DROP

Suppression du contenu du niveau 1 de la pile

Accès : [←] [+]

Drapeaux : -

| | |
|----|--------------------------|
| 3: | |
| 2: | <i>Pile quelconque</i> |
| 1: | <i>objet au niveau 1</i> |

→

| | |
|----|------------------------|
| 3: | |
| 2: | <i>Pile quelconque</i> |
| 1: | |

| | |
|----|-------------------------|
| 1: | <i>objet quelconque</i> |
|----|-------------------------|

→

| | |
|----|--|
| 1: | |
|----|--|

DROP supprime l'objet placé au niveau 1 de la pile et provoque le décalage du reste de la pile d'un niveau (le niveau 2 passe au niveau 1, le niveau 3 au niveau 2, etc.). Uniquement en mode direct (hors programme) et sur HP-48 G/GX, UNDO peut annuler l'effet de DROP.



DROP2

Suppression du contenu des niveaux 1 et 2 de la pile

Accès (HP-48 G/GX) : [←] [↑] [NXT] (DROP2)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] [STK] [NXT] (DROP2)

| | |
|----|--------------------------|
| 3: | <i>Pile quelconque</i> |
| 2: | <i>objet au niveau 2</i> |
| 1: | <i>objet au niveau 1</i> |

→

| | |
|----|------------------------|
| 3: | |
| 2: | <i>Pile quelconque</i> |
| 1: | |

| | |
|----|-------------------------|
| 2: | <i>objet quelconque</i> |
| 1: | <i>objet quelconque</i> |

→

| | |
|----|--|
| 2: | |
| 1: | |

DROP2 est équivalent à DROP DROP. DROP2 supprime le contenu des niveaux 1 et 2 de la pile et décale donc le niveau 3 vers le niveau 1.



DROPN

Suppression de plusieurs niveaux de la pile.

Accès (HP-48 G/GX) : [\leftarrow] [\uparrow] [NXT] (DRPN) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (STK) [NXT] (DRPN)

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---------------------|---------|-----|--|----|---------|----|---------|----|---|---|---|----|---------------------|----|---------------------|----|---------------------|----|---------------------|----|---------------------|
| <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 10%; padding: 2px;">n+1:</td><td style="padding: 2px;">objet_n</td></tr> <tr><td style="padding: 2px;">...</td><td style="padding: 2px;"></td></tr> <tr><td style="padding: 2px;">3:</td><td style="padding: 2px;">objet_2</td></tr> <tr><td style="padding: 2px;">2:</td><td style="padding: 2px;">objet_1</td></tr> <tr><td style="padding: 2px;">1:</td><td style="padding: 2px;">n</td></tr> </table> | n+1: | objet_n | ... | | 3: | objet_2 | 2: | objet_1 | 1: | n | → | <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 10%; padding: 2px;">5:</td><td style="padding: 2px;">objet du niveau n+6</td></tr> <tr><td style="padding: 2px;">4:</td><td style="padding: 2px;">objet du niveau n+5</td></tr> <tr><td style="padding: 2px;">3:</td><td style="padding: 2px;">objet du niveau n+4</td></tr> <tr><td style="padding: 2px;">2:</td><td style="padding: 2px;">objet du niveau n+3</td></tr> <tr><td style="padding: 2px;">1:</td><td style="padding: 2px;">objet du niveau n+2</td></tr> </table> | 5: | objet du niveau n+6 | 4: | objet du niveau n+5 | 3: | objet du niveau n+4 | 2: | objet du niveau n+3 | 1: | objet du niveau n+2 |
| n+1: | objet_n | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | |
| 3: | objet_2 | | | | | | | | | | | | | | | | | | | | | |
| 2: | objet_1 | | | | | | | | | | | | | | | | | | | | | |
| 1: | n | | | | | | | | | | | | | | | | | | | | | |
| 5: | objet du niveau n+6 | | | | | | | | | | | | | | | | | | | | | |
| 4: | objet du niveau n+5 | | | | | | | | | | | | | | | | | | | | | |
| 3: | objet du niveau n+4 | | | | | | | | | | | | | | | | | | | | | |
| 2: | objet du niveau n+3 | | | | | | | | | | | | | | | | | | | | | |
| 1: | objet du niveau n+2 | | | | | | | | | | | | | | | | | | | | | |

L'entier naturel n étant placé au niveau 1, DROPN supprime les objets placés aux niveaux 2 à $n+1$ et fait donc passer les éventuels objets des niveaux $n+2$ et suivants aux niveaux 1 et suivants.

DTAG

Suppression de l'étiquette d'un objet

Accès (HP-48 G/GX) : [PRG] (TYPE) [NXT] (DTAG) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] (DTAG)

| | | | | | | |
|---|-----------|-----------|---|---|----|-------|
| <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 10%; padding: 2px;">1:</td><td style="padding: 2px;">:id:objet</td></tr> </table> | 1: | :id:objet | → | <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 10%; padding: 2px;">1:</td><td style="padding: 2px;">objet</td></tr> </table> | 1: | objet |
| 1: | :id:objet | | | | | |
| 1: | objet | | | | | |

L'instruction →TAG (voir →TAG) associe un message (une chaîne), un nom ou un réel à un objet. L'objet en question devient alors un objet identifié (objet «taggué»). DTAG supprime le message, le nom ou le réel qui constitue l'étiquette (ou identification, ou tag) de l'objet et rend l'objet au niveau de la pile sans son étiquette (qui est perdue).

DUP

Duplication de l'objet placé au niveau 1 de la pile

Accès (HP-48 G/GX) : [\leftarrow] [\uparrow] [NXT] (DUP) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (STK) [NXT] (DUP)

| | | | | | | | | | | | | | | |
|--|-----------------|--|----|-----------------|----|-------------|---|---|----|-----------------|----|-------------|----|-------------|
| <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 10%; padding: 2px;">3:</td><td style="padding: 2px;"></td></tr> <tr><td style="padding: 2px;">2:</td><td style="padding: 2px;">pile quelconque</td></tr> <tr><td style="padding: 2px;">1:</td><td style="padding: 2px;">objet_niv_1</td></tr> </table> | 3: | | 2: | pile quelconque | 1: | objet_niv_1 | → | <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 10%; padding: 2px;">3:</td><td style="padding: 2px;">pile quelconque</td></tr> <tr><td style="padding: 2px;">2:</td><td style="padding: 2px;">objet_niv_1</td></tr> <tr><td style="padding: 2px;">1:</td><td style="padding: 2px;">objet_niv_1</td></tr> </table> | 3: | pile quelconque | 2: | objet_niv_1 | 1: | objet_niv_1 |
| 3: | | | | | | | | | | | | | | |
| 2: | pile quelconque | | | | | | | | | | | | | |
| 1: | objet_niv_1 | | | | | | | | | | | | | |
| 3: | pile quelconque | | | | | | | | | | | | | |
| 2: | objet_niv_1 | | | | | | | | | | | | | |
| 1: | objet_niv_1 | | | | | | | | | | | | | |

DUP crée une copie de l'objet objet_niv_1 qui se trouve au niveau 1 de la pile et l'insère dans la pile. Immédiatement après avoir exécuté DUP, les niveaux 1 et 2 de la pile contiennent donc deux objets identiques mais indépendants. Le reste de la pile est quelconque.



DUP2

Duplication des niveaux 1 et 2 de la pile

Accès (HP-48 G/GX) : [\leftarrow] [\uparrow] [NXT] (DUP2) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (STK) [NXT] (DUP2)

| | |
|----|-------------|
| 4: | |
| 3: | |
| 2: | objet_niv_2 |
| 1: | objet_niv_1 |



| | |
|----|-------------|
| 4: | objet_niv_2 |
| 3: | objet_niv_1 |
| 2: | objet_niv_2 |
| 1: | objet_niv_1 |

A partir d'une pile contenant l'objet objet_niv_1 au niveau 1 et l'objet objet_niv_2 au niveau 2, DUP2 place l'objet objet_niv_1 aux niveaux 1 et 3 et l'objet objet_niv_2 aux niveaux 2 et 4 de la pile. Le reste de la pile est refoulé.

DUPN

Duplication de n niveaux de la pile

Accès (HP-48 G/GX) : [\leftarrow] [\uparrow] [NXT] (DUPN) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (STK) [NXT] (DUPN)

| | |
|------|---------|
| n+4: | |
| n+3: | |
| n+2: | |
| n+1: | objet_n |
| ... | ... |
| 3: | objet_2 |
| 2: | objet_1 |
| 1: | n |



| | |
|------|---------|
| 2×n: | objet_n |
| ... | ... |
| n+2: | objet_2 |
| n+1: | objet_1 |
| n: | objet_n |
| ... | ... |
| 2: | objet_2 |
| 1: | objet_1 |



n étant un entier naturel non nul, DUPN place aux niveaux n+1 à 2n des copies des objets se trouvant initialement aux niveaux 2 à n+1 de la pile (voir les exemples de pile ci-dessus).

e

Nombre de Népère

Accès : [α] [\leftarrow] [E] [ENTER] Drapeaux : -2, -3

| | |
|----|--|
| 1: | |
|----|--|



| | |
|----|-----|
| 1: | 'e' |
|----|-----|

| | |
|----|--|
| 1: | |
|----|--|



| | |
|----|---------------|
| 1: | 2.71828182846 |
|----|---------------|



[α] [\leftarrow] [E] [ENTER] place 'e' dans la pile. Pour obtenir la forme numérique de la base du logarithme népérien utilisez →NUM.

EGV**Valeurs propres et vecteurs propres de droite**

Accès (HP-48 G/GX) : [MTH] (MATR) [NXT] (EGV)

Drapeaux : -

| | | | | |
|----|-------------|---|----|-------------|
| 2: | | → | 2: | [[matrice]] |
| 1: | [[matrice]] | | 1: | [vecteur] |

EGV prend comme argument une matrice carrée. Les colonnes de la matrice renvoyée au niveau 2 correspondent aux vecteurs propres de droite associés aux éléments du vecteur renvoyé au niveau 1. Les éléments de ce vecteur sont les valeurs propres calculées à partir de la matrice donnée en argument. Si seul ce vecteur de valeurs propres vous intéresse, utilisez plutôt EGVL (voir EGVL).

EGVL**Valeurs propres d'une matrice carrée**

Accès : [MTH] (MATR) [NXT] (EGVL)

Drapeaux : -

| | | | | |
|----|-------------|---|----|-----------|
| 1: | [[matrice]] | → | 1: | [vecteur] |
|----|-------------|---|----|-----------|

EGVL prend comme argument une matrice carrée [[matrice]] et renvoie un vecteur [vecteur] contenant les valeurs propres. Voir aussi EGV.

ELSE**En-tête de l'action à effectuer si la condition n'est pas remplie**

Accès (HP-48 G/GX) : [PRG] (BRCH) (IF) (ELSE)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] [NXT] (ELSE)

ELSE est un élément facultatif d'une structure IF... ou IFERR...

Cas IF :

```
IF condition THEN
  instructions à exécuter si la condition est remplie
ELSE
  instructions à exécuter si la condition n'est pas remplie
END
```

Cas IFERR :

```
IF erreur à intercepter THEN
  instructions à exécuter si une erreur est détectée
ELSE
  instructions à exécuter si aucune erreur n'est détectée
END
```

Voir IF et IFERR.



END

Marque de fin de boucle, de test ou d'interception d'erreur

Accès (HP-48 G/GX) : [PRG] (BRCH) (IF) (END) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] (END)

END est l'indispensable marque de fin des structures commençant par IF (test d'une condition), CASE (tests en série), IFERR (interception d'erreur), DO... WHILE (boucle conditionnelle) et REPEAT... UNTIL (boucle conditionnelle). Notez que END est accessible *via* différents sous-menus de la HP-48 G/GX : [PRG] (BRCH) (CASE) (END), [PRG] (BRCH) (DO) (END), [PRG] [NXT] (ERROR) (IFERR) (END) et [PRG] (BRCH) (WHILE) (END).

ENDSUB

Nombre d'arguments de DOSUBS



Accès (HP-48 G/GX) : [PRG] (LIST) (PROC) (ENDS) Drapeaux : -

1: → 1:

ENDSUB place dans la pile un entier naturel correspondant au nombre de blocs d'arguments (destinés à la fonction appliquée par DOSUBS) qu'il est possible de créer à partir de la liste donnée en argument. Voir aussi DOSUBS.



ENG

Mode d'affichage «ingénierie»

Accès (HP-48 G/GX) : [←] [CST] (FMT) (ENG) Drapeaux : -

Accès (HP-48 S/SX) : [←] [CST] (ENG)

1: → 1:

ENG active le format d'affichage «ingénierie» pour les nombres. Avec ce format, les nombres sont présentés sous la forme d'une mantisse dont la partie entière est :

- non nulle quand le nombre est non nul,
- composée de un à trois chiffres.

Par ailleurs, l'exposant du facteur 10 (par lequel est multipliée la mantisse) est toujours un multiple de 3. Le symbole E sépare la mantisse de l'exposant du facteur 10. ENG prend comme argument un entier nb_signif compris entre 0 et 11 inclus, le nombre de chiffres significatifs affichés étant égal à nb_signif+1.

EQ→

Séparation des deux membres d'une équation

Accès (HP-48 G/GX) : [PRG] (TYPE) [NXT] (EQ→)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) (EQ→)

| | | |
|-------------------------------|---|---------------------------------|
| 2: 1: 'mb_gauche=mb_droit' | → | 2: 'mb_gauche' 1: 'mb_droit' |
| 2: 1: réel | → | 2: réel 1: 0 |
| 2: 1: complexe | → | 2: complexe 1: 0 |
| 2: 1: 'nom' | → | 2: 'nom' 1: 0 |
| 2: 1: objet_unité | → | 2: objet_unité 1: 0 |
| 2: 1: 'expr_symb' | → | 2: 'expr_symb' 1: 0 |

Une équation est une expression symbolique composée de deux membres séparés par le signe «=». EQ→ sépare le membre gauche et le membre d'une équation en deux expressions. Ainsi, à partir de l'équation 'mb_gauche=mb_droit', EQ→ recrée deux expressions symboliques 'mb_gauche' et 'mb_droit'. Quand l'argument de EQ→ est un réel, un complexe, un nom, un objet-unité ou une expression symbolique autre qu'une équation, alors cet objet est décalé dans la pile vers le niveau 2 par l'insertion de 0 au niveau 1.

EQNLIB

Ouverture de EQLIB



Accès (HP-48 G/GX) : [↵] [3] (EQLIB) (EQNLI)

Drapeaux : -

EQNLIB donne accès à la bibliothèque d'équations EQ LIB.

ERASE

Suppression du contenu de l'écran graphique

Accès (HP-48 G/GX) : [↵] [8] (ERASE)

Drapeaux : -

Accès (HP-48 S/SX) : [↵] [8] (ERASE)

ERASE initialise l'écran graphique. La variable réservée PICT est donc remplacée par un objet graphique vierge (écran blanc).



ERRO**Efface le dernier message d'erreur mémorisé**

Accès (HP-48 G/GX) : [PRG] [NXT] (ERROR) (ERRO)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] [NXT] (ERRO)

ERRO efface le dernier message d'erreur mémorisé ainsi que son numéro. Après avoir exécuté l'instruction ERRO, l'instruction ERRN affiche l'entier système #0h. (ERRN rappelle le numéro de la dernière erreur mémorisée.)

ERRM**Rappel du dernier message d'erreur**

Accès (HP-48 G/GX) : [PRG] [NXT] (ERROR) (ERRM)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] [NXT] (ERRM)

| | | |
|----|---|------------|
| 1: | → | 1: message |
|----|---|------------|

ERRM (*error message*) rappelle le dernier message d'erreur et l'affiche au niveau 1 de la pile sous la forme d'une chaîne de caractères (placée entre les délimiteurs «"» et «"»). ERRM peut renvoyer le message associé à l'erreur volontairement déclenchée par l'instruction DOERR (voir aussi DOERR). Dans le cas où DOERR a 0 pour argument, ERRM renvoie une chaîne de caractères vide.

ERRN**Rappel du numéro de la dernière erreur**

Accès (HP-48 G/GX) : [PRG] [NXT] (ERROR) (ERRN)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] [NXT] (ERRN)

| | | |
|----|---|-------------------|
| 1: | → | 1: entier_binaire |
|----|---|-------------------|

ERRN (*error number*) renvoie un entier binaire correspondant au numéro de la dernière erreur. Si DOERR vient d'être utilisé avec un message comme argument, ERRN renvoie #70000h.

EVAL**Evaluation de l'objet placé au niveau 1**

Accès : [EVAL]

Drapeaux : -3

| | | |
|----------|---|-------------|
| 1: objet | → | 1: résultat |
|----------|---|-------------|



→ EVAL (évaluation ≈ exécution) provoque l'évaluation du contenu du niveau 1. Les fonctions et les programmes sont exécutées, les calculs sont effectués, les noms évalués entraîne l'évaluation des objets auxquels ils sont associés, etc. Attention ! Ne confondez pas EVAL et →NUM qui sont des instructions différentes mais complémentaires.

EXP

Exponentielle

Accès : [←] [1/x]

Drapeaux : -3

| | | | | |
|----|-------------|---|----|------------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'EXP(expr_symb)' |



EXP utilise l'argument réel ou complexe comme exposant de e (nombre de Népère). EXP fournit, avec les grandes valeurs de l'argument, un résultat plus précis que la constante e élevée à une puissance donnée à l'aide de la touche [y^x].

EXPAN

Développement d'une expression symbolique

Accès : [←] [9] (EXPA)

Drapeaux : -

| | | | | |
|----|-------------|---|----|-------------|
| 1: | 'expr_symb' | → | 1: | 'expr_symb' |
|----|-------------|---|----|-------------|

EXPAN prend en argument une expression symbolique et renvoie une autre expression symbolique. EXPAN développe les expressions symboliques : les factorisations sont développées, les puissances sont décomposées en produits, etc. Pour développer totalement une expression symbolique, il convient de lui appliquer plusieurs fois EXPAN. EXPAN n'a pas d'effet sur les arguments réels ou complexes. Voir aussi COLCT.

EXPFIT

Modèle d'ajustement exponentiel

Accès (HP-48 G/GX) : [←] [5] (Σ PAR) (MODL) (EXPFI) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] (MODL) (EXP)

EXPFIT active le mode d'ajustement exponentiel pour les tracés statistiques (voir LR). La variable réservée Σ PAR est modifiée par EXPFIT.

EXPM

Calcul de $e^x - 1$

Accès (HP-48 G/GX) : [MTH] (HYP) [NXT] (EXPM)

Drapeaux : -3

1: réel →

1: réel

1: 'expr_symb' →

1: 'EXPM(expr_symb)'



Un nombre réel x étant donné en argument, EXPM calcule $e^x - 1$. Quand x tend vers 0, EXPM fournit un résultat plus précis que EXP(x) - 1.

EYEPT

Définition du point d'où est vu un tracé en perspective



Accès (HP-48 G/GX) : [↵] [8] [NXT] (3D) (VPAR) [NXT] (EYEPT)

Drapeaux : -

3: x
2: y
1: z →

3:
2:
1:

EYEPT prend trois réels en argument et définit le point de coordonnées cartésiennes (x, y, z) comme point depuis lequel est vu le tracé en volume. VPAR est modifiée par EYEPT.

FOλ

Fraction du pouvoir émissif total d'un corps noir



Accès (HP-48 G/GX) : [↵] [3] (UTILS) (FOλ)

Drapeaux : -3

2: $y\lambda$
1: temp →

2:
1: fraction

A partir de la longueur d'onde $y\lambda$ (en mètres) et de la température temp (en Kelvin), FOλ renvoie une fraction du pouvoir émissif total d'un corps noir entre les longueurs d'onde 0 et $y\lambda$ à la température temp.

FACT

Factorielle

Accès : [α] [α] [F] [A] [C] [cos] [α]

Drapeaux : -3, -20, -21

1: entier →

1: entier



FACT correspond à la fonction factorielle et équivaut à 1. Si l'argument de FACT est un réel, FACT renvoie $\Gamma(x+1)$. FACT assure la compatibilité ascendante des HP-48 avec la gamme HP-28.

FANNING

Facteur Fanning de friction d'un fluide en mouvement



Accès (HP-48 G/GX) : [←] [3] (UTILS) (FANNI) Drapeaux : -3

| | | | | |
|----|--------------|---|----|-----------|
| 2: | rugosité_rel | → | 2: | |
| 1: | Nb_Reynolds | | 1: | FFFanning |

FANNING prend comme argument la rugosité relative `rugosité_rel` (rugosité du conduit divisée par son diamètre) et le nombre de Reynolds `Nb_Reynolds`. Le résultat renvoyé est `FFFanning`, autrement dit, le facteur de friction Fanning. FANNING n'est utilisable qu'avec une température et une viscosité constantes. La rugosité du conduit et son diamètre doivent aussi être constants.

FC?

Test de l'état d'un drapeau (indicateur)

Accès : [PRG] (TEST) [NXT] [NXT] (FC?) Drapeaux : -

| | | | | |
|----|----------|---|----|---|
| 1: | num_drap | → | 1: | 0 |
| 1: | num_drap | → | 1: | 1 |

FC? prend comme argument le numéro `num_drap` d'un drapeau (c'est-à-dire un indicateur ou *flag*) et renvoie 0 si l'indicateur est actif (armé) ou 1 s'il est inactif (désarmé). Sur HP-48 G/GX, la liste des flags peut être obtenue avec [→] [CST] (FLAG). Par exemple, pour savoir si les transferts *via* le port série s'effectuent en mode binaire, introduisez -35 dans la pile (numéro du flag précédé par le signe «-» obtenu avec [+/-]) puis exécutez FC?. Si FC? renvoie 0, le drapeau (flag) est armé et les transferts s'effectuent en mode binaire.

**FC?C**

Test de l'état d'un drapeau (indicateur) et désarmement

Accès : [PRG] (TEST) [NXT] [NXT] (FC?C) Drapeaux : -

| | | | | |
|----|----------|---|----|---|
| 1: | num_drap | → | 1: | 0 |
| 1: | num_drap | → | 1: | 1 |

FC?C fonctionne exactement comme FC? (voir aussi FC?), place 0 dans la pile si le drapeau (*flag* ou indicateur) n°`num_drap` est armé ou place 1 dans la pile si ce drapeau est désarmé. Contrairement à FC?, FC?C désarme l'indicateur à l'issue du test.

FFT



Transformée de Fourier

Accès (HP-48 G/GX) : [MTH] [NXT] (FFT) (FFT) Drapeaux : -

| | | | | |
|----|---------------|---|----|---------------|
| 1: | [[tableau_1]] | → | 1: | [[tableau_2]] |
|----|---------------|---|----|---------------|

FFT prend comme argument un tableau [[tableau_1]] (vecteur ou matrice dont les dimensions ne sont pas quelconques) et renvoie un autre tableau correspondant à la transformée de Fourier unidimensionnelle ou bidimensionnelle. Soient n et m deux entiers naturels non nuls. FFT calcule la transformée de Fourier unidimensionnelle si [[tableau_1]] est un vecteur de 2^n éléments ou une matrice de dimensions $2^n \times 1$ ou 1×2^n . FFT calcule la transformée de Fourier bidimensionnelle si [[tableau_1]] est une matrice de dimensions $2^n \times 2^m$.



FINDALARM

Index de la sonnerie programmée

Accès (HP-48 G/GX) : [←] [4] (ALRM) (FINDA) Drapeaux : -42

Accès (HP-48 S/SX) : [←] [4] (ALRM) [NXT] (FINDA)

| | | | | |
|----|----------------|---|----|--------------|
| 1: | 0 | → | 1: | index_alarme |
| 1: | { date heure } | → | 1: | index_alarme |
| 1: | date | → | 1: | index_alarme |

FINDALARM renvoie index_alarme qui est un index d'alarme. index_alarme vaut 0 si aucune alarme correspondant à la recherche n'est détectée. Si l'argument est 0, FINDALARM affiche l'index de la première alarme devant être déclenchée. date est une date au format HP-48. heure est une heure au format HP-48. FINDALARM peut prendre date ou { date heure } comme arguments et rechercher la première alarme devant être déclenchée à partir de midi à la date choisie (argument date) ou à partir de l'heure choisie à la date choisie (argument { date heure }).



FINISH

Arrêt du mode serveur Kermit

Accès (HP-48 G/GX) : [←] [1] (SRVR) (FINIS) Drapeaux : -33, -39

Accès (HP-48 S/SX) : [←] [PRG] (FINIS)

L'instruction FINISH, exécutée par le client Kermit, demande au serveur Kermit de quitter le mode serveur.

FIX

Affichage des nombres avec un nombre fixe de décimales

Accès (HP-48 G/GX) : [\leftarrow] [CST] (FMT) (FIX) Drapeaux : -

Accès (HP-48 S/SX) : [\leftarrow] [CST] (FIX)

| | | | | |
|----|----------|---|----|--|
| 1: | nb_decim | → | 1: | |
|----|----------|---|----|--|

FIX prend comme argument nb_decim qui est un entier naturel compris entre 0 et 11 inclus. FIX fait de nb_decim le nombre fixe de décimales avec lequel seront affichés les nombres. Par exemple, après avoir exécuté 2 FIX, l'entier 7 sera affiché sous la forme 7.00.

FLOOR

Plus grand entier inférieur ou égal au réel donné en argument

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] [NXT] (FLOOR) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] (FLOOR)

| | | | | |
|----|------|---|----|--------|
| 1: | réel | → | 1: | entier |
|----|------|---|----|--------|

| | | | | |
|----|------------|---|----|--------------|
| 1: | réel_unité | → | 1: | entier_unité |
|----|------------|---|----|--------------|

| | | | | |
|----|-------------|---|----|--------------------|
| 1: | 'expr_symb' | → | 1: | 'FLOOR(expr_symb)' |
|----|-------------|---|----|--------------------|

FLOOR prend un réel en argument et le remplace par le premier entier immédiatement inférieur (ou égal). Attention ! Avec les nombres négatifs, FLOOR n'est pas équivalent à la partie entière de la valeur absolue.



FOR

En-tête des boucles à compteur For... Next et For... Step

Accès (HP-48 G/GX) : [PRG] (BRCH) (FOR) (FOR) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) (FOR)



Var_compteur est la variable locale utilisée pour compter le nombre d'exécution de la boucle. Var_compteur varie entre Compt_min et Compt_max. Var_incr est le pas d'incrément de Var_compteur, autrement, l'intervalle entre deux valeurs de Var_compteur.

Cas **FOR... NEXT** : Compt_min Compt_Max FOR Var_compteur
 Instructions de la boucle..
 NEXT

Cas **FOR... STEP** : Compt_min Compt_Max FOR Var_compteur
 Instructions de la boucle..
 Var_incr STEP

FP**Extraction de la partie décimale d'un nombre**

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] (FP)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] (FP)

| | | | | |
|----|-------------|---|----|-----------------|
| 1: | réel | → | 1: | réel |
| 1: | 'expr_symb' | → | 1: | 'FP(expr_symb)' |

FP prend un nombre réel en argument et renvoie la partie décimale de ce nombre (la partie entière est perdue). FP fonctionne aussi avec les objets-unités. Voir aussi IP.

FREE**Désolidarisation de la carte RAM (port 1)**

Accès (HP-48 G/GX) : [α] [α] [F] [→] [E] [E] [ENTER]

Drapeaux : -

Accès (HP-48 S/SX) : [←] [VAR] [NXT] [NXT] (FREE)

| | | | | |
|----|------------|---|----|--|
| 2: | { } | → | 2: | |
| 1: | num_port | | 1: | |
| 2: | num_biblio | → | 2: | |
| 1: | num_port | | 1: | |

FREE prend comme argument une liste vide { } ou un numéro de bibliothèque, ou même, un nom de sauvegarde, voire, une liste contenant des noms de sauvegarde et des numéros de bibliothèques. Au niveau 1 doit être placé le numéro du port. FREE libère la mémoire vive utilisateur auparavant fusionnée et associée au port n°num_port.

FREE1**Désolidarisation de la carte RAM**

Accès (HP-48 G/GX) : [←] [2] (FREE1)

Drapeaux : -

| | | | | |
|----|----------------|---|----|--|
| 1: | { } | → | 1: | |
| 1: | num_biblio | → | 1: | |
| 1: | nom_sauvegarde | → | 1: | |

FREE1 annule la fusion de la carte RAM en port 1 avec la RAM interne (cas { }). Si l'argument précise un objet (un numéro de bibliothèque, un nom de sauvegarde ou une liste de numéros de bibliothèque et de noms de sauvegardes), ces objets passent du port 0 au port désolidarisé.

FREEZE**Blocage de l'affichage en attendant une pression sur le clavier**Accès (HP-48 G/GX) : [PRG] [NXT] (OUT) (FREEZ) **Drapeaux : -**

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] [NXT] (FREEZ)

| | | | |
|----|------|---|----|
| 1: | zone | → | 1: |
|----|------|---|----|

Pour permettre la lecture d'un résultat ou d'un message, il est souvent utile d'interdire le rafraîchissement immédiat de l'écran. `zone` est un entier qui détermine la ou les zones de l'écran à ne pas rafraîchir tant qu'une touche n'a pas été pressée. La pile étant affichée, on distingue trois zones d'affichage : en haut se trouve la zone d'état, au milieu est affichée la pile, et en bas se place la barre des menus. La pile étant affichée, `zone` peut prendre des valeurs comprises entre 1 et 7 :

- si `zone` vaut 1, la zone d'état est figée,
- si `zone` vaut 2, la pile (et, éventuellement, la ligne de commande), sont figées,
- si `zone` vaut 3, la zone d'état, la pile (et, éventuellement, la ligne de commande), sont figées,
- si `zone` vaut 4, la barre des menus est figée,
- si `zone` vaut 5, la zone d'état et la barre des menus sont figées,
- si `zone` vaut 6, la barre des menus, la pile (et, éventuellement, la ligne de commande), sont figées,
- si `zone` vaut 7, tout l'écran est figé.

L'écran graphique (PICT, voir `PVIEW`) étant affiché, vous pouvez le figer en utilisant 3 ou 7 comme argument de `FREEZE`.

FS?**Test de l'état d'un drapeau (indicateur)**Accès : [PRG] (TEST) [NXT] [NXT] (FS?) **Drapeaux : -**

| | | | | |
|----|----------|---|----|---|
| 1: | num_drap | → | 1: | 0 |
|----|----------|---|----|---|

| | | | | |
|----|----------|---|----|---|
| 1: | num_drap | → | 1: | 1 |
|----|----------|---|----|---|

`FS?` prend comme argument le numéro `num_drap` d'un drapeau (c'est-à-dire un indicateur ou *flag*) et renvoie 1 si l'indicateur est actif (armé) ou 0 s'il est inactif (désarmé). Sur HP-48 G/GX, la liste des flags peut être obtenue avec [↵] [CST] (`FLAG`). Par exemple, pour savoir si les transferts *via* le port série s'effectuent en mode binaire, introduisez -35 dans la pile (numéro du flag précédé par le signe «-» obtenu avec [+/-]) puis exécutez `FS?`. Si `FS?` renvoie 1, le drapeau (flag) est armé et les transferts s'effectuent en mode binaire. Voir aussi `FC?`.

FS?C**Test de l'état d'un drapeau (indicateur) et désarmement**

Accès : [PRG] (TEST) [NXT] [NXT] (FS?C)

Drapeaux : -

| | | | | |
|----|----------|---|----|---|
| 1: | num_drap | → | 1: | 0 |
| 1: | num_drap | → | 1: | 1 |

FS?C fonctionne exactement comme FS? (voir aussi FS?) et place 1 dans la pile si le drapeau (*flag* ou indicateur) n°num_drap est armé ou place 0 dans la pile si ce drapeau est désarmé. Mais, contrairement à FS?, FS?C désarme l'indicateur à l'issue du test.

FUNCTION**Active le mode graphique consacré aux fonctions cartésiennes**

Accès : [←] [8] (PTYPE) (FUNC)

Drapeaux : -28, -31

FUNCTION active le mode graphique consacré aux fonctions cartésiennes, autrement dit, après l'exécution de FUNCTION, DRAW trace la fonction placée dans la variable réservée EQ.

GET**Extraction d'un élément d'une liste ou d'un tableau**

Accès (HP-48 G/GX) : [PRG] (LIST) (ELEM) (GET)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] [NXT] (GET)

| | | | | |
|----|-------------|---|----|---------|
| 2: | [[tableau]] | → | 2: | |
| 1: | position | | 1: | élément |
| 2: | { liste } | → | 2: | |
| 1: | position | | 1: | élément |

GET renvoie l'élément identifié par *position* dans une liste ou dans un tableau. Si le tableau est un vecteur, *position* est un entier naturel supérieur ou égal 1 déterminant le rang de l'élément à extraire. Si le tableau est une matrice *position* peut être une liste (dont le premier élément correspond à la ligne de l'élément à extraire, et le second, à sa colonne) ou un entier représentant le numéro de l'élément (les éléments d'une matrice étant numérotés de haut en bas et de gauche à droite). Si l'élément est à extraire d'une liste, *position* est le numéro de l'élément à extraire de cette liste. Le tableau ou la liste peuvent être remplacés par leurs noms placés entre apostrophes. Quand *position* est un simple numéro d'ordre, il peut, ou non, être placé entre accolades .

GETI

Extraction d'un élément d'une liste ou d'un tableau

Accès (HP-48 G/GX) : [PRG] (LIST) (ELEM) (GETI) Drapeaux : -64

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] [NXT] (GETI)

| | | |
|--|---|---|
| 3: 2: [[tableau]] 1: position | → | 3: [[tableau]] 2: position+1 1: élément |
| 3: 2: { liste } 1: position | → | 3: { liste } 2: position+1 1: élément |

Comme GET, GETI renvoie l'élément identifié par `position` dans une liste ou dans un tableau. Si le tableau est un vecteur, `position` est un entier naturel supérieur ou égal 1 déterminant le rang de l'élément à extraire. Si le tableau est une matrice, `position` peut être une liste (dont le premier élément correspond à la ligne de l'élément à extraire, et le second, à sa colonne) ou un entier représentant le numéro de l'élément (les éléments d'une matrice étant numérotés de haut en bas et de gauche à droite). Si l'élément est à extraire d'une liste, `position` est le numéro de l'élément à extraire de cette liste. Le tableau ou la liste peuvent être remplacés par leurs noms placés entre apostrophes. Quand `position` est un simple numéro d'ordre, il peut, ou non, être placé entre accolades.



GETI renvoie aussi le tableau ou la liste non modifié au niveau 3 ainsi que l'index `position` pointant sur l'élément suivant de la structure de base (la liste ou le tableau).

GOR

Superposition de deux objets graphiques

Accès (HP-48 G/GX) : [PRG] (GROB) (GOR) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] (GOR)

| | | |
|--|---|--|
| 3: grob_dest 2: coord 1: grob_aj | → | 3: 2: 1: grob |
| 3: PICT 2: coord 1: grob_aj | → | 3: 2: 1: |

GOR place l'objet graphique `grob_aj` au-dessus de l'objet graphique `grob_dest` (ou au-dessus de `PICT`). →



→ GOR renvoie dans la pile l'objet graphique correspondant à la superposition des deux objets graphiques donnés en arguments. Cette superposition s'opère par *OU* logique, ainsi, il suffit que l'un des points de l'un des deux grob superposés soit allumé pour qu'il le soit dans le grob renvoyé. coord correspond aux coordonnées de l'angle supérieur gauche de grob_aj dans grob_dest (ou dans PICT). coord peut être de la forme (x, y) (coordonnées dans le repère défini par l'utilisateur) ou de la forme { #x #y } (coordonnées matérielles).

GRAD

Le grade devient l'unité de mesure d'angle par défaut

Accès (HP-48 G/GX) : [↶] [CST] (ANGL) (GRAD) **Drapeaux :** -17, -18

Accès (HP-48 S/SX) : [↶] [CST] [NXT] [NXT] (GRAD)

GRAD fait du grade l'unité de mesure d'angle par défaut. Donc, après activation de GRAD et lorsque l'unité n'est pas précisée, la calculatrice considère que le grade est l'unité employée. Pour utiliser d'autres unités, voir DEG et RAD.



GRAPH

Appel de l'environnement graphique PICTURE (ou GRAPH)

Accès (HP-48 G/GX) : [α] [α] [MTH] [→] [A] [←] [PRG] **Drapeaux :** -

Accès (HP-48 S/SX) : [↶] [←]

GRAPH provoque l'affichage de l'environnement graphique (PICTURE sur G/GX, GRAPH sur S/SX) et rend disponible le pointeur graphique (à déplacer avec les flèches). Sur HP-48 G/GX, GRAPH équivaut à PICTURE et ne justifie donc sa présence que par le maintien de la compatibilité ascendante entre les HP-48.

GRIDMAP

Mode graphique trois dimensions (grille 3D)



Accès (HP-48 G/GX) : [↶] [8] [NXT] (3D) (PTYPE) (GRID) **Drapeaux :** -

GRIDMAP active le mode graphique consacré aux tracés en trois dimensions (dans l'espace) sous forme de grilles 3D. Après l'exécution de GRIDMAP, l'instruction DRAW trace une grille «mappée» en se fondant sur les contenus des variables réservées EQ (équation à représenter graphiquement) VPAR (vue volumique) et PPAR (paramètres de PICTURE).



→GROB

Création d'un GROB à partir d'un objet placé dans la pile

Accès (HP-48 G/GX) : [PRG] (GROB) (→GRO) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] (→GRO)



→GROB crée un GROB (objet graphique) représentant l'objet objet_à_rep placé au niveau 2 de la pile. Taille est un entier variant entre 0 et 3 inclus et déterminant la taille des caractères utilisés pour représenter objet_à_rep dans GROB. Valeurs de Taille :

- si Taille vaut 0, le GROB est créé à partir de l'image générée par *Equation Writer* (ceci n'est valable que si objet_à_rep est une expression algébrique ou un objet-unité),
- si Taille vaut 1, le GROB est créé avec la petite taille de caractères,
- si Taille vaut 2, le GROB est créé avec les caractères normaux,
- si Taille vaut 3, le GROB est créé avec de grands caractères.



GXOR

Superposition de deux objets graphiques (avec OU exclusif)

Accès (HP-48 G/GX) : [PRG] (GROB) (GXOR) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] (GXOR)



GXOR place l'objet graphique grob_aj au-dessus de l'objet graphique grob_dest (ou au-dessus de PICT). GXOR renvoie dans la pile l'objet graphique correspondant à la superposition des deux objets graphiques donnés en arguments. Cette superposition s'opère par OU logique exclusif, ainsi, pour qu'un point soit allumé dans le GROB résultat, il faut qu'il soit allumé dans l'un des deux objets graphiques mais pas dans les deux. coord correspond aux coordonnées de l'angle supérieur gauche de grob_aj dans grob_dest (ou dans PICT). coord peut être de la forme (x, y) (coordonnées dans le repère défini par l'utilisateur) ou de la forme { #x #y } (coordonnées matérielles).

*H

Multiplication par un facteur de l'intervalle de représentation verticale

Accès (HP-48 G/GX) : [↵] [8] (PPAR) [NXT] (*H) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [8] (PLOT) [NXT] [NXT] (*H)

| | | | |
|----|----------------|---|----|
| 1: | Facteur | → | 1: |
|----|----------------|---|----|

Les axes du repère utilisé pour les représentations graphiques sont représentés sur des intervalles stockés dans la variable réservée PPAR. *H multiplie par le réel Facteur l'intervalle vertical de représentation sans modifier la position du centre de la représentation graphique.

HALT

Interruption de l'exécution d'un programme

Accès (HP-48 G/GX) : [PRG] [NXT] (RUN) (HALT) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) (HALT)



HALT interrompt l'exécution du programme qui ne peut être relancé que par CONT obtenu avec [↵] [ON]. Voir aussi KILL.

HEAD

Extraction du premier élément d'une chaîne ou d'une liste



Accès (HP-48 G/GX) : [PRG] (LIST) (ELEM) [NXT] (HEAD) Drapeaux : -

| | | | | |
|----|----------|---|----|-----|
| 1: | "chaîne" | → | 1: | "c" |
|----|----------|---|----|-----|

| | | | | |
|----|-----------|---|----|-------|
| 1: | { liste } | → | 1: | objet |
|----|-----------|---|----|-------|

HEAD prend comme argument une chaîne de caractères ou une liste d'objets et renvoie, soit une chaîne réduite au premier caractère de la chaîne donnée en argument, soit le premier objet de la liste donnée en argument.

HEX

Mode hexadécimal

Accès : [MTH] (BASE) (HEX) Drapeaux : -5, -6, -7, -8, -9, -10, -11, -12



HEX active le mode hexadécimal (calculs en base 16). Ce mode concerne l'affichage des entiers binaires. La saisie d'un entier système dans une base autre que la base 16 provoque sa conversion dans cette base.

HISTOGRAM**Activation du mode graphique de tracé d'histogrammes**Accès (HP-48 G/GX) : [←] [8] [NXT] (STAT) (PTYPE) (HISTO) **Drapeaux** : -

Accès (HP-48 S/SX) : [←] [8] (PTYPE) [NXT] (HIST)

HISTOGRAM active le mode graphique consacré au tracé d'histogrammes. Après l'exécution de HISTOGRAM, DRAW provoque le tracé d'un histogramme correspondant aux données stockées dans Σ DAT. DRAW est aussi lié à Σ PAR et à PPAR.

HISTPLOT**Tracé d'histogramme**Accès (HP-48 G/GX) : [←] [5] (PLOT) (HISTP) **Drapeaux** : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] (HISTP)

HISTPLOT déclenche le tracé d'un histogramme représentant les données de la colonne de Σ DAT sélectionnée par XCOL (XCOL est l'une des données stockées dans la liste Σ PAR).

→HMS**Conversion vers le format heures-minutes-secondes**Accès (HP-48 G/GX) : [←] [4] [NXT] (→HMS) **Drapeaux** : -

Accès (HP-48 S/SX) : [←] [4] [NXT] [NXT] (→HMS)

| | | | | |
|----|------|---|----|-----------|
| 1: | réel | → | 1: | H.MMSS... |
|----|------|---|----|-----------|

A partir d'un nombre réel ayant une partie décimale, →HMS produit un nombre H.MMSS... au format *heures-minutes-secondes* (système sexagésimal, base 60). Dans ce nombre, le point décimal sépare les heures des minutes. H (heures) correspond à la partie entière du nombre, MM représente les minutes, et SS les secondes. SS peut éventuellement être suivi par d'autres nombres représentant des fractions décimales de secondes (1/10e de seconde, 1/100e, 1/1000e, etc.).

**HMS+****Somme de deux nombres sexagésimaux**Accès (HP-48 G/GX) : [←] [4] [NXT] (HMS+) **Drapeaux** : -

Accès (HP-48 S/SX) : [←] [4] [NXT] [NXT] (HMS+) →

| | | |
|--|---|----------------------------------|
| 2: H.MMSS... 1: H.MMSS... | → | 2: 1: H.MMSS... |
|--|---|----------------------------------|

→ HMS+ renvoie au format sexagésimal la somme de deux nombres sexagésimaux, voir →HMS pour la description de ce format.

HMS-

Différence de deux nombres sexagésimaux

Accès (HP-48 G/GX) : [↵] [4] [NXT] (HMS-) **Drapeaux :** -

Accès (HP-48 S/SX) : [↵] [4] [NXT] [NXT] (HMS-)

| | | |
|--|---|----------------------------------|
| 2: H.MMSS... 1: H.MMSS... | → | 2: 1: H.MMSS... |
|--|---|----------------------------------|

HMS- renvoie au format sexagésimal la différence de deux nombres sexagésimaux (le nombre du niveau 1 est soustrait à celui du niveau 2), voir →HMS pour la description de ce format.

HMS→

Conversion vers le format décimal

Accès (HP-48 G/GX) : [↵] [4] [NXT] (HMS→) **Drapeaux :** -

Accès (HP-48 S/SX) : [↵] [4] [NXT] [NXT] (HMS→)

| | | |
|---------------------|---|----------------|
| 1: H.MMSS... | → | 1: réel |
|---------------------|---|----------------|

A partir d'un nombre H.MMSS... au format *heures-minutes-secondes* (système sexagésimal, base 60), HMS→ crée un nombre réel qui est l'équivalent en base 10 du nombre exprimé en notation sexagésimal. Dans un nombre H.MMSS..., le point décimal sépare les heures des minutes. H (heures) correspond à la partie entière du nombre (en base 10), MM représente les minutes, et SS les secondes. SS peut éventuellement être suivi par d'autres nombres représentant des fractions décimales de secondes (1/10e de seconde, 1/100e, 1/1000e, etc.).

HOME

Retour au répertoire racine

Accès : [↵] ['] **Drapeaux :** -

HOME faite de HOME, le répertoire courant. HOME est le répertoire racine, autrement dit, il est à la base de l'arborescence des répertoires et contient donc tous les autres répertoires.



i

Nombre imaginaire

Accès : $[\alpha]$ $[\leftarrow]$ [CST]

Drapeaux : -2, -3

| | | | |
|----|---|----|-------|
| 1: | → | 1: | (0,1) |
| 1: | → | 1: | 'i' |



i est le nombre imaginaire utilisé par les nombres imaginaires. $[\alpha]$ $[\leftarrow]$ [CST] place 'i' dans la pile, autrement dit constante symbolique i sous la forme d'une expression algébrique. $[\alpha]$ $[\leftarrow]$ [CST] $[\rightarrow]$ [EVAL] (sur HP-48 S/SX) et $[\alpha]$ $[\leftarrow]$ [CST] $[\leftarrow]$ [EVAL] (sur HP-48 G/GX) renvoient (0, 1), autrement dit la forme numérique de i présentée avec le format utilisé par la HP-48 pour représenter les nombres complexes.

IDN

Création d'une matrice identité

Accès (HP-48 G/GX) : [MTH] (MATR) (MAKE) (IDN)

Drapeaux : -

Accès (HP-48 S/SX) : [MTH] (MATR) (IDN)

| | | | | |
|----|-------------|---|----|-------------|
| 1: | entier | → | 1: | [[matrice]] |
| 1: | [[matrice]] | → | 1: | [[matrice]] |



IDN crée une matrice identité. Rappelons qu'une matrice identité est une matrice carrée dont tous les éléments sont nuls sauf ceux de la première diagonale. En effet, ceux-ci valent 1 dans le cas d'une matrice réelle et (0, 1) dans le cas d'une matrice complexe. *entier* est un entier naturel non nul représentant le nombre de colonnes et le nombre de lignes de la matrice créée. Si l'argument est une matrice (cette matrice doit être carrée), IDN renvoie une matrice identité de mêmes dimensions. Dans ce second cas, la matrice renvoyée est complexe si la matrice donnée en argument l'est aussi.

IF

En-tête d'une structure de test IF... THEN... ELSE... END

Accès (HP-48 G/GX) : [PRG] (BRCH)-(IF) (IF)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH)-(IF)

IF introduit une structure de test. Une condition telle qu'une égalité ($b=1$), une différence ($ca \neq 1$) ou un test de supériorité ou d'infériorité doit être remplie pour que des instructions puissent être exécutées. →



→ Cas IF... THEN... END :

```

IF condition à remplir
THEN
  instructions à exécuter si la condition est remplie
END
    
```

Cas IF... THEN... ELSE... END :

```

IF condition à remplir
THEN
  instructions à exécuter si la condition est remplie
ELSE
  instructions à exécuter si la condition n'est pas remplie
END
    
```



IFERR

En-tête d'une structure d'interception d'erreur

Accès (HP-48 G/GX) : [PRG] [NXT] (ERROR)-(IFERR) Drapeaux : -55

Accès (HP-48 S/SX) : [PRG] (BRCH)-[NXT] [NXT] (IFERR)

IFERR introduit une structure d'interception d'erreur.

Cas IFERR... THEN... END :

```

IFERR
  instructions susceptibles de déclencher une erreur
THEN
  instructions à exécuter en cas d'erreur
END
    
```

Cas IFERR... THEN... ELSE... END :

```

IFERR
  instructions susceptibles de déclencher une erreur
THEN
  instructions à exécuter en cas d'erreur
ELSE
  instructions à exécuter si aucune erreur n'a été rencontrée
END
    
```

IFFT

Transformée de Fourier inverse



Accès (HP-48 G/GX) : [MTH] [NXT] (FFT) (IFFT) Drapeaux : -

1: [[tableau]] → 1: [[tableau]]

→ [[tableau]] peut être un vecteur ou une matrice. n et m sont des entiers naturels.

IFFT renvoie la transformée de Fourier inverse unidimensionnelle si [[tableau]] est un vecteur comportant 2^n éléments ou une matrice de dimensions $2^n \times 1$ ou 1×2^n . IFFT renvoie la transformée de Fourier bidimensionnelle si [[tableau]] est une matrice de dimensions $2^n \times 2^m$.

IFT

Evaluation conditionnelle d'un objet

Accès (HP-48 G/GX) : [PRG] (BRCH) [NXT] (IFT) **Drapeaux** : -

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] [NXT] (IFT)

| | | | | |
|----|------------|---|----|--------------------------|
| 2: | réel | → | 2: | |
| 1: | objet_eval | | 1: | selon réel et objet_eval |

IFT évalue objet_eval (qui peut être un programme, une commande ou n'importe quel autre objet pouvant occuper un niveau de la pile) si réel est non nul.



Un test tel que $a \ b \ >$ place 0 dans la pile si la condition n'est pas vérifiée et 1 si elle l'est. Un test peut donc être utilisé pour décider de l'évaluation de objet_eval.

IFTE

Evaluation conditionnelle d'un objet

Accès (HP-48 G/GX) : [PRG] (BRCH) [NXT] (IFTE) **Drapeaux** : -

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] [NXT] (IFTE)

| | | | | |
|----|--------------|---|----|-------------------------------|
| 3: | réel | → | 3: | |
| 2: | objet_eval≠0 | | 2: | |
| 1: | objet_eval=0 | | 1: | selon réel et objet_eval=0/≠0 |

IFTE évalue objet_eval≠0 (qui peut être un programme, une commande ou n'importe quel autre objet pouvant occuper un niveau de la pile) si réel est non nul.



Si réel est nul, c'est objet_eval=0 qui est évalué (objet_eval=0 peut être un objet quelconque). Un test tel que $a \ b \ >$ place 0 dans la pile si la condition n'est pas vérifiée et 1 si elle l'est. Un test peut donc être utilisé pour décider de l'évaluation de objet_eval=0 ou de objet_eval≠0.

IM

Extraction de la partie imaginaire d'un nombre complexe

Accès (HP-48 G/GX) : [MTH] [NXT] (CMPL) (IM) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) (IM)

| | | |
|----------------|---|--------------------|
| 1: réel | → | 1: 0 |
| 1: complexe | → | 1: réel |
| 1: [[matrice]] | → | 1: [[matrice]] |
| 1: 'expr_symb' | → | 1: 'IM(expr_symb)' |

IM renvoie la partie imaginaire du réel ou du complexe donné en argument. Si l'argument est un réel, IM renvoie 0 car, par définition, sa partie imaginaire est nulle. Si l'argument est un nombre complexe, IM renvoie un réel correspondant à la partie imaginaire de ce complexe. Si l'argument est une matrice réelle, IM renvoie une matrice réelle de mêmes dimensions dont tous les éléments sont nuls. Si l'argument est une matrice complexe, IM renvoie une matrice réelle dont tous les éléments correspondent à la partie imaginaire de l'élément correspondant dans la matrice donnée en argument.

INCR

Incrémentation d'une variable réelle

Accès (HP-48 G/GX) : [←] [VAR] (ARITH) (INCR) Drapeaux : -

Accès (HP-48 S/SX) : [↗] [VAR] (INCR)

| | | |
|---------------|---|---------|
| 1: 'nom_réel' | → | 1: réel |
|---------------|---|---------|

INCR prend comme argument le nom d'une variable correspondant à un nombre réel. INCR ajoute 1 au nombre associé à cette variable puis place dans cette variable le nouveau nombre qui est aussi placé au niveau 1 de la pile.



INDEP

Paramètres associés à la variable indépendante

Accès (HP-48 G/GX) : [←] [8] (PPAR) (INDEP) Drapeaux : -

Accès (HP-48 S/SX) : [↗] [8] (INDEP)

| | | |
|--------------------|---|----|
| 1: 'indep' | → | 1: |
| 1: {indep min max} | → | 1: |



| | | |
|--|---|----------|
| 2: min 1: max | → | 2: 1: |
| 1: (min max) | → | 1: |

INDEP définit la variable indépendant utilisée pour les représentations graphiques. INDEP peut définir le nom de cette variable, son intervalle de représentation ou les deux. indep est le nom de la variable indépendante qui doit, selon le cas, être ou non placé entre apostrophes (voir les exemples). min est la borne inférieure de l'intervalle de représentation, max sa borne supérieure.

INFORM



Création d'une zone de dialogue avec champ de saisie

Accès (HP-48 G/GX) : [PRG] [NXT] (IN) (INFOR) Drapeaux : -

| | | |
|--|---|---|
| 5: `nom_zone` 4: { champs_de_saisie } 3: formatage_zone 2: { Valeurs_Reset } 1: { Valeurs_initiales } | → | 5: 4: 3: 2: 1: 0 |
| 5: `nom_zone` 4: { champs_de_saisie } 3: formatage_zone 2: { Valeurs_Reset } 1: { Valeurs_initiales } | → | 5: 4: 3: 2: { valeurs_des_champs } 1: 1 |

INFORM crée des zones de dialogue dotées de champs de saisie. Si la saisie des données a été abandonnée avec [CANCEL] ou (CANCL), la zone de dialogue renvoie 0. Dans le cas contraire, elle renvoie les données saisies sous la forme d'une liste et place 1 au niveau 1 de la pile. Reportez-vous à la première partie de cet ouvrage pour plus de détails.



INPUT

Saisie via la ligne de commande

Accès (HP-48 G/GX) : [PRG] [NXT] (IN) (INPUT) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] (INPUT)

| | | |
|--|---|---|
| 2: "Message_1" 1: "Message_2" | → | 2: 1: "Message_2saisie" |
|--|---|---|

INPUT stoppe l'exécution du programme, affiche "Message_1" en haut de la pile et "Message_2" au niveau de la ligne de commande puis attend la saisie d'une donnée et sa validation par [ENTER]. INPUT place ensuite au niveau 1 une chaîne où sont juxtaposés Message_2 et les données saisies.



INV**Calcul de l'inverse**

Accès : [1/x]

Drapeaux : -3

| | | | | |
|----|-------------|---|----|------------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | [[matrice]] | → | 1: | [[matrice]] |
| 1: | 'expr_symb' | → | 1: | 'INV(expr_symb)' |

Si l'argument est un réel x (ou un objet-unité), INV renvoie $1/x$. Si l'argument est un complexe z , INV renvoie un complexe z' tel que $z \times z' = 1$. Si l'argument est une matrice (nécessairement carrée), INV renvoie la matrice inverse correspondante. Cette matrice, si elle multipliée par la matrice donnée en argument, renvoie une matrice identité de mêmes dimensions.

IP**Partie entière**

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] (IP)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] (IP)

| | | | | |
|----|-------------|---|----|-----------------|
| 1: | réel | → | 1: | entier |
| 1: | 'expr_symb' | → | 1: | 'IP(expr_symb)' |

IP extrait la partie entière d'un nombre réel. Appliquée à un nombre réel, la fonction IP renvoie donc un entier (on obtient un résultat similaire avec un objet-unité).

IR ou IR/W**Sélection d'un port série**

Accès (HP-48 G/GX) : [↵] [1] (IOPAR) (IR)

Drapeaux : -33, -34

Accès (HP-48 S/SX) : [↵] [PRG] (SETUP) (IR/W)

Dans le cadre des transmissions série, IR (sur HP-48 G/GX) et IR/W (sur HP-48 S/SX) permet de passer de la transmission par câble à la transmission par diodes infrarouges et inversement.

Dans un programme, préférez l'armement ou le désarmement direct des drapeaux (*flags*) 33 et 34.



ISOL

Isolation d'une variable au sein d'une expression algébrique

Accès : [←] [9] (ISOL)

Drapeaux : -1, -3

| | | | | |
|----|-------------|---|----|-------------|
| 2: | 'expr_symb' | → | 2: | |
| 1: | 'isol' | | 1: | 'expr_symb' |

ISOL prend en argument une expression algébrique 'expr_symb' et un nom global 'isol' correspondant au nom d'une variable globale intégrée à l'expression symbolique. Celle-ci est parcourue de gauche à droite et la première occurrence de 'isol' devient le membre de gauche d'une équation créée à partir de l'expression symbolique donnée en argument. L'équation produite est renvoyée sous la forme d'une expression symbolique.



Attention ! Si le drapeau 3 est armé, les noms globaux de variables sont remplacés par les valeurs numériques associées à ces noms.

KERRM

Rappel de la dernière erreur Kermit

Accès (HP-48 G/GX) : [←] [1] [NXT] (KERR)

Drapeaux : -

Accès (HP-48 S/SX) : [←] [PRG] [NXT] (KERR)

| | | | | |
|----|--|---|----|-----------|
| 1: | | → | 1: | "message" |
|----|--|---|----|-----------|

KERRM place dans la pile une chaîne de caractères qui correspond au dernier message d'erreur généré par une transmission défectueuse par voie série (câble ou infrarouge) et faisant appel au protocole de transfert de fichier Kermit.

KEY

Détection d'une pression sur une touche

Accès (HP-48 G/GX) : [PRG] [NXT] (IN) (KEY)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] (KEY)

| | | | | |
|----|--|---|----|----------|
| 1: | | → | 1: | 0 |
| 2: | | → | 2: | lign_col |
| 1: | | | 1: | 1 |



KEY est à placer dans une boucle infinie afin de détecter une pression sur une touche. Tant qu'aucune touche n'est pressée, KEY place 0 au niveau 1 de la pile. Si une touche est pressée, KEY place au niveau 2 →

→ de la pile un nombre réel `lign_col` dont le premier chiffre correspond à la ligne de touches sur laquelle se trouve la touche pressée (les lignes de touches sont numérotées de haut en bas) alors que le second chiffre correspond à la colonne dans laquelle se trouve la touche pressée (les colonnes sont numérotées de gauche à droite). Voir aussi `Wait`.

KGET

téléchargement d'un objet depuis un serveur Kermit

Accès (HP-48 G/GX) : [←] [1] (SRVR) (KGET) **Drapeaux** : -33, -35, -36, -39

Accès (HP-48 S/SX) : [←] [PRG] (KGET)

1: → 1:

KGET permet, depuis une HP-48, de télécharger par la voie série un fichier disponible auprès du serveur Kermit connecté. Ce serveur peut être une autre HP-48 (voir le drapeau 35). `nom` est le nom du fichier à télécharger. `nom` peut être un nom global (`'nom'`) ou une chaîne (`"nom"`), ou même une liste de noms d'objets à télécharger (`{nom_1 nom_2 nom_3...}`). Vous pouvez renommer les objets téléchargés. Dans ce cas, `nom` est une liste où se trouve une ou plusieurs sous-listes à deux éléments, par exemple `{ { nom_a1 nom_a2 } { nom_b1 nom_b2 } }` provoque le téléchargement des objets `nom_a1` et `nom_a2` depuis le serveur ; `nom_a1` est renommé `nom_a2` alors que `nom_b1` est renommé `nom_b2`.



KILL

Annulation des programmes interrompus

Accès (HP-48 G/GX) : [PRG] [NXT] (RUN) (KILL) **Drapeaux** : -

Accès (HP-48 S/SX) : [PRG] (CTRL) (KILL)

Un programme en cours d'exécution peut être interrompu par les instructions `HALT` et `PROMPT` (`INPUT` et `WAIT` n'interrompent pas le programme mais le suspendent). `KILL` annule le programme en cours d'exécution et tous les processus qui lui sont associés.

LABEL

Affichage du nom des axes du repère

Accès (HP-48 G/GX) : [←] [8] [NXT] (LABEL) **Drapeaux** : -

Accès (HP-48 S/SX) : [→] [8] [NXT] [NXT] (LABEL) →

→ LABEL provoque l’affichage des variables (indépendante et dépendante) associées à chaque axe du repère ainsi que l’affichage des bornes des intervalles de représentation des axes. Le contenu de la variable réservée PPAR est utilisé.

LAST

Rappel des arguments utilisés par la dernière commande

Accès : [α] [α] [NXT] [A] [SIN] [COS]

Drapeaux : -55

| | | |
|----|---|------------------------|
| 2: | → | 2: |
| 1: | | 1: <i>selon le cas</i> |



LAST est équivalent à LASTARG. Voir LASTARG.

LASTARG

Rappel des arguments utilisés par la dernière commande

Accès (HP-48 G/GX) : [↵] [EEX]

Drapeaux : -55

Accès (HP-48 S/SX) : [↵] [2]

| | | |
|----|---|------------------------|
| 2: | → | 2: |
| 1: | | 1: <i>selon le cas</i> |



LASTARG renvoie dans la pile les arguments qui viennent d’être utilisés par la dernière commande ou le dernier programme évalué.

→LCD

Insertion d’un objet graphique sur la pile

Accès (HP-48 G/GX) : [PRG] (GROB) [NXT] (→LCD)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] [NXT] (→LCD)

| | | |
|---------------------------|---|----|
| 1: <i>objet_graphique</i> | → | 1: |
|---------------------------|---|----|

→LCD prend un objet graphique (GROB) au niveau 1 de la pile et l’affiche à la place de la pile en recouvrant la zone d’état et l’affichage de la pile (la barre des menus n’est pas recouverte avec →LCD). L’objet graphique est affiché à partir de l’angle supérieur gauche de la zone d’état. Ses dimensions peuvent être quelconques mais s’il dépasse 131 pixels de largeur sur 56 pixels de hauteur, les colonnes surnuméraires et/ou les lignes surnuméraires sont perdues (il n’est pas possible de faire défiler l’objet graphique). Ne confondez pas l’insertion d’un objet graphique sur la pile et son insertion dans l’environnement PICT (HP-48 G/GX) ou GRAPH (HP-48 S/SX).



LCD→

Création d'un objet graphique fondé sur l'affichage de la pile

Accès (HP-48 G/GX) : [PRG] (GROB) [NXT] (LCD→) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] [NXT] (LCD→)

1: → 1:

LCD→ crée un objet graphique (GROB) de 131 colonnes et 64 lignes représentant l'affichage de la pile (zone d'état, pile et barre des menus).

Contrairement à →LCD qui ne traite pas la barre des menus (GROB limité à 131×56), LCD→ crée un objet graphique incluant la barre des menus (GROB 131×64). Voir aussi →LCD.



LIBEVAL

Exécution d'une fonction d'une bibliothèque



Accès : [α] [α] [NXT] [CST] [B] [E] [√x] [A] [NXT] Drapeaux : -

1: → 1:

LIBEVAL évalue une fonction de bibliothèque. Cette fonction est définie par l'argument `entier_binaire` qui est un entier système hexadécimal. L'argument est de la forme `#aaaabbbh`. `aaa` est un nombre hexadécimal à trois chiffres qui détermine la bibliothèque alors que `bbb` est un nombre hexadécimal à trois chiffres représentant le numéro de la fonction voulue au sein de la bibliothèque `aaa`.

LIBS

Informations relatives au bibliothèque attachée au répertoire

Accès (HP-48 G/GX) : [←] [2] (LIBS) Drapeaux : -

Accès (HP-48 S/SX) : [←] [VAR] [NXT] (LIBS)

1: → 1:

Pour être utilisables, les bibliothèques doivent être attachées à un des répertoires. LIBS renvoie une liste contenant des informations relatives aux bibliothèques attachées au répertoire en cours (dernier répertoire du chemin courant). `{ liste }` contient des lots de trois informations pour chaque bibliothèque attachée au répertoire courant. Ces lots sont de la forme `"nom" num_bib num_port` où `"nom"` est une chaîne contenant le nom de la bibliothèque et où `num_bib` et `num_port` sont, respectivement, le numéro de la bibliothèque et le numéro du port.

LINE

Tracé d'un segment de droite

Accès (HP-48 G/GX) : [PRG] (PICT) (LINE)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) (LINE)

| | |
|----|---------|
| 2: | coord_1 |
| 1: | coord_2 |



| | |
|----|--|
| 2: | |
| 1: | |



LINE trace un segment de droite entre le point de coordonnées `coord_1` et le point de coordonnées `coord_2`. `coord_1` et `coord_2` peuvent être de la forme (x, y) ou de la forme $\{ \#x \#y \}$. (x, y) sont des coordonnées-utilisateurs dans le repère défini par l'utilisateur alors que $\{ \#x \#y \}$ sont des coordonnées matérielles (dans une matrice 131×64).

Σ LINE

Expression définissant la droite d'ajustement

Accès (HP-48 G/GX) : [\leftarrow] [5] (FIT) (Σ LINE)

Drapeaux : -

Accès (HP-48 S/SX) : [\leftarrow] [5] [NXT] [NXT] (Σ LINE)

| | |
|----|--|
| 1: | |
|----|--|



| | |
|----|-------------|
| 1: | 'expr_symb' |
|----|-------------|

Σ LINE place au niveau 1 de la pile, une expression symbolique `'expr_symb'` définissant la droite d'ajustement fondée sur les données statistiques contenues dans Σ DAT. `'expr_symb'` est directement liée au modèle de régression choisi (LINFIT, LOGFIT, EXPFIT ou PWRFIT).

LININ

Vérification de la linéarité d'une expression algébrique



Accès (HP-48 G/GX) : [PRG] (TEST) [\leftarrow] [NXT] (LININ)

Drapeaux : -

| | |
|----|-------------|
| 2: | 'expr_symb' |
| 1: | 'var' |



| | |
|----|---|
| 2: | |
| 1: | 0 |

| | |
|----|-------------|
| 2: | 'expr_symb' |
| 1: | 'var' |



| | |
|----|---|
| 2: | |
| 1: | 1 |



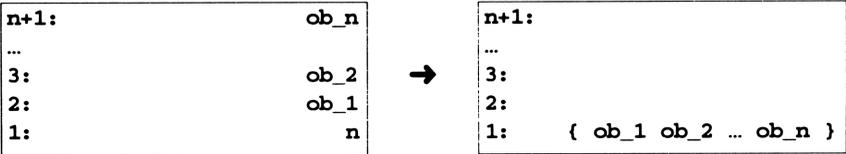
LININ vérifie que la variable globale `'var'` (par exemple, `'x'`) n'est présente dans l'expression symbolique `'expr_symb'` qu'au degré 1 (on teste la linéarité de la variable). LININ place 1 au niveau 1 de la pile si `'var'` n'existe qu'au degré 1 dans `'expr_symb'`. Si ce n'est pas le cas, LININ renvoie 0 au niveau 1 de la pile.

→**LIST**

Création d'une liste à partir des objets placés dans la pile

Accès (HP-48 G/GX) : [PRG] (TYPE) (→LIST) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) (→LIST)



→LIST prend n objets ob_1 à ob_n aux niveaux 2 à n+1 de la pile et constitue une liste { ob_1 ob_2 ... ob_n } contenant ces n objets.

ΔLIST



Différences entre les éléments d'une liste

Accès (HP-48 G/GX) : [MTH] (LIST) (ΔLIST) Drapeaux : -



Soit { ob_1 ob_2 ob_3 ... ob_n-1 ob_n } une liste de n éléments donnée en argument à ΔLIST, ΔLIST renvoie la liste { ob_2-ob_1 ob_3-ob_2 ... ob_n-ob_n-1 } contenant n-1 éléments.



Les éléments de la liste doivent autoriser la soustraction (nombres réels, nombres complexes, expressions algébriques...).

ΠLIST



Produit des éléments d'une liste

Accès (HP-48 G/GX) : [MTH] (LIST) (ΠLIST) Drapeaux : -



Soit { ob_1 ob_2 ob_3 ... ob_n-1 ob_n } une liste de n éléments donnée en argument à ΠLIST, ΠLIST renvoie produit qui correspond au produit ob_1*ob_2*ob_3*... *ob_n-1*ob_n des n éléments de la liste donnée en argument..



Les éléments de la liste doivent autoriser la multiplication (nombres réels, nombres complexes, expressions algébriques...).

Σ LIST



Somme des éléments d'une liste

Accès (HP-48 G/GX) : [MTH] (LIST) (Σ LIST)

Drapeaux : -

| | | | | |
|----|-----------|---|----|-------|
| 1: | { liste } | → | 1: | somme |
|----|-----------|---|----|-------|

Soit { ob_1 ob_2 ob_3 ... ob_n-1 ob_n } une liste de n éléments donnée en argument à Σ LIST, Σ LIST renvoie somme qui correspond à la somme ob_1+ob_2+ob_3+... ..+ob_n-1+ob_n} des n éléments de la liste donnée en argument..

Les éléments de la liste doivent autoriser l'addition (nombres réels, nombres complexes, expressions algébriques...).



LIST→

Répartition dans la pile des éléments d'une liste

Accès : [α] [α] [NXT] [CST] [SIN] [COS]

Drapeaux : -

| | | | | |
|------|------------------------|---|-----|------|
| n+1: | | → | n: | ob_n |
| ... | | | ... | |
| 3: | | | 3: | ob_3 |
| 2: | | | 2: | ob_2 |
| 1: | { ob_1 ob_2 ... ob_n } | | 1: | ob_1 |

LIST→ prend une liste de n objets ob_1 à ob_n et les place aux niveaux 1 à n de la pile. L'objet ob_1 est placé au niveau 1, l'objet ob_2 l'est au niveau 2 et ainsi de suite jusqu'à ob_n qui est placé au niveau n.

OBJ→ a le même effet que LIST→ quand il est appliqué à une liste.



LN

Logarithme népérien

Accès : [Γ] [1/x]

Drapeaux : -1, -3, -22

| | | | | |
|----|-------------|---|----|-----------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'LN(expr_symb)' |

LN renvoie le logarithme népérien (logarithme de base e) de son argument. Voir aussi LNP1.

LNP1
Logarithme népérien plus 1

Accès (HP-48 G/GX) : [MTH] (HYP) [NXT] (LNP1) Drapeaux : -1, -3, -22

| | | | | |
|----|-------------|---|----|-------------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'LNP1(expr_symb)' |

LNP1 renvoie le logarithme népérien (logarithme de base e) de son argument augmenté de 1. Voir aussi LN. L'emploi de LNP1 présente un intérêt pour les valeurs de l'argument proches de 0.

LOG
Logarithme décimal

Accès : [↵] [y^x] Drapeaux : -1, -3, -22

| | | | | |
|----|-------------|---|----|------------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'LOG(expr_symb)' |

LOG correspond au calcul du logarithme décimal (logarithme de base 10 aussi appelé logarithme commun). On a $\log x = \ln x / \ln 10$.

Voir aussi LN et ALOG.



LOGFIT
Modèle logarithmique pour les calculs de régression

Accès (HP-48 G/GX) : [↵] [5] (ΣPAR) (MODL) (LOGFI) Drapeaux : -
 Accès (HP-48 S/SX) : [↵] [5] [NXT] [NXT] [NXT] (MODL) (LOG)

LOGFIT modifie ΣPAR (qui est une variable réservée) de façon à appliquer le modèle de régression logarithmique lors des exécutions de l'instruction LR.

LQ
Factorisation d'une matrice

Accès (HP-48 G/GX) : [MTH] (MATR) (FACTR) (LQ) Drapeaux : - →





| | | |
|--|---|--|
| 3: 2: 1: [[matrice_arg]] | → | 3: [[matrice_1]] 2: [[matrice_2]] 1: [[matrice_3]] |
|--|---|--|

LQ factorise la matrice [[matrice_arg]] donnée en argument et renvoie trois matrices [[matrice_1]], [[matrice_2]] et [[matrice_3]] vérifiant l'égalité :

$$[[matrice_3]] \times [[matrice_arg]] = [[matrice_1]] \times [[matrice_2]].$$

LR

Régression linéaire

Accès (HP-48 G/GX) : [←] [5] (FIT) (LR) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] (LR)

| | | |
|----------|---|--|
| 2: 1: | → | 2: b 1: m |
|----------|---|--|

LR renvoie les paramètres de régression linéaire fondés sur les données de ΣDAT. b est l'ordonnée à l'origine alors que m est la pente de la droite de régression. Selon le modèle de régression choisi, voici comment m et b sont utilisés :

- régression exponentielle : $\ln(y) = m \cdot x + \ln(b)$,
- régression logarithmique : $y = m \cdot \ln(x) + b$,
- régression de puissance : $\ln(y) = m \cdot \ln(x) + \ln(b)$.

LSQ

Norme minimale d'un système d'équations linéaires



Accès (HP-48 G/GX) : [MTH] (MATR) (LSQ) Drapeaux : -54

| | | |
|--|---|---|
| 2: [vecteur_b] 1: [[matrice_a]] | → | 2: 1: [vecteur_x] |
|--|---|---|

| | | |
|--|---|---|
| 2: [[matrice_b]] 1: [[matrice_a]] | → | 2: 1: [[matrice_x]] |
|--|---|---|

LSQ calcule à l'aide de la méthode des moindres carrés la norme minimale d'un système d'équations linéaires :

$$[\text{vecteur_b}] \times [[\text{matrice_a}]] = [\text{vecteur_x}]$$

$$[[\text{matrice_b}]] \times [[\text{matrice_a}]] = [[\text{matrice_x}]]$$

Note : si [[matrice_a]] a plus de colonnes que de lignes, il existe une infinité de solutions. Inversement, si [[matrice_a]] a plus de lignes que de colonnes, le système peut n'avoir aucune solution.

LU



Décomposition LU d'une matrice carrée

Accès (HP-48 G/GX) : [MTR] (MATR) (FACTR) (LU)

Drapeaux : -

| | | | | |
|----|-----------------|---|----|---------------|
| 3: | | → | 3: | [[matrice_1]] |
| 2: | | | 2: | [[matrice_2]] |
| 1: | [[matrice_arg]] | | 1: | [[matrice_3]] |

A partir d'une matrice [[matrice_arg]], LU renvoie trois matrices [[matrice_1]], [[matrice_2]] et [[matrice_3]] vérifiant l'égalité $[[matrice_3]] \times [[matrice_arg]] = [[matrice_1]] \times [[matrice_2]]$. [[matrice_1]] est une matrice triangulaire inférieure alors que [[matrice_2]] est une matrice triangulaire supérieure.

MANT

Extraction de la mantisse

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] (MANT)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] (MANT)

| | | | | |
|----|-------------|---|----|-------------------|
| 1: | réel | → | 1: | réel |
| 1: | 'expr_symb' | → | 1: | 'MANT(expr_symb)' |

Un nombre réel est représenté par un réel (la mantisse), multiplié par un facteur 10 élevé à une puissance entière. Par exemple, 2.56E5 représente 2.56×10^5 , soit 256000. Dans le cadre de cet exemple, 2.56 est la mantisse et 5 est l'exposant du facteur 10. MANT renvoie la mantisse de son argument.

↓MATCH

Réécriture d'une expression algébrique

Accès (HP-48 G/GX) : [↵] [9] [NXT] (↓MAT)

Drapeaux : -

| | | | | |
|----|-----------------------|---|----|---------------------|
| 2: | 'exp_symb à réécrire' | → | 2: | 'exp_symb_réécrite' |
| 1: | {'conf' 'rempl'} | | 1: | entier |

↓MATCH réarrange l'expression symbolique placée au niveau 2 en se fondant sur la règle exposée par la liste placée au niveau 1. Cette liste contient deux expressions algébriques : 'conf' détermine le motif à rechercher et 'rempl' fixe l'expression qui doit remplacer le motif. On utilise le caractère '&' avant les variables à remplacer. Ainsi, pour appliquer $(a+b)^2 = a^2 + b^2 + 2.a.b$ à $(c+d)^2$, on place $(c+d)^2$ au niveau 2 de la pile puis { '&(a&b)^2' '&a^2+&b^2+2*&a*&b' } →



→ au niveau 1. Cette liste fixe la règle de réarrangement qui doit être appliquée à `'exp_symb_à_réécrire'`. Une fois \downarrow MATCH activée, on obtient `'exp_symb_réécrite'` au niveau 2 puis un entier au niveau 1. Ici, `'exp_symb_réécrite'` est `'c^2+d^2+2*c*d'`. L'entier au niveau 1 vaut 1 si l'expression a pu être réécrite, il vaut 0 si elle n'a pas été modifiée. La méthode descendante utilisée par \downarrow MATCH est efficace dans le cadre du développement d'une expression. La méthode ascendante utilisée par \uparrow MATCH convient plutôt aux simplifications. Voir aussi \uparrow MATCH.

\uparrow MATCH

Réécriture d'une expression algébrique

Accès (HP-48 G/GX) : [\leftarrow] [9] [NXT] (\uparrow MAT) Drapeaux : -

| | |
|----|------------------------------------|
| 2: | <code>'exp_symb_à_réécrire'</code> |
| 1: | <code>{'conf' 'rempl'}</code> |



| | |
|----|----------------------------------|
| 2: | <code>'exp_symb_réécrite'</code> |
| 1: | entier |

\uparrow MATCH réarrange l'expression symbolique placée au niveau 2 en se fondant sur la règle exposée par la liste placée au niveau 1. Cette liste contient deux expressions algébriques : `'conf'` détermine le motif à rechercher et `'rempl'` fixe l'expression qui doit remplacer le motif. On utilise le caractère `'&'` avant les variables à remplacer. Ainsi, pour appliquer $a^2+b^2+2.a.b=(a+b)^2$ à $c^2+d^2+2.cd$, on place `'c^2+d^2+2*c*d'` au niveau 2 de la pile puis `{ '&a^2+&b^2+2*&a*&b' '(&a+&b)^2' }` au niveau 1. Cette liste fixe la règle de réarrangement qui doit être appliquée à `'exp_symb_à_réécrire'`. Une fois \uparrow MATCH activée, on obtient `'exp_symb_réécrite'` au niveau 2 puis un entier au niveau 1. Ici, `'exp_symb_réécrite'` est `'(c+d)^2'`. L'entier au niveau 1 vaut 1 si l'expression a pu être réécrite, il vaut 0 si elle n'a pas été modifiée. La méthode descendante utilisée par \downarrow MATCH est efficace dans le cadre du développement d'une expression. La méthode ascendante utilisée par \uparrow MATCH convient, comme ici, aux simplifications. Voir aussi \downarrow MATCH.



MAX

Extraction de la plus grande valeur

Accès (HP-48 G/GX) : [MTH] (REAL) (MAX) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] (MAX)

| | |
|----|--------------------|
| 2: | <code>val_1</code> |
| 1: | <code>val_2</code> |



| | |
|----|------------------|
| 2: | |
| 1: | <code>Max</code> |

`val_1` et `val_2` pouvant être des réels, des expressions algébriques ou des objets unités, `Max` renvoie la plus grande des deux valeurs ou une expression algébrique la calculant.

MAXΣ

Extraction des valeurs maximales de la matrice statistique

Accès (HP-48 G/GX) : [←] [5] (1VAR) (MAXΣ) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] (MAXΣ)

| | | |
|----|---|----------------------------|
| 1: | → | 1: max |
| 1: | → | 1: [max_1 max_2 ... max_n] |

maxΣ extrait de la matrice statistiques ΣDAT les valeurs maximales placées dans chaque colonne. Si ΣDAT ne comporte qu'une seule colonne, MAXΣ renvoie un réel max. Si ΣDAT est une matrice de n colonnes, MAXΣ renvoie un vecteur à n éléments (chaque élément du vecteur est le plus grand élément de la colonne correspondante).

MAXR

Plus grand nombre pouvant être traité par la HP-48

Accès (HP-48 G/GX) : [MTH] [NXT] (CONS) [NXT] (MAXR) Drapeaux : -2, -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] [NXT] (MAXR)

| | | |
|----|---|-----------------------|
| 1: | → | 1: 9.999999999999E499 |
| 1: | → | 1: 'MAXR' |

MAXR est une constante représentant le plus grand nombre pouvant être traité par la HP-48. Si le drapeau (*flag*) 2 ou 3 est armé, MAXR renvoie la forme numérique de la constante. Dans le cas contraire, MAXR affiche 'MAXR' (forme numérique obtenue avec →NUM).

MCALC

Déclaration d'une variable comme valeur calculée



Accès (HP-48 G/GX) : [←] [3] (MES) (MCAL) Drapeaux : -

| | | |
|-----------------------------------|---|----|
| 1: 'nom' | → | 1: |
| 1: { 'nom_1' 'nom_2'... 'nom_n' } | → | 1: |
| 1: "ALL" | → | 1: |

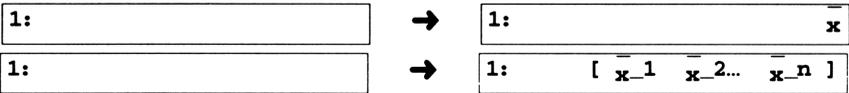
MCALC déclare la variable nommée 'nom' comme valeur calculée par le solveur d'équations. Une liste de noms peut être utilisée en argument. Si la chaîne de caractères "ALL" est l'argument de MCALC, toutes les variables sont déclarées comme valeurs calculées.

MEAN

Moyenne arithmétique des données de Σ_{DAT}

Accès (HP-48 G/GX) : [\leftarrow] [5] (1VAR) (MEAN) Drapeaux : -

Accès (HP-48 S/SX) : [\leftarrow] [5] [NXT] (MEAN)



MEAN calcule la moyenne arithmétique du contenu de chaque colonne de la matrice statistique Σ_{DAT} (pour chaque colonne, la somme des éléments est divisée par le nombre d'éléments). Si Σ_{DAT} ne contient qu'une seule colonne, seule une moyenne \bar{x} est renvoyée. Si Σ_{DAT} contient n colonnes, MEAN renvoie un vecteur [\bar{x}_1 \bar{x}_2 ... \bar{x}_n] rassemblant les moyennes de chaque colonne.

MEM

Mémoire disponible

Accès : [\leftarrow] [VAR] (MEM) Drapeaux : -



MEM renvoie la mémoire disponible exprimée en octets. L'unité d'information de la HP-48 étant le quartet (demi-octet), ne vous étonnez pas si mem est une valeur décimale, par exemple 18495.5. L'opération de *garbage collection* (aussi appelée glanage de cellules ou récupération de mémoire) permet de récupérer la mémoire attribuée aux objets devenus inutiles. La *garbage collection* s'effectue automatiquement dès que la mémoire manque mais MEM permet de forcer la HP-48 à effectuer une *garbage collection* prématurément.



MENU

Affichage d'un menu intégré ou personnalisé

Accès (HP-48 G/GX) : [\leftarrow] [CST] (MENU) (MENU) Drapeaux : -

Accès (HP-48 S/SX) : [\rightarrow] [CST] (MENU)



MENU affiche un menu intégré identifié par son numéro num_menu (voir la table des menus intégrés) ou un menu défini par une liste associant les articles du menu à des commandes.

MERGE**Fusion de la carte en port 1 avec la RAM interne**Accès (HP-48 GX) : [α] [α] [$'$] [E] [\rightarrow] [MTH] [E] **Drapeaux** : -Accès (HP-48 SX) : [\leftarrow] [VAR] [NXT] [NXT] (MERG)

| | | | |
|----|---|---------------|----|
| 1: | 1 | \rightarrow | 1: |
|----|---|---------------|----|

MERGE provoque la fusion la mémoire vive et de la carte RAM en port 1. Sur HP-48 GX, utilisez plutôt MERGE1. Voir MERGE1.

MERGE1**Fusion de la carte en port 1 avec la RAM interne**Accès (HP-48 GX) : [\leftarrow] [2] (MERG) **Drapeaux** : -

MERGE provoque la fusion la mémoire vive et de la carte RAM en port 1 (en fait, il s'agit d'une partition de 128 ko maximum de la carte en port 1). Sur HP-48 SX, utilisez MERGE. Voir MERGE.

MIN**Extraction de la plus petite valeur**Accès (HP-48 G/GX) : [MTH] (REAL) (MIN) **Drapeaux** : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] (MIN)

| | | | | |
|----|-------|---------------|----|-----|
| 2: | val_1 | \rightarrow | 2: | Min |
| 1: | val_2 | | 1: | |



val_1 et val_2 pouvant être des réels, des expressions algébriques ou des objets unités, Min renvoie la plus petite des deux valeurs ou une expression algébrique la calculant.

MIN Σ **Extraction des valeurs minimales de la matrice statistique**Accès (HP-48 G/GX) : [\leftarrow] [5] (1VAR) (MIN Σ) **Drapeaux** : -Accès (HP-48 S/SX) : [\leftarrow] [5] [NXT] (MIN Σ)

| | | | |
|----|---------------|----|-----|
| 1: | \rightarrow | 1: | min |
|----|---------------|----|-----|

| | | | |
|----|---------------|----|------------------------|
| 1: | \rightarrow | 1: | [min_1 min_2... min_n] |
|----|---------------|----|------------------------|

min Σ extrait de la matrice statistiques Σ_{DAT} les valeurs minimales placées dans chaque colonne. \rightarrow

→ Si Σ_{DAT} ne comporte qu'une seule colonne, $\text{Min}\Sigma$ renvoie un réel min. Si Σ_{DAT} est une matrice de n colonnes, $\text{Min}\Sigma$ renvoie un vecteur à n éléments (chaque élément du vecteur est le plus petit élément de la colonne correspondante).

MINEHUNT



Jeu du démineur

Accès (HP-48 G/GX) : [←] [3] (UTILS) (MINE) Drapeaux : -



MINEHUNT lance le jeu du démineur intégré à la bibliothèque d'équations.

MINIT



Initialisation du menu d'équations multiples

Accès (HP-48 G/GX) : [←] [3] (MES) (MINIT) Drapeaux : -

MINIT crée M_{par} à partir de Eq. M_{par} et Eq sont des variables réservées.

MINR

Plus petit nombre pouvant être traité par la HP-48

Accès (HP-48 G/GX) : [MTH] [NXT] (CONS) [NXT] (MINR) Drapeaux : -2, -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] [NXT] (MINR)

1: → 1: 1.00000000000E-499



MINR est une constante représentant le plus petit nombre pouvant être traité par la HP-48. Si le drapeau (*flag*) 2 ou 3 est armé, MINR renvoie la forme numérique de la constante. Dans le cas contraire, MINR affiche 'MINR' (forme numérique obtenue avec →NUM).

MITM



Modification du menu d'équations multiples

Accès (HP-48 G/GX) : [←] [3] (MES) (MITM) Drapeaux : -

2: "nom" → 2:
1: { list_var } 1:



MITM change l'ordre et les titres du menu d'équations multiples. { list_var } est la liste des noms des variables.

MOD

Modulo

Accès (HP-48 G/GX) : [MTH] (REAL) (MOD)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] (MOD)

| | |
|----|--------|
| 2: | réel_1 |
| 1: | réel_2 |



| | |
|----|--------|
| 2: | |
| 1: | réel_3 |

| | |
|----|--------------|
| 2: | 'exp_symb_1' |
| 1: | 'exp_symb_2' |



| | |
|----|-------------------------------|
| 2: | |
| 1: | 'MOD(exp_symb_1, exp_symb_2)' |



Si les arguments de Mod sont réel_1 et réel_2, Mod renvoie réel_3 qui correspond à $\text{réel}_1 \bmod \text{réel}_2$, c'est-à-dire $\text{réel}_1 - \text{réel}_2 \cdot \text{floor}(\text{réel}_1 / \text{réel}_2)$ - en RPN.

MROOT

Résolutions d'équations multiples



Accès (HP-48 G/GX) : [←] [3] (MES) (MROO)

Drapeaux : -

| | |
|----|-------|
| 1: | 'var' |
|----|-------|



| | |
|----|-----|
| 1: | res |
|----|-----|

| | |
|----|-------|
| 1: | "ALL" |
|----|-------|



| | |
|----|--|
| 1: | |
|----|--|

MROOT se fonde sur MPar (variable réservée contenant des équations) pour résoudre (à l'aide du solveur) une ou plusieurs inconnues. MROOT affecte leurs valeurs aux variables correspondant aux inconnues calculées. Si un nom de variable (inconnue) 'var' est donné en argument, MROOT renvoie res, valeur calculée de cette inconnue. Si l'argument est "ALL", MROOT calcule toutes les inconnues et place les valeurs calculées dans les variables correspondantes.

MSGBOX

Affichage d'un message dans une zone de dialogue



Accès (HP-48 G/GX) : [PRG] [NXT] (OUT) (MSGB)

Drapeaux : -

| | |
|----|---------|
| 1: | "texte" |
|----|---------|



| | |
|----|--|
| 1: | |
|----|--|

MSGBOX affiche "texte" dans une zone de dialogue (boîte de dialogue, *dialog box*). La chaîne de caractères "texte" est limitée à 75 caractères parmi lesquels peuvent se trouver des retours à la ligne ([↵]) et des espaces ([SPC]). Lors de l'exécution du programme, une pression sur (OK) permet de quitter la zone de dialogue.

MSOLVR



Utilisation des variables du solveur d'équations multiples

Accès (HP-48 G/GX) : [\leftarrow] [3] (MES) (MSOL) Drapeaux : -

MSOLVR applique les variables du solveur d'équations multiples aux équations définies via la variable réservée Mpar.

MUSER



Déclaration d'une variable utilisateur

Accès (HP-48 G/GX) : [\leftarrow] [3] (MES) (MUSE) Drapeaux : -

| | | | | |
|----|-------------------------|---|----|--|
| 1: | 'nom_var' | → | 1: | |
| 1: | { 'nom_1' 'nom_2' ... } | → | 1: | |
| 1: | "ALL" | → | 1: | |

Dans le cadre du solveur d'équations multiples, MUSER déclare une ou plusieurs variables comme variables utilisateur. 'nom_var' est un nom de variable. Vous pouvez aussi utiliser une liste de noms de variables ou même "ALL" qui indique que toutes les variables utilisées par le solveur d'équations multiples sont des variables-utilisateur.

NDIST



Distribution normale (probabilités)

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) [NXT] (NDIST) Drapeaux : -

| | | | | |
|----|----------|---|----|---------|
| 3: | moyenne | → | 3: | |
| 2: | variance | | 2: | |
| 1: | x | | 1: | Distrib |

NDIST calcule Distrib qui est la distribution de probabilité normale en x fondée sur la variance variance de la distribution normale et la moyenne moyenne.

NEG

recherche de l'opposé

Accès (HP-48 G/GX) : [+/-] ou [MTH] [NXT] (Cmpl) [NXT] (NEG) Drapeaux : -3, -5, -6, -7, -8, -9, -10

Accès (HP-48 S/SX) : [+/-]

| | | | | |
|----|------|---|----|------|
| 1: | réel | → | 1: | réel |
|----|------|---|----|------|



Index alphabétique RPL



| | | | | |
|----|------------------------|---|----|-------------------------|
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'NEG(expr_symb)' |
| 1: | [[tableau]] | → | 1: | [[tableau]] |
| 1: | objet_graphique | → | 1: | objet_graphique |
| 1: | entier_binaire | → | 1: | entier_binaire |

→ NEG renvoie l'opposé de son argument. NEG peut s'appliquer aux nombres réels, aux objets-unités, aux complexes, aux expressions symboliques, etc. Si l'argument est un entier binaire (entier système de la forme #nb, #no, #nd ou #nh), l'emploi du signe «-» est impossible et NEG renvoie donc le complément à 2 (tous les bits codant l'entier binaire sont inversés puis on ajoute 1). Si l'argument est un tableau (matrice ou vecteur), NEG renvoie un tableau dont tous les éléments sont les opposés des éléments correspondant dans le tableau donné en argument. Si l'argument est un objet graphique (GROB), NEG renvoie l'objet graphique «négatif», autrement dit, NEG inverse l'état de tous les pixels de l'objet graphique.

NEWOB

Copie d'un objet et suppression de l'original

Accès (HP-48 G/GX) : [←] [VAR] (NEWO) Drapeaux : -55

Accès (HP-48 S/SX) : [←] [VAR] [NXT] (NEWO)

| | | | | |
|----|---------------------|---|----|---------------------|
| 1: | selon le cas | → | 1: | selon le cas |
|----|---------------------|---|----|---------------------|

NEWOB provoque la création d'un nouvel objet. Ainsi, utiliser NEWOB permet de créer un nouvel objet à partir d'un élément extrait d'une liste (ou d'un tableau, ou de tout objet composite) et de supprimer cette liste en mémoire (à condition que le *flag 55* soit armé). NEWOB peut aussi prendre en argument une bibliothèque ou une sauvegarde (objet-sauvegarde), NEWOB crée alors une copie renvoyée au niveau 1 et provoque la suppression de l'objet original (si le *flag 55* est armé).

NEXT

Fin d'une boucle commencée par FOR ou START

Accès (HP-48 G/GX) : [PRG] (BRCH) (START) (BRCH) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] (NEXT)



NEXT marque la fin d'une boucle à compteur du type FOR... NEXT ou START... NEXT. Voir FOR et START.

NOT

Inversion logique

Accès (HP-48 G/GX) : [PRG] (TEST) [NXT] (NOT) **Drapeaux** : -3, -5, -6, -7, -8, -9, -10

Accès (HP-48 S/SX) : [PRG] (TEST) (NOT)

| | | | | |
|----|----------------|---|----|------------------|
| 1: | entier_binaire | → | 1: | entier_binaire |
| 1: | expr_booléenne | → | 1: | 0 ou 1 |
| 1: | chaîne | → | 1: | chaîne |
| 1: | 'expr_symb' | → | 1: | 'NOT(expr_symb)' |



Un entier binaire (entier système) est composé d'une série de bits représentés en base 2, 8, 10 ou 16 selon le choix de l'utilisateur. NOT inverse tous les bits de l'entier binaire qui lui est donné en argument. NOT inverse le résultat d'un test (expression booléenne) devant renvoyer *vrai* ou *faux* (ces états sont représentés par 0 et 1). Dans le cas d'une chaîne, NOT inverse chacun des huit bits codant chaque caractère de la chaîne.

NOVAL

Absence de valeur par défaut dans une zone de dialogue



Accès (HP-48 G/GX) : [PRG] [NXT] (IN) (NOVA) **Drapeaux** : -

Dans le cadre de la création avec INFORM d'une zone de dialogue avec champs de saisie, NOVAL peut être placé comme valeur par défaut d'un champ. NOVAL symbolise l'absence de valeur par défaut.

NΣ

Nombre de lignes de la matrice statistique courante



Accès (HP-48 G/GX) : [↵] [5] (SUMS) (NΣ) **Drapeaux** : -

| | | | | |
|----|--|---|----|--------|
| 1: | | → | 1: | entier |
|----|--|---|----|--------|



NΣ affiche un entier naturel correspondant au nombre de lignes que contient la matrice statistique ΣDAT.

NSUB

Numéro du bloc traité par DOSUBS



Accès (HP-48 G/GX) : [PRG] (LIST) (PROC) (NSUB) **Drapeaux** : -

| | | | | |
|----|--|---|----|--------|
| 1: | | → | 1: | entier |
|----|--|---|----|--------|



→ NSUB renvoie dans la pile un entier naturel qui correspond au numéro du bloc en cours de traitement par DOSUBS. Voir DOSUBS.

NUM

Code ASCII du premier caractère d'une chaîne

Accès (HP-48 G/GX) : [PRG] (TYPE) [NXT] (NUM) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] (NUM)



| | | | | |
|----|----------|---|----|--------|
| 1: | "chaîne" | → | 1: | entier |
|----|----------|---|----|--------|

NUM place au niveau 1 de la pile le code ASCII (ASCII HP-48) du premier caractère de la chaîne de caractères chaîne. Ce code est un entier naturel compris entre 0 et 255.

→NUM

Evaluation fournissant un résultat symbolique

Accès (HP-48 G/GX) : [←] [EVAL] Drapeaux : -

Accès (HP-48 G/GX) : [→] [EVAL]

| | | | | |
|----|-------------|---|----|--------------------|
| 1: | 'expr_symb' | → | 1: | résultat numérique |
|----|-------------|---|----|--------------------|

| | | | | |
|----|-------|---|----|------------------|
| 1: | 'nom' | → | 1: | valeur numérique |
|----|-------|---|----|------------------|



→NUM place au niveau 1 de la pile la résultat numérique fournit par l'évaluation d'une expression symbolique. Si l'argument est le nom d'une variable contenant un nombre, ce nombre est renvoyé.

NUMX

Nombre de points représentés pour la variable indépendante

Accès (HP-48 G/GX) : [←] [8] [NXT] (3D) (VPAR) [NXT] (NUMX) Drapeaux : -

| | | | | |
|----|--------|---|----|--|
| 1: | entier | → | 1: | |
|----|--------|---|----|--|

NUMX prend en argument un entier supérieur ou égal à 2. Celui-ci correspond au nombre de points tracés associés à la variable indépendante pour chaque point tracé de la variable dépendante dans le cadre des tracés 3D avec vue en perspective. La valeur définie avec NUMX est conservée dans la variable réservée VPAR et est utilisée par les tracés 3D (sauf YSLICE).

Voir aussi NUMY.



NUMY



Nombre de points représentés pour la variable dépendante

Accès (HP-48 G/GX) : [\leftarrow] [8] [NXT] (3D) (V_{PAR}) [NXT] (NUMY) Drapeaux : -

| | | | |
|----|--------|---|----|
| 1: | entier | → | 1: |
|----|--------|---|----|

NUMY prend en argument un entier supérieur ou égal à 2. Celui-ci correspond au nombre de points tracés associés à la variable dépendante dans le cadre des tracés 3D avec vue volumique. La valeur définie avec NUMY est conservée dans la variable réservée V_{PAR} et est utilisée par les tracés 3D. Voir aussi NUMX.

OBJ→

Dissociation des éléments composant un objet composite

Accès (HP-48 G/GX) : [PRG] (TYPE) (OBJ→) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) (OBJ→)



| | | | | |
|----|----------|---|----|-------------------|
| 2: | | → | 2: | partie réelle |
| 1: | complexe | | 1: | partie imaginaire |

| | | | | |
|------|---------------------------|---|------|---------|
| n+1: | | → | n+1: | obj_1 |
| ... | | | ... | |
| 3: | | → | 3: | obj_n-1 |
| 2: | | | 2: | obj_n |
| 1: | { obj_1 obj_2 ... obj_n } | | 1: | n |

| | | | | |
|--------|-------------|---|--------|---------|
| 1xc+1: | | → | 1xc+1: | elt_1.1 |
| ... | | | ... | |
| 2: | | → | 2: | elt_1.c |
| 1: | [[tableau]] | | 1: | { 1 c } |

| | | | | |
|------|-------------|---|------|-------|
| n+2: | | → | n+2: | arg_1 |
| ... | | | ... | |
| 3: | | → | 3: | arg_n |
| 2: | | | 2: | n |
| 1: | 'expr_symb' | | 1: | 'fct' |

| | | | | |
|----|-------------|---|----|---------|
| 2: | | → | 2: | réel |
| 1: | objet_unité | | 1: | 1_unité |

| | | | | |
|----|----------|---|----|-----|
| 2: | | → | 2: | obj |
| 1: | :tag:obj | | 1: | tag |

| | | | | |
|----|----------|---|----|--------------|
| 1: | "chaîne" | → | 1: | selon le cas |
|----|----------|---|----|--------------|

OBJ→ décompose un objet composite en éléments plus élémentaires. Si l'argument est un complexe, OBJ→ affiche sa partie réelle puis sa partie imaginaire. Si l'argument est une liste { obj_1 obj_2 ... obj_n } de →

→ n éléments, OBJ→ place n au niveau 1 puis les objets obj_1 à obj_ n entre les niveaux $n+1$ et 2 de la pile. Si l'argument est un tableau (vecteur ou matrice) comportant 1 lignes et c colonnes, OBJ→ affiche une liste { 1 c } au niveau 1 et place les éléments elt_1.1 à elt_1. c entre les niveaux $1 \times c + 1$ et 2 de la pile. Si l'argument est une expression symbolique, OBJ→ affiche entre les niveaux $n+2$ et 3 les arguments de l'expression symbolique en respectant l'ordre d'évaluation des arguments et des sous-expressions. Le nombre n d'arguments est placé au niveau 2 alors que le nom de la fonction de plus haut niveau est placé au niveau 1. Si l'argument de OBJ→ est un objet «taggué» ou un objet-unité, OBJ→ dissocie les deux parties de l'objet. Selon le cas, on obtient donc un réel et une unité ou un objet et son ancienne étiquette. Si l'argument est une chaîne, son contenu est évalué.

OCT

Activation du mode de calcul octal

Accès : [MTH] (BASE) (OCT) Drapeaux : -5, -6, -7, -8, -9, -10, -11, -12

OCT active la base 8 (octal) pour la représentation des entiers binaires. Les entiers binaires sont ainsi représentés suivis de «o» (et précédés de «#» comme toujours). Voir aussi BIN, DEC et HEX.

OFF

Extinction

Accès (HP-48 G/GX) : [PRG] [NXT] (RUN) [NXT] (OFF) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] [NXT] (OFF)



OFF éteint la calculatrice. Placée dans un programme, l'instruction OFF provoque le redémarrage du programme dès le prochain allumage.

OLDPRT

Adaptation du jeu de caractère à celui de la HP 82240A

Accès (HP-48 G/GX) : [↵] [1] (PRINT) (PRTPA) (OLDPR) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [MTH] [NXT] (OLDPR)

OLDPRT est à utiliser avant d'imprimer du texte à l'aide de l'imprimante infrarouge HP 82240A. En effet, OLDPRT adapte le codage des caractères afin de rendre ceux de la HP-48 compatibles avec ceux de la HP 82240A. Attention ! OLDPRT ne doit pas être utilisé si vous imprimez du graphisme.



OPENIO

Ouverture du port IR ou du port série

Accès (HP-48 G/GX) : [←] [1] [NXT] (SERIA) (OPENI) Drapeaux : -33

Accès (HP-48 S/SX) : [←] [PRG] [NXT] (OPENI)



OPENIO ouvre le port série (infrarouge ou câble) en se fondant sur les paramètres contenus dans IOPAR. Notez que toutes les commandes KERMIT comprennent l'exécution de OPENIO qui ne doit donc être utilisé que dans quelques cas particuliers.

OR

Ou logique

Accès (HP-48 G/GX) : [PRG] (TEST) [NXT] (OR) Drapeaux : -3, -5, -6, -7, -8, -9, -10

Accès (HP-48 S/SX) : [PRG] (TEST) (OR)

| | |
|----|----------------|
| 2: | entier_binaire |
| 1: | entier_binaire |



| | |
|----|----------------|
| 2: | |
| 1: | entier_binaire |

| | |
|----|----------|
| 2: | "chaîne" |
| 1: | "chaîne" |



| | |
|----|----------|
| 2: | |
| 1: | "chaîne" |

| | |
|----|-----------|
| 2: | expr_bool |
| 1: | expr_bool |



| | |
|----|--------|
| 2: | |
| 1: | 0 ou 1 |

| | |
|----|--------------|
| 2: | 'expr_symb1' |
| 1: | 'expr_symb2' |



| | |
|----|----------------------------|
| 2: | |
| 1: | 'expr_symb1 OR expr_symb2' |



OR correspond au OU logique. Lorsque les arguments sont des entiers binaires, OR renvoie un entier binaire dont les bits à 1 correspondent à des bits à 1 dans l'un ou l'autre des entiers binaires donnés en arguments. De même, quand les arguments sont des chaînes de caractères, OR considère les huit bits codant chaque caractère et produit une nouvelle chaîne dont les caractères sont créés à partir des bits armés dans l'une ou l'autre des chaînes données en argument. OR peut aussi prendre en argument un test (à la suite de IF par exemple) ou une expression booléenne.

ORDER

Classement des variables du répertoire courant

Accès (HP-48 G/GX) : [←] [VAR] (DIR) (ORDER) Drapeaux : -

Accès (HP-48 S/SX) : [←] [VAR] (ORDER)

| | |
|----|------------------------|
| 1: | { 'nom_1'... 'nom_2' } |
|----|------------------------|



| | |
|----|--|
| 1: | |
|----|--|



→ ORDER prend en argument une liste contenant des noms de variables ou de sous-répertoires se trouvant dans le répertoire courant (affichez-les en appuyant sur [VAR]). ORDER classe les variables dans le répertoire en respectant l'ordre des noms dans la liste donnée en argument. Il n'est pas indispensable de citer dans la liste toutes les variables et tous les sous-répertoires du répertoire courant. Les variables et sous-répertoires non cités dans la liste seront placés à la suite de ceux cités dans la liste.

OVER

Copie du contenu du niveau 2 de la pile

Accès (HP-48 G/GX) : [←] [↑] (OVER)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (STK) (OVER)

| | |
|----|---------|
| 3: | |
| 2: | objet_a |
| 1: | objet_b |



| | |
|----|---------|
| 3: | objet_a |
| 2: | objet_b |
| 1: | objet_a |



OVER copie au niveau 1 l'objet placé au niveau 2. Ne confondez pas DUP qui duplique l'objet du niveau 1 avec OVER qui duplique l'objet placé au niveau 2. Voir aussi DUP.

PARAMETRIC

Activation du mode de tracé paramétrique

Accès : [←] [8] (PTYPE) (PARA)

Drapeaux : -28, -31

PARAMETRIC active le mode de tracé paramétrique. Après l'exécution de PARAMETRIC, DRAW représente graphiquement l'équation paramétrique placée dans EQ.

PARITY

Définition du mode de contrôle de la parité (liaison série)

Accès (HP-48 G/GX) : [←] [1] (IOPAR) (PARIT)

Drapeaux : -

Accès (HP-48 S/SX) : [←] [PRG] (SETUP) (PARIT)

| | |
|----|--------|
| 1: | parité |
|----|--------|



| | |
|----|--|
| 1: | |
|----|--|

Dans le cadre des transmissions par la voie série, parité est un entier compris entre 0 et 4 inclus qui détermine le mode de contrôle de la parité (moyen primitif de détection d'erreur). Si parité vaut 0, il n'y a pas de contrôle, si parité vaut 1 on contrôle la parité impaire, si parité vaut 2 on contrôle la parité paire, si parité vaut 3 on se place en mode «mark», si parité vaut 4 il s'agit du mode «espace».

PARSURFACE

Activation du mode de tracé 3 vecteurs à 2 variables

Accès : [\leftarrow] [8] [NXT] (3D) (PTYPE) (PARSU) Drapeaux : -

PARSURFACE active le mode de tracé permettant à DRAW de représenter graphiquement une fonction de trois vecteurs à deux variables.

PATH

Chemin d'accès au répertoire courant depuis la racine

Accès (HP-48 G/GX) : [\leftarrow] [VAR] (DIR) (PATH) Drapeaux : -Accès (HP-48 S/SX) : [\leftarrow] [VAR] (PATH)1: → 1: { HOME rep_1... rep_courant }

Rappel : On accède à un répertoire en évaluant un chemin d'accès. Un chemin d'accès est la liste des répertoires à ouvrir pour accéder au répertoire qui nous intéresse. Un chemin d'accès commence à partir de la racine (HOME) puis énumère chacun des répertoires qu'il faut ouvrir.



PATH place dans la pile une liste correspondant au chemin d'accès au répertoire courant. Cette liste commence donc par HOME et finit par le nom du répertoire courant.

PCOEF

Commande de calcul des coefficients d'un polynôme unitaire

Accès (HP-48 G/GX) : [\leftarrow] [7] (POLY) (PCOEF) Drapeaux : -1: [vecteur] → 1: [vecteur]

A partir d'un vecteur contenant les racines (réelles ou complexes) d'un polynôme, PCOEF renvoie un autre vecteur contenant les coefficients (réels ou complexes) d'un polynôme unitaire correspondant aux racines données en argument.

Les coefficients sont rangés par ordre décroissant dans le vecteur renvoyé, le coefficient dominant valant 1.

Par exemple, si [-1 1] est donné en argument à PCOEF, cette instruction renvoie [1 0 -1], autrement dit, le polynôme $P(x) = 1 \cdot x^2 + 0 \cdot x - 1$ accepte -1 et 1 comme racines.

PCONTOUR

S/SX

Mode de tracé d'une fonction scalaire à deux variables

Accès : [←] [8] [NXT] (3D) (PTYPE) (PCON) Drapeaux : -

PCONTOUR active le mode de tracé permettant à DRAW de représenter graphiquement un contour d'une fonction scalaire à deux variables.

PCOV

S/SX

Covariance de population

Accès (HP-48 G/GX) : [←] [5] (FIT) [NXT] (PCOV) Drapeaux : -

| | | |
|----|---|-------------------|
| 1: | → | 1: covariance_pop |
|----|---|-------------------|

PCOV calcule la covariance de population en se fondant sur la colonne de données indépendantes et sur la colonne de données dépendantes de la matrice statistique ΣDAT.

PDIM

Remplacement de PICT par un objet graphique vierge

Accès (HP-48 G/GX) : [PRG] (PICT) (PDIM) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [8] [NXT] [NXT] (PDIM)

| | | |
|---------------|---|----|
| 2: (x_1, y_1) | → | 2: |
| 1: (x_2, y_2) | | 1: |

| | | |
|------------|---|----|
| 2: largeur | → | 2: |
| 1: hauteur | | 1: |

PDIM remplace PICT par un objet graphique vierge. Si les arguments sont (x_1, y_1) et (x_2, y_2) , la variable réservée PPAR est modifiée et les coordonnées utilisateur deviennent telles que les abscisses sont représentées sur l'intervalle $[x_1, x_2]$ et les ordonnées sur $[y_1, y_2]$. largeur et hauteur sont des entiers binaires (entiers systèmes). Utiliser largeur et hauteur comme arguments ne modifie pas PPAR mais remplace PICT par un objet graphique de dimensions largeurxhauteur. Il faut que $131 \leq \text{largeur} \leq 2048$ et $\text{hauteur} \geq 64$. Par exemple, pour remplacer PICT par un objet graphique vierge de 200 pixels de largeur sur 100 pixels de hauteur, il faut fournir #200d et #100d en arguments à PDIM.



Sur HP-48 S/SX, PDIM peut aussi être utilisé en mode direct pour placer largeur et hauteur dans la pile (dimensions du GROB PICT courant). Pour cela, saisissez [↵] [8] [NXT] [NXT] [↵] (PDIM).

PERM

Permutations

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) (PERM)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PROB) (PERM)

| | |
|----|---|
| 2: | n |
| 1: | p |



| | |
|----|--------|
| 2: | |
| 1: | entier |

| | |
|----|---------------|
| 2: | 'expr_symb_1' |
| 1: | 'expr_symb_2' |



| | |
|----|---------------------------------|
| 2: | |
| 1: | 'PERM(expr_symb_1,expr_symb_2)' |



PERM renvoie le nombre de permutation (A_n^p). Par exemple, combien de listes de deux éléments peut-on constituer à partir d'un ensemble comportant trois éléments ? Pour le savoir, il faut calculer A_3^2 . Pour cela, tapez [3] [ENTER] [2] [ENTER] [MTH] ([NXT] sur HP-48 G/GX) (PROB) (PERM).

PEVAL

Valeur d'un polynôme pour une valeur donnée de la variable



Accès (HP-48 G/GX) : [←] [7] (POLY) (PEVAL)

Drapeaux : -

| | |
|----|-----------|
| 2: | [vecteur] |
| 1: | x |



| | |
|----|------|
| 2: | |
| 1: | P(x) |



PEVAL calcule la valeur prise par un polynôme P défini par P(x) pour une valeur x de la variable indépendante. Le vecteur [vecteur] contient les n+1 coefficients du polynôme de degré n classés par ordre de degré décroissant. Bien entendu, les coefficients nuls ne doivent pas être omis. Ainsi, P(x)=x²-1 est défini par le vecteur [2 0 -1]. PEVAL affiche la valeur de P(x) au niveau 1 de la pile.

PGDIR

Suppression d'un répertoire et de ce qu'il contient

Accès (HP-48 G/GX) : [←] [VAR] (DIR) (PGDIR)

Drapeaux : -

Accès (HP-48 S/SX) : [←] [VAR] [NXT] [NXT] (PGDIR)

| | |
|----|-----------|
| 1: | 'nom_rep' |
|----|-----------|



| | |
|----|--|
| 1: | |
|----|--|

Le répertoire dont le nom est donné en argument est supprimé ainsi que son éventuel contenu.

Attention ! Ne confondez pas CLEAR (suppression du contenu de la pile), PURGE (suppression d'une variable), et PGDIR.

PICK

Copie au niveau 1 d'un niveau spécifié de la pile

Accès (HP-48 G/GX) : [←] [↑] (PICK)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (STK) (PICK)

| | |
|------|-------|
| n+1: | obj_n |
| ... | |
| 3: | obj_2 |
| 2: | obj_1 |
| 1: | n |



| | |
|------|-------|
| n+1: | obj_n |
| ... | |
| 3: | obj_2 |
| 2: | obj_1 |
| 1: | obj_n |



n étant l'argument utilisé par PICK, PICK copie au niveau 1 de la pile le contenu du niveau n+1.

PICT

Place PICT au niveau 1 de la pile

Accès (HP-48 G/GX) : [PRG] (PICT) (PICT)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) (PICT)

| | |
|----|--|
| 1: | |
|----|--|



| | |
|----|------|
| 1: | PICT |
|----|------|

PICT place PICT au niveau 1 de la pile. PICT est la variable réservée contenant l'affichage graphique (représentations graphiques, etc.).

PICTURE

Activation de l'environnement graphique



Accès (HP-48 G/GX) : [←] [←]

Drapeaux : -

Exécutée depuis un programme, l'instruction PICTURE provoque l'affichage de l'environnement graphique et attend une pression sur [ON].

PINIT

Initialisation des ports ouverts



Accès (HP-48 G/GX) : [←] [2] [NXT] (PINIT)

Drapeaux : -

PINIT ne modifie ni n'efface les données relatives à un port mais initialise tous les ports actifs, en effet, sur HP-48 GX, les cartes RAM contiennent un volume de mémoire multiple de 128 ko et divisé en partitions de 128 ko assimilées à des ports.

PIX?**Test de l'état d'un pixel**

Accès (HP-48 G/GX) : [PRG] (PICT) [NXT] (PIX?)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] (PIX?)

| | | | | |
|----|-------|---|----|------|
| 1: | (x,y) | → | 1: | état |
|----|-------|---|----|------|

| | | | | |
|----|-----------|---|----|------|
| 1: | { #x #y } | → | 1: | état |
|----|-----------|---|----|------|

PIX? prend comme argument les coordonnées d'un pixel (point de l'écran). Ces coordonnées peuvent être des coordonnées utilisateur (x,y) relatives au repère défini par l'utilisateur ou des coordonnées matérielles { #x #y } repérant le pixel dans la matrice de points composant PICT. état vaut 1 si le pixel choisi est allumé et 0 s'il est éteint.

**PIXOFF****Extinction d'un pixel**

Accès (HP-48 G/GX) : [PRG] (PICT) [NXT] (PIXOFF)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] (PIXOFF)

| | | | | |
|----|-------|---|----|--|
| 1: | (x,y) | → | 1: | |
|----|-------|---|----|--|

| | | | | |
|----|-----------|---|----|--|
| 1: | { #x #y } | → | 1: | |
|----|-----------|---|----|--|

PIXOFF prend comme argument les coordonnées d'un pixel (point de l'écran) et éteint ce pixel. Ces coordonnées peuvent être des coordonnées utilisateur (x,y) relatives au repère défini par l'utilisateur ou des coordonnées matérielles { #x #y } repérant le pixel dans la matrice de points composant PICT. Voir aussi PIXON.

PIXON**Allumage d'un pixel**

Accès (HP-48 G/GX) : [PRG] (PICT) [NXT] (PIXON)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] (PIXON)

| | | | | |
|----|-------|---|----|--|
| 1: | (x,y) | → | 1: | |
|----|-------|---|----|--|

| | | | | |
|----|-----------|---|----|--|
| 1: | { #x #y } | → | 1: | |
|----|-----------|---|----|--|

PIXON prend comme argument les coordonnées d'un pixel (point de l'écran) et allume ce pixel. Ces coordonnées peuvent être des coordonnées utilisateur (x,y) ou des coordonnées matérielles { #x #y } repérant le pixel dans PICT. Voir aussi PIXOFF.

PKT**Emission de paquets (Kermit)**

Accès (HP-48 G/GX) : [←] [1] (SRVR) (PKT) **Drapeaux** : -33, -35, -39

Accès (HP-48 S/SX) : [←] [PRG] [NXT] (PKT)

| | | | | |
|----|----------|---|----|-------|
| 2: | "donnée" | → | 2: | |
| 1: | "type" | | 1: | "ans" |

PKT prend deux chaînes en arguments. La première contient les données à émettre par la voie série à l'aide du protocole Kermit, la seconde fixe le type des données. "ans" est une chaîne contenant la réponse reçue à l'issue de l'émission. Cette chaîne peut être vide ou contenir un message d'erreur.

**PMAX****Coordonnées de l'angle supérieur droit**

Accès : [α] [α] [←] ['] [A] [1/x] **Drapeaux** : -

| | | | | |
|----|--------|---|----|--|
| 1: | (x, y) | → | 1: | |
|----|--------|---|----|--|

PMAX affecte les coordonnées (x, y) à l'angle supérieur droit de l'écran graphique.

PMIN**Coordonnées de l'angle inférieur droit**

Accès : [α] [α] [←] ['] [CST] [STO] **Drapeaux** : -

| | | | | |
|----|--------|---|----|--|
| 1: | (x, y) | → | 1: | |
|----|--------|---|----|--|

PMIN affecte les coordonnées (x, y) à l'angle inférieur gauche de l'écran graphique.

POLAR**Activation du mode de tracé polaire**

Accès : [←] [8] (PTYPE) (POLAR) **Drapeaux** : -28, -31

L'exécution de POLAR permet à DRAW de tracer une fonction en utilisant un système de coordonnées polaires (r, θ). Ici θ est la variable indépendante (r est donc la variable dépendante). L'équation représentée graphiquement est stockée dans la variable réservée EQ.

POS

Position d'un fragment de chaîne ou d'un élément d'une liste

Accès (HP-48 G/GX) : [PRG] (LIST) (ELEM) (POS) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] (POS)

| | | |
|---|---|--------------|
| 2: "chaîne_principale" 1: "fragment_de_chaine" | → | 2: 1: pos |
| 2: { liste } 1: élément | → | 2: 1: pos |



POS renvoie pos. pos est un entier naturel correspondant à la position du début d'un fragment de chaîne de caractères au sein d'une chaîne principale ou à la position d'un élément donné dans une liste.

PR1

Impression de l'objet placé au niveau 1 de la pile

Accès (HP-48 G/GX) : [↵] [1] (PR1) Drapeaux : -33, -34, -37, -38

Accès (HP-48 S/SX) : [↵] [MTH] (PR1)

PR1 provoque l'impression de l'objet occupant le niveau 1 de la pile.

PREDV

Valeur de la variable dépendante

Accès : [α] [α] [←] [→] [E] [D] [√x] Drapeaux : -

| | | |
|----------|---|--------|
| 1: indep | → | 1: dep |
|----------|---|--------|

Dans le cadre des calculs de régression, PREDV affiche la valeur de la variable dépendante relative à la valeur de la variable indépendante donnée en argument. L'instruction PREDV est issue du langage des HP-28, et est équivalente à PREDY. Voir PREDY.

PREDX

Valeur de la variable indépendante

Accès (HP-48 G/GX) : [↵] [5] (FIT) (PREDX) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [5] [NXT] [NXT] [NXT] (PREDX)

| | | |
|--------|---|----------|
| 1: dep | → | 1: indep |
|--------|---|----------|

Dans le cadre des calculs de régression, PREDX affiche la valeur →

→ de la variable indépendante relative à la valeur de la variable dépendante donnée en argument.

PREDY

Valeur de la variable dépendante

Accès (HP-48 G/GX) : [←] [5] (FIT) (PREDY) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] (PREDY)

1: → 1:

Dans le cadre des calculs de régression, PREDY affiche la valeur de la variable dépendante relative à la valeur de la variable indépendante donnée en argument.

PRLCD

Impression de l'écran courant

Accès (HP-48 G/GX) : [←] [1] (PRINT) (PRLCD) Drapeaux : -33, -34, -38

Accès (HP-48 S/SX) : [←] [MTH] (PRLCD)

PRLCD imprime le contenu de l'écran courant (pile ou environnement graphique).

PROMPT

Affichage d'une chaîne de caractères

Accès (HP-48 G/GX) : [PRG] [NXT] (IN) [NXT] (PROM) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] (PROM)

1: → 1:

PROMPT affiche une chaîne de caractères donnée en argument dans la partie supérieure de l'écran (zone d'état au-dessus de la pile). [←] [ON] permet la poursuite du programme à l'issue de cet affichage.

PROOT

Racines d'un polynôme calculées à partir de ses coefficients



Accès (HP-48 G/GX) : [←] [7] (POLY) (PROOT) Drapeaux : -22

1: → 1:



→ A partir d'un vecteur contenant les coefficients (rangés par ordre de degrés décroissants) d'un polynôme, **PROOT** renvoie un vecteur contenant les racines dudit polynôme. Par exemple, pour trouver les racines de $P(x) = x^2 - 1$, on donne le vecteur $[1 \ 0 \ -1]$ en argument à **PROOT** qui renvoie $[-1 \ 1]$ signifiant que les racines de $P(x)$ sont $x_1 = -1$ et $x_2 = 1$.

PRST

Impression du contenu de la pile

Accès (HP-48 G/GX) : [↵] [1] (*PRINT*) (*PRST*) **Drapeaux** : -33, -34, -37, -38

Accès (HP-48 S/SX) : [↵] [MTH] (*PRST*)

La pile contenant n objets entre ses niveau 1 et n , **PRST** imprime tous les objets placés dans la pile en commençant par celui qui se trouve au niveau n .

PRSTC

Impression du contenu de la pile

Accès (HP-48 G/GX) : [↵] [1] (*PRINT*) (*PRSTC*) **Drapeaux** : -33, -34, -37, -38

Accès (HP-48 S/SX) : [↵] [MTH] (*PRSTC*)

La pile contenant n objets entre ses niveau 1 et n , **PRSTC** imprime tous les objets placés dans la pile en commençant par celui qui se trouve au niveau n . Les objets occupant plusieurs lignes sont tronqués et réduits à leur première ligne. Voir aussi **PRST**.

PRVAR

Impression d'une variable

Accès (HP-48 G/GX) : [↵] [1] (*PRINT*) (*PRVAR*) **Drapeaux** : -33, -34, -37, -38

Accès (HP-48 S/SX) : [↵] [MTH] (*PRVAR*)

1: 'nom' → 1:

1: { nom_1 nom_2... nom_n } → 1:

PRVAR prend en argument un nom de variable globale ou une liste de noms puis imprime chaque nom de variable suivi de l'objet contenu dans ladite variable.

Avec **PRVAR**, l'impression des objets occupant plusieurs lignes ne provoque pas leur réduction à leur première ligne comme c'est le cas avec **PRSTC**.



PSDEV



Ecart standard de population

Accès (HP-48 G/GX) : [←] [5] (1VAR) [NXT] (PSDEV) Drapeaux : -

| | | |
|----|---|------------------------------|
| 1: | → | 1: e_s_p |
| 1: | → | 1: [e_s_p1 e_s_p2... e_s_pn] |

Si la matrice statistique Σ DAT ne contient qu'une seule colonne, PSDEV renvoie un réel `e_s_p` qui correspond à l'écart standard de population des données statistiques. Si la matrice statistique Σ DAT contient n colonnes, PSDEV renvoie un vecteur `[e_s_p1 e_s_p2... e_s_pn]` dont les éléments sont les écarts standards de population pour chacune des n colonnes de Σ DAT.

PURGE

Suppression de variables ou de répertoires vides

Accès (HP-48 G/GX) : [←] [EEX] Drapeaux : -

Accès (HP-48 S/SX) : [←] [DEL]

| | | |
|-----------------------------|---|----|
| 1: 'nom' | → | 1: |
| 1: { nom_1 nom_2... nom_n } | → | 1: |
| 1: PICT | → | 1: |

PURGE prend en argument un nom de variable globale ou un nom de répertoire et supprime cet objet. Une variable à supprimer doit se trouver dans le répertoire courant. Un répertoire à supprimer doit se trouver dans le répertoire courant et être vide (voir aussi PGDIR). Les noms de variables à supprimer peuvent être réunis sous la forme d'une liste. Si l'argument est PICT, la sauvegarde de l'affichage graphique est supprimée. Cela se traduit par l'affectation à PICT d'un objet graphique de dimensions nulles.

PUT

Remplacement d'un élément dans un tableau ou une liste

Accès (HP-48 G/GX) : [PRG] (LIST) (ELEM) (PUT) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] [NXT] (PUT) →

| | | |
|----------------|---|----------------|
| 3: [[tableau]] | → | 3: |
| 2: pos | | 2: |
| 1: elt | | 1: [[tableau]] |





| | | |
|------------------|---|--------------|
| 3: 'nom_tableau' | → | 3: |
| 2: pos | | 2: |
| 1: elt | | 1: |
| 3: { liste } | → | 3: |
| 2: pos | | 2: |
| 1: elt | | 1: { liste } |
| 3: 'nom_liste' | → | 3: |
| 2: pos | | 2: |
| 1: elt | | 1: |

PUT place un élément `elt` dans une liste ou un tableau (qui peut être une matrice ou un vecteur). `elt` remplace dans une liste ou un tableau l'élément correspondant à la position `pos`. Inséré dans un tableau, `elt` doit être compatible avec le type du tableau modifié, `elt` doit donc être réel ou complexe en fonction du type du tableau. `pos` est le numéro de l'élément à remplacer. `pos` est un entier et correspond au numéro de l'élément d'une liste compté à partir de la gauche ou à l'élément d'un tableau compté à partir de l'élément se trouvant en haut à gauche. Dans le cas d'une matrice, `pos` peut aussi être une liste `{ lign col }` où `lign` et `col` sont les numéros de la ligne et de la colonne où doit avoir lieu l'insertion. Si le tableau ou la liste à modifier sont placés dans la pile, le tableau ou la liste modifié sont renvoyés au niveau 1 après la modification. Si un nom de tableau ou un nom de liste est utilisé comme argument, aucun résultat n'est renvoyé (mais la liste ou le tableau sont effectivement modifiés). Voir aussi `PUTI`.

PUTI

Remplacement d'un élément dans un tableau ou une liste

Accès (HP-48 G/GX) : [PRG] (LIST) (ELEM) (PUTI)

Drapeaux : -64

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] [NXT] (PUTI)

| | | |
|------------------|---|------------------|
| 3: [[tableau]] | → | 3: [[tableau]] |
| 2: pos | | 2: pos+ |
| 1: elt | | 1: pos+ |
| 3: 'nom_tableau' | → | 3: 'nom_tableau' |
| 2: pos | | 2: pos+ |
| 1: elt | | 1: pos+ |
| 3: { liste } | → | 3: { liste } |
| 2: pos | | 2: pos+ |
| 1: elt | | 1: pos+ |





| | | |
|--|---|--|
| <pre>3: 'nom_liste' 2: pos 1: elt</pre> | → | <pre>3: 'nom_liste' 2: pos+ 1: pos+</pre> |
|--|---|--|

PUTI place un élément `elt` dans une liste ou un tableau (qui peut être une matrice ou un vecteur). `elt` remplace dans une liste ou un tableau l'élément correspondant à la position `pos`. Inséré dans un tableau, `elt` doit être compatible avec le type du tableau modifié, `elt` doit donc être réel ou complexe en fonction du type du tableau. `pos` est le numéro de l'élément à remplacer. `pos` est un entier et correspond au numéro de l'élément d'une liste compté à partir de la gauche ou à l'élément d'un tableau compté à partir de l'élément se trouvant en haut à gauche. Dans le cas d'une matrice, `pos` peut aussi être une liste { `lign col` } où `lign` et `col` sont les numéros de la ligne et de la colonne où doit avoir lieu l'insertion. Le tableau ou la liste modifié (ces objets eux-mêmes ou leurs noms, selon le cas), sont renvoyés après l'exécution de PUTI ainsi que `pos+` qui correspond à l'incrément de `pos`. Autrement dit, PUTI renvoie aussi un pointeur sur l'élément qui suit l'élément modifié et prépare ainsi le remplacement de l'élément qui suit celui qui vient d'être modifié. Voir aussi PUT.

PVAR

S/SX

Variance d'une population

Accès (HP-48 G/GX) : [←] [5] (1VAR) [NXT] (PVAR) Drapeaux : -

| | | |
|-------------------------|---|--|
| 1: <code>var_pop</code> | → | 1: [<code>v_p1 v_p2... v_pn</code>] |
|-------------------------|---|--|

Si la matrice statistique Σ DAT ne contient qu'une seule colonne, PVAR renvoie le réel `var_pop` correspondant à la variance de population. Si Σ DAT contient plusieurs colonnes, PVAR renvoie un vecteur [`v_p1 v_p2... v_pn`] dont chaque élément correspond à la variance de population de chaque colonne de Σ DAT.

PVARs

Liste des sauvegardes et des bibliothèques

Accès (HP-48 G/GX) : [←] [2] (PVARs) Drapeaux : -

Accès (HP-48 S/SX) : [←] [VAR] [NXT] (PVARs)

| | | |
|--------------------------|---|---|
| 2: <code>Num_port</code> | → | 2: { <code>:NP:OB ...</code> } |
| 1: <code>mem</code> | | 1: <code>mem</code> |

PVARs prend en argument `Num_port` qui est le numéro d'un port ($0 \leq \text{Num_port} \leq 33$) renvoie une liste { `:NP:OB ...` }. →

→ Cette liste { :NP:OB ... } est la liste des bibliothèques et des sauvegardes dépendant du port spécifié par Num_port. Dans la liste, NP correspond à Num_port et OB au nom de l'objet sauvegarde ou au numéro de l'objet bibliothèque selon le cas.

Si l'argument Num_port vaut 0, PVARs renvoie mem qui correspond alors au nombre d'octets disponibles en RAM interne. mem peut correspondre à différentes données :

- si le port Num_port correspond à la mémoire morte, mem est une chaîne "ROM",
- si le port Num_port correspond à une portion de mémoire vive fusionnée, mem est une chaîne "SYSRAM",
- si le port Num_port correspond à une portion de mémoire vive non fusionnée, mem est le nombre d'octets disponibles dans cette mémoire.

PVIEW

Visualisation de PICT

Accès (HP-48 G/GX) : [PRG] (PICT) [NXT] (PVIEW)

Drapeaux : -

Accès (HP-48 S/SX) : PRG (DSPL) (PVIEW)

| | | |
|--------------|---|--------|
| 1: {} | → | 1: [] |
| 1: { #x #y } | → | 1: [] |
| 1: (x,y) | → | 1: [] |



PVIEW affiche PICT (variable réservée contenant le GROB de l'écran graphique). Si l'argument est une liste vide {}, PVIEW affiche PICT en plaçant l'angle supérieur gauche de PICT sur l'angle supérieur gauche de l'écran. Si l'argument est constitué par des coordonnées matérielles ({ #x #y }) ou utilisateur ((x,y)), PVIEW place le pixel de PICT correspondant à ces coordonnées au niveau de l'angle supérieur gauche de l'écran.

PWRFIT

Modèle d'ajustement de puissance pour les calculs de régression

Accès (HP-48 G/GX) : [←] [5] (ΣPAR) (MODL) (PWRFI)

Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] (MODL) (PWRFI)

PWRFIT impose le modèle d'ajustement de puissance pour les régressions linéaires effectuées par LR.

PX→C

Conversion de coordonnées (matérielles→utilisateur)

Accès (HP-48 G/GX) : [PRG] (PICT) [NXT] (PX→C)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] (PX→C)

1: (#x1 #y1) → 1: (x2, y2)



PX→C convertit des coordonnées matérielles en coordonnées utilisateur (dépendantes du repère défini par l'utilisateur).

→Q

Transformation des nombres décimaux en fractions

Accès (HP-48 G/GX) : [↵] [9] [NXT] (→Q) Drapeaux : -45, -46, -47, -48, -49, -50

Accès (HP-48 S/SX) : [↵] [EVAL]

1: réel → 1: 'expr_symb'

1: complexe → 1: 'expr_symb'

1: 'expr_symb' → 1: 'expr_symb'



Si l'argument est un réel, par exemple 2.56, →Q renvoie une expression symbolique (ici, '64/25') contenant une fraction équivalente au réel donné en argument. De même, si l'argument est un complexe, par exemple (2.3, 3.2) (c'est-à-dire 2,3+3,2.i), →Q renvoie une expression symbolique où les nombres décimaux sont remplacés par des fractions (on obtient ici '(23/10, 16/5)'). Si l'argument est une expression symbolique, →Q renvoie une autre expression symbolique en remplaçant les nombres décimaux par des fractions. Voir aussi →Qπ.

→Qπ

Transformation des nombres décimaux en fractions multiples de π

Accès (: [↵] [9] [NXT] (→Qπ)

Drapeaux : -45, -46, -47, -48, -49, -50

1: réel → 1: 'expr_symb'

1: complexe → 1: 'expr_symb'

1: 'expr_symb' → 1: 'expr_symb'



→Qπ fonctionne comme →Q mais tente d'insérer le facteur π à la suite des fractions. →Qπ factorise donc par π son argument lorsque cela est possible. Voir →Q.

QR**Factorisation d'une matrice**

Accès (HP-48 G/GX) : [MTH] (MATR) (FACTR) (QR)

Drapeaux : -

| | | | | |
|----|-----------------|---|----|---------------|
| 3: | | → | 3: | [[matrice_1]] |
| 2: | | | 2: | [[matrice_2]] |
| 1: | [[matrice_arg]] | | 1: | [[matrice_3]] |

QR renvoie à partir de la matrice [[matrice_arg]] trois matrices [[matrice_1]], [[matrice_2]] et [[matrice_3]] telles que :
 $[[matrice_arg]] \times [[matrice_3]] = [[matrice_1]] \times [[matrice_2]]$.

QUAD**Résolution d'équation quadratique**

Accès : [↵] [9] (QUAD)

Drapeaux : -1

| | | | | |
|----|-------------|---|----|-----------------|
| 2: | 'expr_symb' | → | 2: | |
| 1: | 'var' | | 1: | 'expr_symb_res' |

QUAD résout l'expression algébrique 'expr_symb' pour la variable 'var' et renvoie la solution sous la forme d'une expression symbolique 'expr_symb_res'. Attention ! Les variables intervenant dans 'expr_symb' ne doivent pas exister dans le répertoire courant sous peine d'être ramenées à leurs valeurs numériques en perdant les avantages du calcul symbolique.

QUOTE**Place une expression entre apostrophes**

Accès (HP-48 G/GX) : [↵] [9] [NXT] [NXT] (QUOT)

Drapeaux : -

Accès (HP-48 S/SX) : [↵] [9] [NXT] (QUOT)

QUOTE doit être utilisé pour placer entre apostrophes une expression algébrique utilisée comme argument. Exemple d'utilisation : $fct(arg1, arg2, QUOTE(arg3))$ où fct est une fonction-utilisateur et où $arg1$, $arg2$ et $arg3$ sont des arguments quelconques. QUOTE sert à placer entre apostrophes les arguments symboliques.

RAD**Activation du mode radian**

Accès (HP-48 G/GX) : [↵] [CST] (ANGL) (RAD) Drapeaux : -17, -18

Accès (HP-48 S/SX) : [↵] [CST] [NXT] [NXT] (RAD)



→ RAD active le mode radians. Le radian devient donc l'unité par défaut. Après exécution de RAD la HP-48 considère que le radian est l'unité de mesure d'angle utilisée lorsqu'aucune unité de mesure d'angle n'est précisée.

RAND

Nombre aléatoire

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) (RAND) **Drapeaux : -**

Accès (HP-48 S/SX) : [MTH] (PROB) (RAND)

1: → 1: réel

RAND place au niveau 1 de la pile un nombre aléatoire compris dans l'intervalle $[0,1[$.



RANK

Rang d'une matrice



Accès (HP-48 G/GX) : [MTH] (MATR) (NORM) [NXT] (RANK) **Drapeaux : -54**

1: [[matrice]] → 1: entier

RANK calcule le rang de la matrice rectangulaire donnée en argument.

RANM

Matrice aléatoire



Accès (HP-48 G/GX) : [MTH] (MATR) (MAKE) (RANM) **Drapeaux : -**

1: [[matrice]] → 1: [[matrice]]

1: { 1 c } → 1: [[matrice]]

RANK crée une matrice dont les éléments sont des entiers choisis aléatoirement dans l'intervalle $[-9,9]$. Si l'argument est une matrice, RANM renvoie une matrice de mêmes dimensions dont les éléments sont aléatoires. Si l'argument est une liste $\{ 1 c \}$, RANM renvoie une matrice de dimensions $1 \times c$ dont les éléments sont aléatoires.

RATIO

•••

Division (opérateur en position préfixe)

Accès : $[\alpha]$ $[\alpha]$ $[-\rightarrow]$ [A] [COS] [CST] [EVAL] **Drapeaux : -** →

?

→ **RATIO** est la forme préfixe de / qui lui, est infixe.

Un opérateur infixe (comme +, -, /, etc.) se place entre ses arguments alors qu'un opérateur préfixe se place avant ses arguments. Rappelons que la notation polonaise inversée est une notation postfixe (ou suffixe) imposant la saisie des opérateurs à la suite de leurs arguments.

Dans une expression algébrique, diviser x par 3 peut être obtenu avec 'x/3' ou avec 'RATIO(x, 3)'. Les deux résultats et les deux notations sont absolument équivalents.

R→B

Conversion d'un entier positif en entier binaire

Accès (HP-48 G/GX) : [MTH] (BASE) (R→B) **Drapeaux** : -5, -6, -7, -8, -9, -10, -11, -12

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] (R→B)

| | | | | |
|----|---------------|---|----|-----------------------|
| 1: | entier | → | 1: | entier_binaire |
|----|---------------|---|----|-----------------------|

R→B prend en argument un entier positif (objet de type réel) et renvoie un entier binaire équivalent exprimé à l'aide de la base en cours pour les entiers binaires. R→B ne traite pas les valeurs négatives de l'argument (R→B renvoie alors #0h) et ne traite pas non plus les valeurs supérieures à #FFFFFFFFFFFFFFh.

R→C

Création d'un complexe à partir de deux réels

Accès (HP-48 G/GX) : [PRG] (TYPE) [NXT] (R→C) **Drapeaux** : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] (R→C)

| | | | | |
|----|--------------------------|---|----|-----------------|
| 2: | partie_réelle | → | 2: | complexe |
| 1: | partie_imaginaire | | 1: | |

| | | | | |
|----|------------------|---|----|------------------|
| 2: | [[mat_1]] | → | 2: | [[mat_3]] |
| 1: | [[mat_2]] | | 1: | |

R→C crée un complexe à partir de deux réels (l'un correspond à la partie réelle du complexe, et l'autre, à sa partie imaginaire). R→C peut aussi créer une matrice complexe à partir de deux matrices réelles de mêmes dimensions. La matrice [[mat_1]] contient les parties réelles des complexes contenus dans [[mat_3]] alors que [[mat_2]] contient leurs parties imaginaires. Par exemple, 2 1 R→C crée le complexe (2, 1) ; [[1 2][3 4]] [[5 6][7 8]] R→C crée la matrice complexe [[1,5) (2,6)][(3,7) (4,8)].

R→D

Conversion de radians en degrés

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] [NXT] (R→D)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (VECTR) [NXT] (R→D)

| | | | | |
|----|------|---|----|------|
| 1: | réel | → | 1: | réel |
|----|------|---|----|------|

R→D prend en argument un réel correspondant à une mesure d'angle utilisant le radian comme unité de mesure et renvoie ce réel multiplié par $180 \div \pi$.



RCEQ

Rappel de EQ

Accès (HP-48 G/GX) : [←] [8] [→] (EQ)

Drapeaux : -

Accès (HP-48 S/SX) : [←] [8] [→] (STEQ)

| | | | | |
|----|--|---|----|----|
| 1: | | → | 1: | EQ |
|----|--|---|----|----|

RCEQ place le contenu de la variable réservée EQ (c'est une liste) au niveau 1 de la pile.

RCI

Produit d'une ligne d'un tableau par une constante



Accès (HP-48 G/GX) : [MTH] (MATR) (ROW) (RCI)

Drapeaux : -

| | | | | |
|----|-------------|---|----|-------------|
| 3: | [[matrice]] | → | 3: | |
| 2: | facteur | | 2: | |
| 1: | num_ligne | | 1: | [[matrice]] |

RCI prend une matrice (éventuellement réduite à un vecteur) en argument et multiplie par *facteur* les éléments de la ligne n°*num_ligne* (dans le cas d'un vecteur, l'élément n°*num_ligne* est multiplié par *facteur*). La matrice modifiée est placée au niveau 1 de la pile.

RCIJ

Produit d'une ligne par une constante et somme de lignes



Accès (HP-48 G/GX) : [MTH] (MATR) (ROW) (RCIJ)

Drapeaux : - →

| | | | | |
|----|-------------|---|----|-------------|
| 4: | [[matrice]] | → | 4: | |
| 3: | facteur | | 3: | |
| 2: | num_ligne_2 | | 2: | |
| 1: | num_ligne_1 | | 1: | [[matrice]] |

→ RCIJ multiplie par facteur les éléments de la ligne n°num_lign_2 de la matrice [[matrice]] puis ajoute ces produits aux éléments de la ligne n°num_lign_1 (la ligne n°num_lign_2 n'est pas modifiée dans la matrice finale que renvoie RCIJ). Par exemple, [[1 1][2 2][5 5]] 5 2 3 RCIJ renvoie [[1 1][2 2][15 15]].

RCL

Place un objet dans la pile sans l'évaluer

Accès : [↵] [STO]

Drapeaux : -

1: → 1:



RCL est une instruction très importante dans la mesure où elle permet le rappel de l'objet associé à une variable sans évaluer ledit objet. 'nom' est le nom de l'objet à placer dans la pile. Pour éditer l'objet rappelé sans évaluation, vous pouvez utiliser EDIT avec [↵] [+/-]. RCL peut aussi prendre PICT comme argument et renvoie alors dans la pile le grob associé à PICT. RCL fonctionne aussi avec les bibliothèques et les sauvegardes. Il prend alors comme argument :num_port:num_bib ou :num_port:nom_sauv. num_port est un numéro de port (c'est un entier compris entre 0 et 33). num_bib est un numéro de bibliothèque alors que nom_sauv est un nom de sauvegarde.

RCLALARM

Rappel d'une sonnerie programmée

Accès (HP-48 G/GX) : [↵] [4] (ALRM) (RCLAL) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [4] (ALRM) [NXT] (RCLAL)

1: → 1:

RCLALARM prend en argument un entier qui est le numéro d'une alarme programmée et renvoie une liste contenant la date, l'heure, l'action à réaliser et le délai de répétition de l'alarme spécifiée. Ce délai est exprimé en 1/8192^e de seconde.

RCLF

Etats des 128 indicateurs (flags)

Accès (HP-48 G/GX) : [↵] [CST] (FLAG) [NXT] (RCLF) Drapeaux : -5, -6, -7, -8, -9, -10

Accès (HP-48 S/SX) : [↵] [CST] [NXT] (RCLF) →

1: → 1:

?

→ RCLF renvoie une liste contenant deux entiers binaires exprimés à l'aide de la base courante (voir DEC, HEX, BIN et OCT). Ces entiers binaires sont en fait des mots binaires de 64 bits. Le premier entier binaire de la liste correspond aux états des 64 drapeaux (*flags* ou indicateurs) réservés au système. Le second entier binaire correspond aux états des 64 *flags* destinés à l'utilisateur. Un indicateur est actif (armé) si le bit qui lui correspond vaut 1. De droite à gauche, les 64 bits d'un entier binaire ont pour poids 2^0 à 2^{63} . Le bit de plus faible poids correspond au premier indicateur (*flag* 1 ou -1, selon les conventions de notation). Le bit de plus fort poids (celui de gauche) correspond au drapeau 64 (indicateur ou *flag* 64 ou -64).

RCLKEYS

Liste des touches redéfinies

Accès (HP-48 G/GX) : [←] [CST] (KEYS) (RCLK) Drapeaux : -61, -62

Accès (HP-48 S/SX) : [→] [CST] (RCLK)

1: → 1: { liste }

RCLKEYS renvoie une liste résumant les redéfinitions de touches. Cette liste est composée d'objets suivis de la combinaison de touches provoquant leur évaluation. Les combinaisons de touches sont au format LigCol.Basc. Lig est le numéro de la ligne de touches compté à partir du haut, Col est le numéro de la colonne de touches compté à partir de la gauche. Basc indique l'emploi préalable de touches de bascule :

- 0 ou 1, pas de touche de bascule,
- 2, [←],
- 3, [→],
- 4, [α],
- 5, [α] [←],
- 6, [α] [→].

Par exemple, 21.4 correspond à [α] [MTH]. Notez que la liste renvoyée par RCLKEYS commence par S si les affectations normales des touches non redéfinies restent disponibles malgré la redéfinition d'une partie du clavier.

RCLMENU

Numéro du menu courant

Accès (HP-48 G/GX) : [←] [CST] (MENU) (RCLM) Drapeaux : -

Accès (HP-48 S/SX) : [→] [CST] [NXT] (RCLM) →

1: → 1: menu.page

→ RCLMENU renvoie menu . page qui détermine le menu affiché. menu est le numéro du menu, page est le numéro de la page affichée (on fait défiler les pages avec [NXT]). Par exemple, sur HP-48 G/GX, le menu accessible avec [MTH] [NXT] renvoie 3 . 02.



Attention ! En mode direct (hors d'un programme), il faut taper l'instruction RCLMENU caractère par caractère et non la mettre en œuvre depuis un menu car vous obtiendrez alors toujours 68 . 01 (références du menu où se trouve RCLMENU).

RCLΣ

Place la matrice statistique courante dans la pile

Accès (HP-48 G/GX) : [←] [5] (DATA) (ΣDAT) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [→] (STOΣ)

1: → 1:

RCLΣ place au niveau 1 de la pile la matrice statistique courante (ΣDAT).

RCWS

Place dans la pile la taille du mot binaire courant

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (RCWS) Drapeaux : -5, -6, -7, -8, -9, -10, -11, -12

Accès (HP-48 S/SX) : [MTH] (BASE) (RCWS)

1: → 1:

RCWS place au niveau 1 de la pile un entier qui correspond au nombre de bits composant les mots binaires qui codent les entiers binaires. De 1 à 64 bits peuvent coder les entiers binaires.

R→D

Conversion de radians en degrés

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] [NXT] (R→D) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (VECTR) [NXT] (R→D)

1: → 1:

R→D prend en argument un réel correspondant à une mesure d'angle utilisant le radian comme unité de mesure et renvoie ce réel multiplié par $180 \div \pi$.



RDM**Changement des dimensions d'un tableau**

Accès (HP-48 G/GX) : [MTH] (MATR) (MAKE) (RDM)

Drapeaux : -

Accès (HP-48 S/SX) : [MTH] (MATR) (RDM)

| | |
|----|-------------|
| 2: | [[tableau]] |
| 1: | dim |

→

| | |
|----|-------------|
| 2: | [[tableau]] |
| 1: | [[tableau]] |

| | |
|----|---------------|
| 2: | 'nom_tableau' |
| 1: | dim |

→

| | |
|----|--|
| 2: | |
| 1: | |

RDM change les dimensions d'un tableau (matrice ou vecteur). `dim` peut être de la forme $\{ 1 \ c \}$ où 1 est un nombre de lignes et c un nombre de colonnes, ou de la forme $\{ n \}$ où n est un nombre d'éléments.

Quatre cas sont possibles :

- si `[[tableau]]` est une matrice et si `dim` est de la forme $\{ 1 \ c \}$, RDM renvoie une matrice de dimensions $1 \times c$,
- si `[[tableau]]` est une matrice et si `dim` est de la forme $\{ n \}$, RDM renvoie un vecteur de n éléments,
- si `[[tableau]]` est un vecteur et si `dim` est de la forme $\{ 1 \ c \}$, RDM renvoie une matrice de dimensions $1 \times c$,
- si `[[tableau]]` est un vecteur et si `dim` est de la forme $\{ n \}$, RDM renvoie un vecteur de n éléments.

Par exemple, `[[1 2][3 4][5 6]] {2 3}` RDM renvoie `[[1 2 3][4 5 6]]`. Si le tableau utilisé comme argument est représenté par son nom ('nom_tableau'), il est modifié selon les principes déjà évoqués mais n'est pas renvoyé dans la pile.

**RDZ****Définition du nombre générateur de nombres aléatoires**

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) (TDZ)

Drapeaux : -

Accès (HP-48 S/SX) : [MTH] (PROB) (TDZ)

| | |
|----|------|
| 1: | réel |
|----|------|

→

| | |
|----|--|
| 1: | |
|----|--|

Le nombre réel donné en argument à RDZ est utilisé comme nombre de base pour générer les nombres aléatoires à l'aide de la fonction RAND (voir aussi RAND).



Si l'argument réel est 0, RAND génère le prochain nombre aléatoire en se fondant sur l'horloge système.

RE

Partie réelle

Accès (HP-48 G/GX) : [MTH] [NXT] (CMPL) (RE)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) (RE)

| | | | | |
|----|-------------|---|----|-----------------|
| 1: | complexe | → | 1: | réel |
| 1: | [[tableau]] | → | 1: | [[tableau]] |
| 1: | 'expr_symb' | → | 1: | 'RE(expr_symb)' |
| 1: | réel_unité | → | 1: | réel |

RE extrait la partie réelle. Si l'argument est un complexe (x, y) , RE renvoie la partie réelle x . Si l'argument est un tableau complexe, RE renvoie un tableau dont les éléments sont les parties réelles des éléments appartenant au tableau donné en argument. Si l'argument est un objet-unité (réel_unité), RE renvoie le nombre réel sans son unité.



RECN

Réception de données via Kermit

Accès (HP-48 G/GX) : [←] [1] [NXT] (RECN)

Drapeaux : -33, -35, -36, -39

Accès (HP-48 S/SX) : [←] [PRG] [NXT] (RECN)

| | | | | |
|----|-----|---|----|--|
| 1: | nom | → | 1: | |
|----|-----|---|----|--|

Ici, nom peut être de la forme 'nom' (nom de variable globale entre apostrophes) ou de la forme "nom" (chaîne de caractères entre guillemets). RECN permet comme RECV de placer la HP-48 dans un état la rendant apte à la réception de données. RECN prend comme argument le nom de la variable globale utilisée pour le stockage de l'objet reçu dans le répertoire courant. Voir aussi RECV.

RECT

Activation du mode coordonnées rectangulaires



Accès (HP-48 G/GX) : [MTH] (VECTR) [NXT] (RECT)

Drapeaux : -15, -16

Dans le cadre des représentations graphiques en trois dimensions, le mode RECT (rectangulaires) est à comparer au mode CYLIN (coordonnées cylindriques) ainsi qu'au mode SPHERE (coordonnées sphériques). RECT agit sur les flags 15 et 16.

RECV

Réception de données *via* Kermit

Accès (HP-48 G/GX) : [←] [1] (RECV) Drapeaux : -33, -35, -36, -39

Accès (HP-48 S/SX) : [←] [PRG] (RECV)

RECV de place la HP-48 dans un état la rendant apte à la réception de données. Les objets reçus sont placés dans le répertoire courant sous forme de variable globale portant un nom fixé par l'émetteur, ou, plus simplement, lors de la sauvegarde de l'objet émis.

REPEAT

Structure de boucle While... Repeat... End

Accès (HP-48 G/GX) : [PRG] (BRCH) (WHILE) (REPEAT) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] (REPEAT)

REPEAT fait partie de la structure de boucle conditionnelle WHILE... REPEAT... END.

Structure :

```

WHILE condition à remplir
REPEAT
instructions à exécuter si la condition est remplie
END
    
```

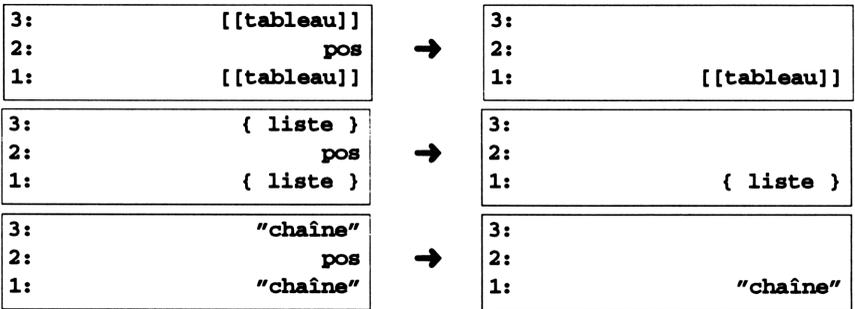


REPL

Remplacement d'une partie d'un objet composite

Accès (HP-48 G/GX) : [PRG] (LIST) (REPL) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] (REPL)





| | | |
|---|---|---|
| 3: objet_graphique 2: pos 1: objet_graphique | → | 3: 2: 1: objet_graphique |
|---|---|---|

→ REPL remplace une portion d'un objet composite (tableau, chaîne de caractères, objet graphique, liste). Dans tous les cas REPL remplace dans l'objet placé au niveau 3 les éléments commençant à la position `pos` par l'objet placé au niveau 1.

Tableaux : `pos` peut être de la forme `{ 1 c }` ou de la forme `n`. `1` est un numéro de ligne dans le tableau, alors que `c` est un numéro de colonne. `n` est le numéro d'un élément compté à partir de l'élément se trouvant en haut à gauche. Si le tableau est un vecteur, `pos` est de la forme `n`. Par exemple, `[[1 1 1][1 1 1][1 1 1]] { 2 2 } [[2 2][3 3]]` REPL renvoie `[[1 1 1][1 2 2][1 3 3]]`. De même, `[[1 1 1][1 1 1][1 1 1]] 5 [[2 2][3 3]]` REPL renvoie la matrice `[[1 1 1][1 2 2][1 3 3]]`

Listes et chaînes : `pos` est un entier `n` qui correspond au numéro de l'élément (listes) ou du caractères (chaînes de caractères) à partir duquel doit être inséré l'objet se trouvant au niveau 1 de la pile. Par exemple, `"BONJOUR" 4 "SOI" REPL` renvoie `"BONSOIR"`. De la même manière `{ B O N J O U R } 4 { S O I } REPL` renvoie `{ B O N S O I R }`.

Objets graphiques (GROB) : `pos` correspond à des coordonnées utilisateur (`x, y`) ou à des coordonnées matérielles `{ #x #y }`. L'objet graphique placé au niveau 1 remplace une portion de l'objet graphique placé au niveau 3 (ce dernier peut être `PICT`). L'angle supérieur gauche du rectangle remplacé a `pos` pour coordonnées. L'objet graphique renvoyé a les mêmes dimensions que l'objet graphique donné en argument au niveau 3, le rectangle remplacé est donc limité par l'objet graphique du niveau 3.

RES

Intervalle entre deux valeurs calculées de la variable indépendante

Accès (HP-48 G/GX) : [←] [8] (PPAR) (RES) Drapeaux : -

Accès (HP-48 S/SX) : [→] [8] [NXT] (RES)

| | | |
|---------------|---|-----------|
| 1: res | → | 1: |
|---------------|---|-----------|

Dans le cadre des tracés, RES détermine à partir de `res` l'intervalle séparant deux valeurs calculées de la variable indépendante. Si `res` est un réel, `res` correspond à l'intervalle entre deux valeurs calculées. Si `res` est un entier binaire, il correspond au nombre de pixels séparant deux points dont les coordonnées sont calculées.

RESTORE**Remplacement de HOME par l'une de ses sauvegardes**Accès (HP-48 G/GX) : [\leftarrow] [VAR] [NXT] (RESTO) Drapeaux : -Accès (HP-48 S/SX) : [\leftarrow] [VAR] [NXT] [NXT] (RESTO)

| | | | |
|----|-------------|---|----|
| 1: | sauv | → | 1: |
|----|-------------|---|----|

RESTORE remplace HOME (et donc, toute la RAM utilisateur interne) par **sauv** qui est une sauvegarde de HOME préalablement effectuée avec ARCHIVE. **sauv** peut être un objet se trouvant sur un port, **sauv** est alors de la forme : num_port : nom où num_port est compris entre 0 et 33. Notez que les ports externes ne doivent pas, avec RESTORE, être fusionnés avec la RAM interne. **sauv** peut aussi, tout simplement être le nom de la variable dans laquelle HOME a été enregistré.

**REVLIST****Inversion de l'ordre des éléments d'une liste**

Accès (HP-48 G/GX) : [MTH] (LIST) (REVL)

Drapeaux : -

| | | | | |
|----|-----------|---|----|-----------|
| 1: | { liste } | → | 1: | { liste } |
|----|-----------|---|----|-----------|

REVLIST «retourne» une liste, ainsi { 1 2 3 } REVLIST renvoie la liste { 3 2 1 }.

RKF**Résolution d'équation différentielle**Accès (HP-48 G/GX) : [\leftarrow] [7] (DIFFE) (RKF)

Drapeaux : -

| | | | | |
|----|------------------|---|----|------------------|
| 3: | { t y mb_droit } | → | 3: | { t y mb_droit } |
| 2: | err_abs | | 2: | err_abs |
| 1: | finale | | 1: | err_abs |

RKF résout une équation différentielle (méthode Runge-Kutta-Fehlberg). **t** est le nom de la variable globale correspondant à la variable indépendante. **y** est le nom de la variable globale correspondant à la variable solution. **mb_droit** est une expression algébrique contenant le membre droit de l'équation différentielle (ou le nom d'une variable globale contenant cette expression symbolique). **err_abs** est la tolérance d'erreur absolue (c'est un réel, par exemple 10^{-5}). **finale** est la valeur finale de la variable indépendante **T**. Avant d'utiliser RKF, il convient de placer dans **T** la valeur initiale de la variable indépendante, et dans **Y**, la valeur initiale de la variable dépendante. Si **mb_droit** est un nom de variable, celle-ci doit contenir l'expression algébrique définissant le membre droit de l'équation. →

→ Après activation de RKF, les résultats sont renvoyés dans les variables globales T (variable indépendante) et Y (variable dépendante).

RKFERR

Erreur liée à RKF



Accès (HP-48 G/GX) : [←] [7] (DIFFE) (RKFE)

Drapeaux : -

| | |
|----|------------------|
| 4: | |
| 3: | |
| 2: | { t y mb_droit } |
| 1: | inter |



| | |
|----|------------------|
| 4: | { t y mb_droit } |
| 3: | inter |
| 2: | inter_sol |
| 1: | err_abs |



RKFERR fournit une estimation de l'erreur absolue liée à l'emploi de la méthode Runge-Kutta-Fehlberg (voir RKF). *t* est le nom de la variable globale correspondant à la variable indépendante. *y* est le nom de la variable globale correspondant à la variable solution. *mb_droit* est le nom de la variable globale contenant le membre droit de l'équation différentielle. *inter* est l'intervalle entre deux valeurs calculées (pas de calcul). *inter_sol* est le changement de solution lié à *inter*. *err_abs* est l'erreur absolue liée à *inter*.

RKFSTEP

Intervalle utilisé pour le prochain calcul



Accès (HP-48 G/GX) : [←] [7] (DIFFE) (RKFS)

Drapeaux : -

| | |
|----|------------------|
| 3: | { t y mb_droit } |
| 2: | err_abs |
| 1: | inter |



| | |
|----|------------------|
| 3: | { t y mb_droit } |
| 2: | err_abs |
| 1: | inter_suisvant |

Voir avant tout RKF et RKFERR. RKFSTEP renvoie l'intervalle suivant entre deux valeurs calculées dans le cadre de la résolution d'une équation différentielle.

RL

Rotation d'un bit vers la gauche

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BIT) (RL)

Drapeaux : -5 à -12

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] (RL)

| | |
|----|----------------|
| 1: | entier_binaire |
|----|----------------|



| | |
|----|----------------|
| 1: | entier_binaire |
|----|----------------|

RL décale tous les bits (jusqu'à 64) d'un mot binaire vers la gauche et fait passer le bit de plus fort poids (bit de gauche) à droite du mot binaire.

RLB**Rotation d'un octet vers la gauche**Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BYTE) (RLB) **Drapeaux** : -5 à -12

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] (RLB)

| | |
|----|----------------|
| 2: | entier_binaire |
| 1: | |

→

| | |
|----|----------------|
| 2: | entier_binaire |
| 1: | |



RLB décale tous les octets (jusqu'à 8) d'un mot binaire vers la gauche et fait passer l'octet de plus fort poids (octet de gauche) à droite du mot binaire. Voir aussi RL. Par exemple, #FFF0000000000000h devient, après exécution de RLB, #F0000000000000FFh.

RND**Ajustement du nombre de décimales ou de chiffres significatifs**Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] [NXT] (RND) **Drapeaux** : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] [NXT] (RND)

| | | |
|--------------------------|---|--------------------------------|
| 2: réel 1: arr | → | 2: réel 1: |
| 2: objet_unité 1: arr | → | 2: objet_unité 1: |
| 2: complexe 1: arr | → | 2: complexe 1: |
| 2: 'expr_symb' 1: arr | → | 2: 'RND(expr_symb, arr)' 1: |
| 2: [[tableau]] 1: arr | → | 2: [[tableau]] 1: |

ARR étant un entier compris entre -11 et 12 inclus. RND modifie en fonction de ARR le nombre de décimales ou le nombre de chiffres significatifs de l'argument placé au niveau 2. Le mode d'affichage déterminé par ARR devient le mode d'affichage courant. ARR peut prendre trois types de valeurs :

- si $ARR=12$, le mode d'affichage courant est appliqué à l'argument placé au niveau 2,
- si $0 \leq ARR \leq 11$, le nombre de décimales est fixé à la valeur de ARR,
- si $-11 \leq ARR \leq -1$, le nombre de chiffres significatifs est fixé à la valeur de -ARR.

RND prend comme argument des nombres réels (réel, partie réelle d'un objet-unité, complexes se résumant à un couple de réels, matrices ou vecteurs réels ou complexes, expression symbolique, etc.).

RNRM

Norme de ligne d'un tableau

Accès (HP-48 G/GX) : [MTH] (MATR) (NORM) (RNRM) **Drapeaux : -**

Accès (HP-48 S/SX) : [MTH] (MATR) [NXT] (RNRM)

| | | |
|-----------------------------------|---|----------------------------|
| 1: [[tableau]] | → | 1: réel |
|-----------------------------------|---|----------------------------|

RNRM renvoie un réel correspondant à la norme de ligne du tableau donné en argument.

ROLL

Déplace un objet de la pile vers le niveau 1

Accès (HP-48 G/GX) : [←] [↑] (ROLL) **Drapeaux : -**

Accès (HP-48 S/SX) : [PRG] (STK) (ROLL)

| | | |
|---------------------------------|---|---------------------------------|
| n+1: objet_n | → | n+1: objet_n-1 |
| n: objet_n-1 | | n: objet_n-1 |
| ... | | ... |
| 3: objet_2 | | 3: objet_2 |
| 2: objet_1 | | 2: objet_1 |
| 1: n | | 1: objet_n |

ROLL prend en argument un entier n et déplace le contenu du niveau $n+1$ de la pile vers le niveau 1, le reste de la pile étant refoulé.



ROLLD

Déplace un objet de la pile vers un niveau n

Accès (HP-48 G/GX) : [←] [↑] (ROLLD) **Drapeaux : -**

Accès (HP-48 S/SX) : [PRG] (STK) (ROLLD)

| | | |
|---------------------------------|---|-------------------------------|
| n+1: objet_n | → | n: objet_1 |
| n: objet_n-1 | | n-1: objet_n |
| ... | | n-2: objet_n-1 |
| 3: objet_2 | | ... |
| 2: objet_1 | | 2: objet_3 |
| 1: n | | 1: objet_2 |

ROLLD (*roll down*) prend en argument un entier n et déplace le contenu du niveau 2 de la pile vers le niveau n , le reste de la pile descend d'un niveau vers le niveau 1.



Voir aussi ROLL.

ROOT

Calcul de racine

Accès (HP-48 G/GX) : [\leftarrow] [7] (ROOT) (ROOT) Drapeaux : -

Accès (HP-48 S/SX) : [\leftarrow] [7] (ROOT)

| | |
|----|-------------|
| 3: | 'expr_symb' |
| 2: | 'x' |
| 1: | approx |



| | |
|----|--------|
| 3: | |
| 2: | |
| 1: | racine |



ROOT renvoie la racine d'une expression symbolique. 'expr_symb' est une expression symbolique à partir de laquelle sera résolue l'équation $expr_symb=0$ pour la variable 'x'. approx est une approximation de la solution recherchée. racine, la valeur de 'x' pour laquelle 'expr_symb' s'annule, est renvoyée au niveau 1. Par exemple, 'X^2-1' 'X' 1.5 ROOT résout $x^2-1=0$ et renvoie la valeur de x correspondant à la solution la plus proche de l'approximation fournie (ici, 1,5).

ROT

Fait passer le contenu du niveau 3 au niveau 1

Accès (HP-48 G/GX) : [\leftarrow] [\uparrow] (ROT) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (STK) (ROT)

| | |
|----|---------|
| 4: | objet_4 |
| 3: | objet_3 |
| 2: | objet_2 |
| 1: | objet_1 |



| | |
|----|---------|
| 4: | objet_4 |
| 3: | objet_2 |
| 2: | objet_1 |
| 1: | objet_3 |

ROT fait passer au niveau 1 de la pile le contenu du niveau 3. Voir aussi SWAP, ROLL et ROLLD.

→ROW

Décomposition d'une matrice en vecteurs



Accès (HP-48 G/GX) : [MTH] (MATR) (ROW) (→ROW) Drapeaux : -

| | |
|-----|-------------|
| n: | |
| ... | |
| 2: | |
| 1: | [[matrice]] |



| | |
|------|-----------|
| n+1: | [ligne_n] |
| ... | |
| 2: | [ligne_1] |
| 1: | n |



| | |
|-----|-----------|
| n: | |
| ... | |
| 2: | |
| 1: | [vecteur] |

| | |
|------|-----------|
| n+1: | élément_n |
| ... | |
| 2: | élément_1 |
| 1: | n |



→ →ROW décompose une matrice en vecteurs ou décompose un vecteur en ses éléments. Le nombre de lignes de la matrice (ou le nombre d'éléments du vecteur) est renvoyé à la suite des vecteurs (ou à la suite des éléments dans le cas d'un vecteur). Par exemple, $[[1\ 2][3\ 4]] \rightarrow \text{ROW}$ renvoie $[1\ 2]$, $[3\ 4]$ et 2 aux niveaux 1 à 3 de la pile.

ROW+



Insertion d'un tableau dans un autre tableau

Accès (HP-48 G/GX) : [MTH] (MATR) (ROW) (ROW+)

Drapeaux : -

3: [[tableau]]
2: [[tableau]]
1: pos



3:
2:
1: [[tableau]]

3: [[tableau]]
2: [vecteur]
1: pos



3:
2:
1: [[tableau]]

3: [vecteur]
2: elt
1: pos



3:
2:
1: [vecteur]

ROW+ insère l'élément placé au niveau 2 dans le tableau placé au niveau 3. L'insertion crée une ou plusieurs nouvelles lignes dans le tableau, pos étant le numéro de la première ligne insérée. Par exemple, $[[1\ 2\ 3][4\ 5\ 6][7\ 8\ 9]]\ [[15\ 16\ 17][18\ 19\ 20]]\ 2\ \text{ROW+}$ renvoie la matrice :

```

[[1 2 3]
 [15 16 17]
 [18 19 20]
 [4 5 6]
 [7 8 9]].
    
```

De même, un vecteur représentant une ligne peut être inséré dans une matrice, ainsi $[[1\ 2\ 3][4\ 5\ 6][7\ 8\ 9]]\ [15\ 16\ 17]\ 4\ \text{ROW+}$ renvoie $[[1\ 2\ 3][4\ 5\ 6][7\ 8\ 9][15\ 16\ 17]]$. Dans le cas d'un vecteur, ROW+ se contente d'insérer un élément elt, pos représentant alors le numéro de l'élément inséré. Ainsi, $[1\ 2\ 3]\ 10\ 1\ \text{ROW+}$ renvoie $[10\ 1\ 2\ 3]$.



Attention ! L'insertion d'une matrice dans une autre matrice (ou d'un vecteur dans une matrice) n'est possible que si les deux matrices comportent le même nombre de colonnes (ou si le vecteur inséré comporte autant d'éléments qu'il y a de colonnes dans la matrice cible).

ROW-



Suppression d'une ligne dans une matrice

Accès (HP-48 G/GX) : [MTH] (MATR) (ROW) (ROW-)

Drapeaux : -

2: [[matrice]]
1: num_ligne



2: [[matrice]]
1: [vecteur]

2: [vecteur]
1: num_elt



2: [vecteur]
1: elt

Dans une matrice, ROW- supprime la ligne de numéro num_ligne et renvoie la ligne supprimée sous la forme [vecteur] et la nouvelle matrice. Dans le cas d'un vecteur, ROW- supprime l'élément de numéro num_elt et renvoie cet élément elt ainsi que le nouveau vecteur. Par exemple, [[1 2][3 4][5 6]] 2 ROW- renvoie [[1 2][5 6]] et [3 4]. Dans le cas d'un vecteur, par exemple [5 6 7] 2 ROW- renvoie [5 7] et 6.

ROW→



Création d'une matrice à partir de ses lignes

Accès (HP-48 G/GX) : [MTH] (MATR) (ROW) (ROW→)

Drapeaux : -

n+1: [ligne_1]
...
3: [ligne_n-1]
2: [ligne_n]
1: n



n+1:
...
3:
2:
1: [[tableau]]

n+1: élément_1
...
3: élément_n-1
2: élément_n
1: n



n+1:
...
3:
2:
1: [vecteur]



ROW→ construit une matrice de n lignes (ou un vecteur de n éléments) à partir de vecteurs représentant chaque ligne (ou à partir des n éléments devant constituer le vecteur). Par exemple, [1 2] [3 4] [5 6] 3 ROW→ renvoie [[1 2][3 4][5 6]]. De même 1 2 3 3 ROW→ renvoie [1 2 3].

RR

Rotation d'un bit vers la droite

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BIT) (RR)

Drapeaux : -12 à -5

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] (RR)



1: entier_binaire → 1: entier_binaire
 → RR décale tous les bits (jusqu'à 64) d'un mot binaire vers la droite et fait passer le bit de plus faible poids (bit de droite) à gauche du mot binaire.

RRB

Rotation d'un octet vers la droite

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BYTE) (RRB) Drapeaux : -12 à -5

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] (RRB)

1: entier_binaire → 1: entier_binaire

RRB décale tous les octets (jusqu'à 8) d'un mot binaire vers la droite et fait passer l'octet de plus faible poids (octet de droite) à gauche du mot binaire. Voir aussi RR.



RREF

Forme réduite échelonnée



Accès (HP-48 G/GX) : [MTH] (MATR) (FACTR) (RREF) Drapeaux : -54

1: [[matrice]] → 1: [[matrice]]

RREF renvoie la forme réduite échelonnée de la matrice donnée en argument.

RRK

Résolution d'équation différentielle



Accès (HP-48 G/GX) : [←] [7] (DIFFE) (RRK) Drapeaux : -

| | | |
|---------------------------|---|---------------------------|
| 3: { T Y mb_droit dY dT } | → | 3: |
| 2: Tol | | 2: { T Y mb_droit dY dT } |
| 1: Finale | | 1: Tol |

RRK résout une équation différentielle (valeurs initiales et dérivées partielles connues). La méthode utilisée est la méthode Runge-Kutta-Rosenbrock. Au niveau 3 doit se trouver une liste contenant : T, Y, mb_droit, dY et dT. T est une variable globale contenant la valeur initiale de la variable indépendante. Y est une variable globale contenant la valeur initiale de la variable solution. mb_droit est une variable contenant l'expression algébrique déterminant le membre droit de l'équation différentielle à résoudre. dY est une expression algébrique contenant l'expression algébrique représentant la dérivée partielle $\partial f / \partial y$. De même, dT contient $\partial f / \partial t$. →

→ Tol est la tolérance (par exemple, 0,000001) acceptée pour le résultat. Finale est la dernière valeur calculée pour la variable indépendante. Après activation de RRK, les résultats sont placés dans les variables Y et T qu'il convient de consulter.

RRKSTEP



Recherche du prochain intervalle de calcul

Accès (HP-48 G/GX) : [←] [7] (DIFFE) (RRKS) Drapeaux : -

```
4: { T Y mb_droit dY dT }
3:                               Tol
2:                               inter_init
1:                               meth
```



```
4: { T Y mb_droit dY dT }
3:                               Tol
2:                               inter_suiv
1:                               meth_courante
```

RRKSTEP calcule `inter_suiv` qui est le prochain intervalle séparant deux valeurs calculées au cours de l'utilisation des fonctions RRK ou RKF (voir RRK ou RKF pour les définitions des éléments de la liste au niveau 4 et de Tol). `inter_init` est l'intervalle initial entre deux valeurs calculées. On notera que, si Tol l'exige, `inter_init` peut être ajusté par la HP-48. `meth` indique la méthode de calcul courante (si vous utilisez RRKSTEP pour la première fois, `meth` doit valoir 0, si la méthode courante est RKF, `meth` vaut 1, si la méthode courante est RRK, `meth` vaut 2). RRKSTEP renvoie `meth_courante`, autrement dit la méthode en cours d'utilisation sous la forme d'un entier compris entre 0 et 2 inclus à interpréter comme `meth`.

RSBERR



Erreur liée à l'emploi de la méthode de Rosenbrock

Accès (HP-48 G/GX) : [←] [7] (DIFFE) (RRKS) Drapeaux : -

```
4:
3:
2: { T Y mb_droit dY dT }
1:                               inter_init
```



```
4: { T Y mb_droit dY dT }
3:                               inter_init
2:                               chgt_sol
1:                               erreur
```

RSBERR calcule `chgt_sol` et `erreur` qui sont respectivement le changement de solution et l'erreur absolue associées à l'emploi de la méthode de Rosenbrock avec `inter_init`.

Voir impérativement RRK pour connaître la signification du contenu de la liste.



Attention ! `erreur` peut valoir 0 ce qui signifie que la méthode de Rosenbrock n'a pas pu être appliquée aux arguments.

RSD

Calcul de résidu

Accès (HP-48 G/GX) : [MTH] (MATR) [NXT] (RSD)

Drapeaux : -

Accès (HP-48 S/SX) : [MTH] (MATR) (RSD)

| | |
|----|---------------|
| 3: | [[tableau_B]] |
| 2: | [[matrice_A]] |
| 1: | [[tableau_Z]] |



| | |
|----|-------------------|
| 3: | |
| 2: | |
| 1: | [[tableau_B-A.Z]] |

Rappelons qu'un tableau peut être un vecteur ou une matrice. A partir des tableaux B, A et Z, RSD calcule B-A.Z. Si B est une matrice, Z doit l'être aussi et comporter autant de colonnes que B. Si Z est un vecteur, son nombre d'éléments doit correspondre au nombre de colonnes de A. Si Z est une matrice, le nombre de lignes de Z doit correspondre au nombre de colonnes de A. Si B est un vecteur, il doit comporter un nombre d'éléments correspondant au nombre de lignes de A. Si B est une matrice, cette matrice doit comporter un nombre de lignes égal au nombre de lignes de A.



RSWP

Permutation de deux lignes d'une matrice

Accès (HP-48 G/GX) : [MTH] (MATR) (ROW) [NXT] (RSWP)

Drapeaux : -

| | |
|----|-------------|
| 3: | [[tableau]] |
| 2: | ligne_1 |
| 1: | ligne_2 |



| | |
|----|-------------|
| 3: | |
| 2: | |
| 1: | [[tableau]] |

RSWP (*row swap*) permute les lignes de numéros ligne_1 et ligne_2 d'un tableau (matrice ou vecteur). Par exemple, [[1 2][3 4][5 6][7 8]] 2 3 RSWP renvoie [[1 2][5 6][3 4][7 8]]. RSWP peut aussi être appliqué à un vecteur. Dans ce cas, ligne_1 et ligne_2 représentent des numéros d'éléments. Par exemple, [7 8 9 10] 2 3 RSWP renvoie [7 9 8 10].



SAME

Deux objets sont-ils identiques ?

Accès (HP-48 G/GX) : [PRG] (TEST) [NXT] (SAME)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (TEST) (SAME)

| | |
|----|---------|
| 2: | objet_a |
| 1: | objet_b |



| | |
|----|--------|
| 2: | |
| 1: | entier |

SAME compare objet_a et objet_b puis renvoie 1 si les objets sont identiques ou 0 sinon.



SBRK

Interruption d'une communication par la voie série

Accès (HP-48 G/GX) : [←] [1] [NXT] (SERIA) (SBRK) Drapeaux : -33

Accès (HP-48 S/SX) : [←] [PRG] [NXT] [NXT] (SBRK)

SBRK met fin à la communication en cours par la voie série (port infra-rouge ou câble).

SCALE

Intervalle entre les graduations sur les axes

Accès (HP-48 G/GX) : [←] [8] (PPAR) [NXT] (SCALE) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [8] [NXT] (SCALE)

| | | | |
|----|--------|---|----|
| 2: | grad_x | → | 2: |
| 1: | grad_y | | 1: |

Dans le cadre des représentations graphiques, grad_x et grad_y sont respectivement, le nombre d'unités-utilisateur séparant chaque graduation de l'axe des abscisses et le nombre d'unités-utilisateur séparant chaque graduation de l'axe des ordonnées. SCALE modifie PPAR en fonction de grad_x et de grad_y.

SCATRLOT

Nuage de points

Accès (HP-48 G/GX) : [←] [5] (PLOT) (SCATR) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] (SCATR)

Le mode graphique SCATTER étant actif (voir SCATTER), SCATRLOT trace un nuage de points représentant les données de la matrice statistique Σ DAT. Les abscisses des points tracés sont stockées dans la matrice de Σ DAT spécifiée par XCOL. De même, les ordonnées sont définies par YCOL.

**SCATTER**

Activation du mode « nuage de points »

Accès (HP-48 G/GX) : [←] [8] [NXT] (STAT) (PTYPE) (SCATT) Drapeaux : -

Accès (HP-48 S/SX) : [←] [8] (PTYPE) [NXT] (SCATT)

SCATTER active le type de tracé permettant le tracé d'un nuage de points (voir aussi SCATRLOT).



SCHUR



Décomposition de Schur

Accès (HP-48 G/GX) : [MTH] (MATR) (FACTR) (SCHUR) Drapeaux : -

| | | | | |
|----|-----------------|---|----|---------------|
| 2: | | → | 2: | [[matrice_Q]] |
| 1: | [[matrice_arg]] | | 1: | [[matrice_T]] |

SCHUR prend une matrice [[matrice_arg]] en argument et renvoie deux matrices [[matrice_Q]] et [[matrice_T]] vérifiant l'égalité :
 $[[matrice_arg]] = [[matrice_Q]] \times [[matrice_T]] \times [[matrice_Q]]^T$ où
 $[[matrice_Q]]^T$ est la transposée (conjuguée) de [[matrice_Q]].
 Rappelons que TRN transpose une matrice.

SCI

Affichage dit scientifique

Accès (HP-48 G/GX) : [←] [CST] (FMT) (SCI) Drapeaux : -

Accès (HP-48 S/SX) : [←] [CST] (SCI)

| | | | | |
|----|---|---|----|--|
| 1: | n | → | 1: | |
|----|---|---|----|--|

SCI active un mode d'affichage imposant l'affichage des réels avec une partie entière limitée à un chiffre et une partie décimale comportant n chiffres, le tout est suivi d'un facteur dix élevé à une puissance signalée à la suite d'un «E» chargé de séparer la partie décimale de cet exposant. Ainsi, 1,05E-2 représente 0,0105.



SCLΣ



Ajustement des paramètres graphiques pour SCATRPLOT

Accès : [α] [α] [SIN] [C] [NXT] [α] [↗] [TAN] [←] [←] [⇐] [→] Drapeaux : -

SCLΣ ajuste les intervalles de représentation afin de les adapter à la représentation d'un nuage de points fondés sur la matrice statistique ΣDAT (voir SCATTER et SCATRPLOT). SCLΣ ne présente pas d'intérêt puisque SCATRPLOT l'exécute automatiquement.

SCONJ

Remplace le contenu d'une variable par son conjugué

Accès (HP-48 G/GX) : [←] [VAR] (ARITH) [NXT] (SCON) Drapeaux : -

Accès (HP-48 S/SX) : [↗] [VAR] [NXT] (SCON) →

| | | | | |
|----|-------|---|----|--|
| 1: | 'nom' | → | 1: | |
|----|-------|---|----|--|

→ **SCONJ** prend un nom de variable globale en argument, applique **CONJ** à son contenu puis stocke le résultat de cette application dans ladite variable. Voir **CONJ**.

SDEV

Ecart standard sur échantillon

Accès (HP-48 G/GX) : [←] [5] (1 VAR) (SDEV) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] (SDEV)

1: → 1: réel

1: → 1: [vecteur]

SDEV calcule l'écart standard sur échantillon à partir de la matrice Σ_{DAT} . Si Σ_{DAT} ne comporte qu'une seule colonne, SDEV ne renvoie qu'un réel. Si Σ_{DAT} comporte plusieurs colonnes, SDEV renvoie un vecteur comportant autant d'éléments qu'il y a de colonnes dans Σ_{DAT} .

SEND

Emission d'une variable

Accès (HP-48 G/GX) : [←] [1] (SEND) Drapeaux : -33, -35, -39

Accès (HP-48 S/SX) : [←] [PRG] (SEND)

1: nom → 1:

SEND envoie la variable de nom *nom* par la voie série en utilisant le protocole Kermit. *nom* peut aussi être une liste de noms entre accolades, par exemple, { *nom_1 nom_2 nom_3* }. *nom* doit dans tous les cas être relatif à une ou plusieurs variables du répertoire courant.

SEQ

Calcul séquentiel



Accès (HP-48 G/GX) : [PRG] (LIST) (PROC) [NXT] (SEQ) Drapeaux : -

| | | |
|---|---|---|
| 5: à_exécuter 4: var 3: valeur_début 2: valeur_fin 1: intervalle_entre_val | → | 5: 4: 3: 2: 1: { liste résultats } |
|---|---|---|

SEQ exécute séquentiellement *à_exécuter* (qui peut être un programme, une expression algébrique, etc.) en faisant varier l'élément variable *var* →



→ entre valeur_début et valeur_fin, chaque valeur calculée étant séparée de la suivante par intervalle_entre_val. Les résultats sont renvoyés sous forme d'une liste. Par exemple, 'A+2' 'A' 1 5 2 SEQ renvoie la liste { 3 3.5 4 4.5 5 5.5 6 6.5 7 }.

SERVER

Mode serveur dans le cadre d'une communication Kermit

Accès (HP-48 G/GX) : [←] [1] (SRVR) (SERVE) Drapeaux : -33, -35, -36, -39

Accès (HP-48 S/SX) : [←] [PRG] (SERV)

SERVER place la HP-48 en mode serveur et la rend donc interrogeable par une autre HP-48 ou par un micro-ordinateur dans le cadre d'une communication série faisant appel au protocole Kermit.

SF

Armement d'un drapeau

Accès (HP-48 G/GX) : [PRG] (TEST) [NXT] [NXT] (SF) Drapeaux : -

1: → 1:

Entier est non nul, compris dans l'intervalle [-64,64] et correspond à un drapeau (aussi appelé flag ou indicateur). Le drapeau correspondant à entier est armé (activé) par SF (set flag).



SHOW

Evaluation de certaines variables d'une expression algébrique

Accès : [←] [9] (SHOW) Drapeaux : -3

2: → 2:
 1: → 1:

Dans l'expression symbolique 'expr_symb' donnée en argument, SHOW remplace toutes les variables globales par leurs valeurs. Seules les valeurs dont les noms figurent dans {liste} ne sont pas remplacées par leurs valeurs.

Par exemple, si C existe dans le répertoire courant et correspond à la valeur 5,

'A+B+C' { A B } SHOW

renvoie 'A+B+5' si C existe dans le répertoire courant et correspond à la valeur 5.

SIDENS**Densité intrinsèque du silicium**

Accès (HP-48 G/GX) : [←] [3] (UTILS) (SIDEN) Drapeaux : -3

1: temp → 1: densité

SIDENS renvoie densité la densité intrinsèque du silicium (Si) en fonction de la température temp. densité et temp peuvent être des réels, des objets-unités ou des expressions symboliques. temp est exprimée en degrés Kelvin (si une autre unité est utilisée, temp doit alors être un objet-unité).

SIGN**Signe**

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] (SIGN) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) (SIGN)

1: complexe → 1: signe

1: réel → 1: signe



Si l'argument est un réel (ou un objet-unité), signe vaut 0 si l'argument vaut 0, -1 si l'argument est négatif ou 1 si l'argument est positif. Si l'argument est un complexe, signe correspond au nombre complexe divisé par son module.

SIMU**Tracés simultanés**

Accès (HP-48 G/GX) : [←] [8] [NXT] (FLAG) (SIMU) Drapeaux : -28

Les représentations graphiques peuvent être tracées les unes après les autres ou simultanément. SIMU active le tracé simultané des différentes représentations graphiques.

SIN**Fonction sinus**

Accès : [SIN] Drapeaux : -3, -17, -18

1: réel → 1: réel



SIN calcule le sinus du réel (ou de l'objet-unité) donné en argument. SIN tient compte de l'unité de mesure d'angle courante.

SINH**Sinus hyperbolique**

Accès : [MTH] (HYP) (SINH)

Drapeaux : -3

| | | | | |
|----|-------------|---|----|-------------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'SINH(expr_symb)' |

SINH calcule le sinus hyperbolique de son argument ou renvoie une nouvelle expression symbolique si l'argument est une expression symbolique.

SINV**Remplace un objet par son inverse**

Accès (HP-48 G/GX) : [←] [VAR] (ARITH) [NXT] (SINV) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [VAR] [NXT] (SINV)

| | | | | |
|----|-------|---|----|--|
| 1: | 'var' | → | 1: | |
|----|-------|---|----|--|

SINV applique INV à l'objet stocké dans la variable nommée 'var' puis stocke le résultat dans ladite variable.

SIZE**Taille d'un objet composite**

Accès (HP-48 G/GX) : [PRG] (LIST) (ELEM) (SIZE)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] (SIZE)

| | | | | |
|----|-----------------|---|----|---------|
| 1: | { liste } | → | 1: | taille |
| 1: | [[tableau]] | → | 1: | taille |
| 1: | "chaîne" | → | 1: | taille |
| 1: | 'expr_symb' | → | 1: | taille |
| 1: | objet_unité | → | 1: | taille |
| 2: | objet_graphique | → | 2: | largeur |
| 1: | | | 1: | hauteur |

SIZE renvoie la taille d'un objet composite. taille est le nombre de caractères d'une chaîne de caractères, le nombre d'éléments d'une liste ou une liste correspondant aux dimensions d'un tableau (matrice ou vecteur). Si l'argument est un objet graphique, SIZE renvoie →

→ deux entiers binaires. L'un correspond à la largeur, l'autre à la hauteur. Ils correspondent tous les deux à des dimensions exprimées en pixels. Si l'argument est un objet-unité, taille correspond à la somme des tailles des champs (chaque champ correspond à 1 ou à 2). Pour un objet quelconque, SIZE renvoie 1.

SL

Décalage d'un bit vers la gauche

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BIT) (SL) **Drapeaux** : -12 à -5

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] [NXT] (SL)

1: → 1:

Quelle que soit sa base de représentation (binaire, octale, décimale ou hexadécimale), SL considère la forme binaire d'un entier binaire et décale tous ses bits d'un rang vers la gauche (le bit de poids le plus fort est perdu). SL peut être utilisé pour une multiplication par deux. Par exemple, #5h SL renvoie #Ah. Ne pas confondre avec RL.



SLB

Décalage d'un octet vers la gauche

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BYTE) (SLB) **Drapeaux** : -12 à -5

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] [NXT] (SLB)

1: → 1:

Quelle que soit sa base de représentation (binaire, octale, décimale ou hexadécimale), SLB considère la forme binaire d'un entier binaire et décale tous ses bits de huit rangs vers la gauche (l'octet de poids le plus fort est perdu). SLB peut être utilisé pour une multiplication par 256. Par exemple, #1h SLB renvoie #100h.



SLOPEFIELD

Mode graphique SLOPEFIELD



Accès (HP-48 G/GX) : [←] [8] [NXT] (3D) (PTYPE) (SLOPE) **Drapeaux** : -

Le mode graphique SLOPEFIELD permet les représentations graphiques sous forme de fonctions scalaires à deux variables indépendantes. SLOPEFIELD, comme toute commande activant un mode graphique, SLOPEFIELD modifie le rôle de DRAW.

SPHERE



Coordonnées sphériques

Accès (HP-48 G/GX) : [MTH] (VECTR) [NXT] (SPHER) Drapeaux : -16, -15

SPHERE active le mode de coordonnées sphériques.

SQ

Carré

Accès : [\leftarrow] [\sqrt{x}]

Drapeaux : -3

| | | | | |
|----|-------------|---|----|-----------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'SQ(expr_symb)' |
| 1: | [[tableau]] | → | 1: | [[tableau]] |



SQ élève au carré son argument qui peut être un nombre réel, un nombre complexe, une expression symbolique, un vecteur ou une matrice, ces deux derniers pouvant être réels ou complexes.

SR

Décalage d'un bit vers la droite

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BIT) (SR) Drapeaux : -12 à -5

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] [NXT] (SR)

| | | | | |
|----|----------------|---|----|----------------|
| 1: | entier_binaire | → | 1: | entier_binaire |
|----|----------------|---|----|----------------|

SR prend en argument un entier binaire (quelle que soit la base avec laquelle il est exprimé, il est ramené à sa forme binaire, c'est-à-dire au maximum 64 bits). SR décale tous les bits de l'entier binaire d'un rang vers la droite et provoque ainsi une division entière par deux.

SRAD



Rayon spectral d'une matrice

Accès (HP-48 G/GX) : [MTH] (MATR) (NORM) (SRAD) Drapeaux : -

| | | | | |
|----|-------------|---|----|--------|
| 1: | [[matrice]] | → | 1: | entier |
|----|-------------|---|----|--------|

SRAD renvoie un entier correspondant au rayon spectral de la matrice donnée en argument. Cette matrice doit être carrée (autant de lignes que de colonnes).

SRB

Décalage d'un octet vers la droite

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (BYTE) (SRB) Drapeaux : -12 à -5

Accès (HP-48 S/SX) : [MTH] (BASE) [NXT] [NXT] (SRB)

1: entier_binaire → 1: entier_binaire



SRB prend en argument un entier binaire (quelle que soit la base avec laquelle il est exprimé, il est ramené à sa forme binaire, c'est-à-dire au maximum 64 bits). SRB décale tous les bits de l'entier binaire de huit rangs vers la droite et provoque ainsi une division entière par 256.

SRECV

Lecture de caractères reçus dans le buffer série

Accès (HP-48 G/GX) : [←] [1] [NXT] (SERIA) (SRECV) Drapeaux : -33

Accès (HP-48 S/SX) : [←] [PRG] [NXT] [NXT] (SRECV)

2: "chaîne_reçue"
1: entier → 2: "chaîne_reçue"
1: erreur



SRECV suppose l'emploi de OPENIO (ouverture du port série) et de STIME (définition du temps d'attente des caractères). erreur est un entier valant 1 si aucune erreur n'a été détectée. erreur vaut 0 en cas d'erreur. "chaîne_reçue" comporte un nombre de caractères défini par l'entier donné en argument.

SST

Exécution pas à pas

Accès (HP-48 G/GX) : [PRG] [NXT] (RUN) (SST) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) (SST)

SST exécute un programme pas à pas (évaluation de la prochaine instruction ou évaluation de l'ensemble d'un sous-programme).

SST↓

Exécution pas à pas

Accès (HP-48 G/GX) : [PRG] [NXT] (RUN) (SST↓) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) (SST↓)

SST exécute un programme pas à pas (évaluation de la prochaine instruction ou évaluation de la première instruction d'un sous-programme).

START

En-tête de structure START... STEP ou START... NEXT

Accès (HP-48 G/GX) : [PRG] (BRCH) (START) (START) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) (START)



START constitue l'en-tête d'une boucle START... NEXT ou START... STEP.

Structure **START... NEXT** :

```
val_deb val_fin START
    instructions de la boucle
```

NEXT

Structure **START... STEP** :

```
val_deb val_fin START
    instructions de la boucle
pas STEP
```

Dans le premier cas, une variable de boucle varie entre les valeurs `val_deb` et `val_fin` et est augmentée de une unité après chaque exécution des instructions de la boucle. Le réel `val_fin` limite le nombre d'exécutions de la boucle. Le second cas est identique à ceci près que la variable de boucle est augmentée de la valeur de `pas` (qui est un réel ou assimilable à un réel après évaluation) après chaque exécution des instructions de la boucle.

STD

Mode standard d'affichage des nombres

Accès (HP-48 G/GX) : [←] [CST] (FMT) (STD) Drapeaux : -50, -49

Accès (HP-48 S/SX) : [←] [CST] (STD)



STD active le mode d'affichage standard des nombres. Autrement dit, les nombres sont affichés sans facteur dix tant que les douze chiffres affichables suffisent.

STEP

Définition du pas d'incrémentation d'une variable de boucle

Accès (HP-48 G/GX) : [PRG] (BRCH) (FOR) (STEP) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] (STEP)

STEP fixe la valeur du pas d'incrémentation d'une variable de boucle (voir FOR et START).

STEQ

Affectation d'un objet à la variable réservée EQ

Accès (HP-48 G/GX) : [←] [8] [←] (EQ) Drapeaux : -

Accès (HP-48 S/SX) : [←] [8] (STEQ)

| | | | |
|----|-------|---|----|
| 1: | objet | → | 1: |
|----|-------|---|----|

STEQ est équivalent à 'EQ' STO et place donc dans la variable EQ du répertoire courant (cette variable est créée si elle n'existe pas dans le répertoire courant) l'objet se trouvant au niveau 1 de la pile. Attention ! Aucune vérification de la nature de EQ n'est effectuée par STEQ.



STIME

Temporisation

Accès (HP-48 G/GX) : [←] [1] [NXT] (SERIA) (STIME) Drapeaux : -

Accès (HP-48 S/SX) : [←] [PRG] [NXT] [NXT] (STIME)

| | | | |
|----|-------|---|----|
| 1: | tempo | → | 1: |
|----|-------|---|----|

tempo étant un délai exprimé en secondes, STIME fixe ce délai comme temps maximal d'attente lors d'une émission (voir XMIT) ou une réception (voir SRECV) empruntant la voie série. si tempo vaut 0, le délai d'attente est infini. Si tempo est non nul, sa valeur ne doit pas dépasser 25 secondes. Par défaut, le temps d'attente est fixé à 10 secondes.

STO

Affectation d'un objet à une variable ou création d'une variable

Accès : [STO] Drapeaux : -

| | | | |
|----|----------------------|---|----|
| 2: | objet | → | 2: |
| 1: | 'nom' | | 1: |
| 2: | objet | → | 2: |
| 1: | :num_port:sauvegarde | | 1: |
| 2: | bibliothèque | → | 2: |
| 1: | num_port | | 1: |
| 2: | sauvegarde | → | 2: |
| 1: | num_port | | 1: |



Dans le cas général, STO affecte un objet à la variable nommée 'nom' du répertoire courant (cette variable est créée si elle n'existe pas dans le répertoire courant). num_port étant un numéro de port, STO peut aussi créer un objet-sauvegarde.

STO+

Addition appliquée au contenu d'une variable

Accès (HP-48 G/GX) : [\leftarrow] [VAR] (ARITH) (STO+) Drapeaux : -

Accès (HP-48 S/SX) : [\rightarrow] [VAR] (STO+)



STO+ ajoute objet (opération +) au contenu de la variable nommée 'nom' du répertoire courant (cette variable doit impérativement exister dans le répertoire courant). Le contenu de la variable 'nom' est donc modifié par STO+. objet et 'nom' peuvent chacun se trouver indifféremment au niveau 1 ou au niveau 2 de la pile.

STO-

Soustraction appliquée au contenu d'une variable

Accès (HP-48 G/GX) : [\leftarrow] [VAR] (ARITH) (STO-) Drapeaux : -

Accès (HP-48 S/SX) : [\rightarrow] [VAR] (STO-)



STO- soustrait objet (opération -) au contenu de la variable nommée 'nom' du répertoire courant (cette variable doit impérativement exister dans le répertoire courant). Le contenu de la variable 'nom' est donc modifié par STO-. objet et 'nom' peuvent chacun se trouver indifféremment au niveau 1 ou au niveau 2 de la pile.

STO*

Multiplication appliquée au contenu d'une variable

Accès (HP-48 G/GX) : [\leftarrow] [VAR] (ARITH) (STO*) Drapeaux : -

Accès (HP-48 S/SX) : [\rightarrow] [VAR] (STO*)



STO* multiplie le contenu de la variable nommée 'nom' du répertoire courant (cette variable doit impérativement exister dans le répertoire courant) par objet (opération *). Le contenu de la variable 'nom' est donc modifié par STO*. objet et 'nom' peuvent chacun se trouver indifféremment au niveau 1 ou au niveau 2 de la pile.

STO/

Division appliquée au contenu d'une variable

Accès (HP-48 G/GX) : [←] [VAR] (ARITH) (STO/) Drapeaux : -

Accès (HP-48 S/SX) : [→] [VAR] (STO/)



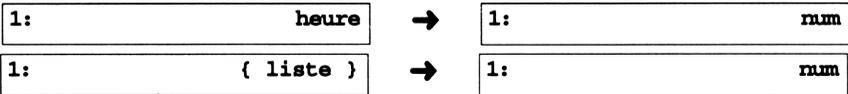
STO/ divise le contenu de la variable nommée 'nom' du répertoire courant (cette variable doit impérativement exister dans le répertoire courant) par objet (opération /). Le contenu de la variable 'nom' est donc modifié par STO/. objet et 'nom' peuvent chacun se trouver indifféremment au niveau 1 ou au niveau 2 de la pile.

STOALARM

Ajout d'une sonnerie programmée à la liste des alarmes système

Accès (HP-48 G/GX) : [←] [4] (ALRM) (STOAL) Drapeaux : -44 à -42

Accès (HP-48 S/SX) : [←] [4] (ALRM) [NXT] (STOAL)



STOALARM ajoute une alarme (sonnerie programmée) à la liste des alarmes et renvoie num qui est le numéro de cette alarme dans la liste des alarmes. Pour programmer une alarme à déclencher pendant les prochaines 24 heures, il suffit de donner une heure en argument à STOALARM, ainsi 17.1520 STOALARM programme une alarme à déclencher à 17h 15 minutes et 20 secondes. { liste } permet de spécifier une date, un objet à évaluer et un délai de répétition sous la forme { date heure objet_à_évaluer délai }. date est alors de la forme mois.jour, autrement dit, le 25 décembre est représenté par 12.25 alors que délai est exprimé en 1/8192^e de seconde (8192 représente une seconde).

STOF

Définition des états des indicateurs

Accès (HP-48 G/GX) : [←] [CST] (FLAG) [NXT] (STOF) Drapeaux : -10 à -5

Accès (HP-48 S/SX) : [→] [CST] [NXT] (STOF) →



→ **STOF** (*store flag*) définit les états des drapeaux (64 flags système et 64 flags utilisateur). *états* est un entier binaire (de la forme #nb) comportant 64 bits. Ces 64 bits correspondent aux états des *flags* (drapeaux) système -1 à -64. Les bits valant 1 arment les flags correspondant. Si *états* comporte moins de 64 bits (selon la taille des entiers binaires), les bits manquant sont considérés comme valant 0 et désarment donc les derniers *flags*. *états* peut aussi être une liste contenant deux entiers binaires de 64 bits. Dans ce cas, le premier entier binaire définit les états des 64 *flags* système alors que le second entier binaire définit les états des 64 *flags* utilisateurs.

STOKEYS

Affectations des touches du clavier

Accès (HP-48 G/GX) : [↵] [CST] (KEYS) (STOK) Drapeaux : -62 à -61

Accès (HP-48 S/SX) : [↵] [CST] (STOK)

1: { liste } → 1: []

1: 'S' → 1: []



Si son argument est 'S', STOKEYS permet l'utilisation des touches non redéfinies avec leurs fonctions standard. { liste } est de la forme { objet_1 touche_1 objet_2 touche_2... } et provoque l'évaluation de objet_n en cas de pression sur touche_n. touche_n est de la forme LigneColonne.Shift. Ainsi, 32.2 désigne [↵] [STO] (voir ASN pour le codage de Shift). { liste } peut commencer par S pour autoriser l'utilisation normale des touches non redéfinies.

STOΣ

Affectation d'un objet à la variable ΣDAT

Accès (HP-48 G/GX) : [↵] [5] (DATA) [↵] (ΣDAT) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [5] (STOΣ)

1: objet → 1: []

STOΣ est équivalent à 'ΣDAT' STO et place donc dans la variable réservée ΣDAT du répertoire courant (cette variable est créée si elle n'existe pas dans le répertoire courant) l'objet se trouvant au niveau 1 de la pile.



Attention ! Aucune vérification de la nature de l'objet n'est effectuée par STOΣ. Une erreur relative à la nature ΣDAT peut donc ensuite survenir.

→STR

Place un objet dans une chaîne de caractères

Accès (HP-48 G/GX) : [PRG] (TYPE) (→STR) Drapeaux : -50, -49, -12 à -5

Accès (HP-48 S/SX) :

| | | | | |
|----|-------|---|----|----------|
| 1: | objet | → | 1: | "chaîne" |
|----|-------|---|----|----------|

→STR place un objet quelconque dans une chaîne et renvoie l'argument «encapsulé» dans une chaîne tout en respectant les espaces et les sauts de lignes. Ainsi, 'A' →STR renvoie " 'A' ".



STR→

Evaluation du contenu d'une chaîne de caractères

Accès : [α] [α] [SIN] [COS] [→] [↵] [0] Drapeaux : -

| | | | | |
|----|----------|---|----|--------------|
| 1: | "chaîne" | → | 1: | selon le cas |
|----|----------|---|----|--------------|

STR→ extrait le contenu d'une chaîne de caractères, l'insère dans la ligne de commande et l'évalue. OBJ→ a, sur les chaînes, le même effet que STR→.



STREAM

Exécution récursive



Accès (HP-48 G/GX) : [PRG] (LIST) (PROC) () Drapeaux : -

| | | | | |
|----|-----------|---|----|--------------|
| 2: | { liste } | → | 2: | selon le cas |
| 1: | programme | | 1: | |

STREAM extrait les deux premiers éléments de { liste } puis exécute programme et place le résultat dans la pile. L'élément suivant de la liste est alors extrait et placé dans la pile avant une nouvelle exécution de programme. programme est ainsi appliqué à son précédent résultat et à chaque élément de la liste tant que la fin de celle-ci n'est pas atteinte.

STWS

Taille du mot binaire

Accès (HP-48 G/GX) : [MTH] (BASE) [NXT] (STWS) Drapeaux : -12 à -5

Accès (HP-48 S/SX) : [MTH] (BASE) (STWS) →

| | | | | |
|----|--------|---|----|--|
| 1: | entier | → | 1: | |
|----|--------|---|----|--|

| | | | | |
|----|----------------|---|----|--|
| 1: | entier binaire | → | 1: | |
|----|----------------|---|----|--|

→ *STWS* prend en argument un entier ou un entier binaire compris entre 1 et 64 (respectivement, entre #1d et #64d). Cet entier définit le nombre de bits composant les entiers binaires. Si les entiers binaires saisis par l'utilisateur dépassent la taille du mot binaire défini avec *STWS*, les bits de poids fort de l'entier binaire introduit sont tronqués pour n'afficher de cet entier binaire que le nombre de bits autorisés.

SUB

Extraction d'une portion d'un objet composite

Accès (HP-48 G/GX) : [PRG] (LIST) (SUB)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] [NXT] (SUB)

| | | |
|---|---|---|
| 3: [[tableau_arg]] 2: pos_deb 1: pos_fin | → | 3: 2: 1: [[tableau_res]] |
| 3: objet graphique 2: pos_deb 1: pos_fin | → | 3: 2: 1: objet_graphique |
| 3: "chaîne_arg" 2: pos_deb 1: pos_fin | → | 3: 2: 1: "chaîne_res" |
| 3: { liste_arg } 2: pos_deb 1: pos_fin | → | 3: 2: 1: { liste_res } |



SUB prend en argument un objet composite (tableau, objet graphique, chaîne de caractères ou liste) et crée un objet composite de même type fondé sur l'extraction de la partie délimitée par *pos_deb* et *pos_fin* de l'objet composite donné en argument. Dans le cas d'une chaîne ou d'une liste, *pos_deb* et *pos_fin* sont simplement les numéros d'ordre du premier et du dernier élément (ou caractère à extraire).

Dans le cas d'un tableau, *pos_deb* et *pos_fin* peuvent, soit être le numéro de l'élément compté à partir de l'élément se trouvant en haut à gauche, soit être une liste { ligne colonne } déterminant la position des éléments à extraire. Par exemple, [4 5 6 7] 2 3 SUB renvoie [5 6]. Dans le cas d'une matrice : [[1 2 3] [4 5 6]] { 1 2 } { 2 3 } SUB renvoie [[2 3] [5 6]].

Dans le cas d'un objet graphique, *pos_deb* et *pos_fin* sont les coordonnées utilisateur (x, y) ou matérielles {#x #y} des extrémités de la première diagonale du rectangle à extraire de l'objet graphique donné en argument.

SVD



Décomposition en valeurs singulières

Accès (HP-48 G/GX) : [MTH] (MATR) (FACTR) (SVD) Drapeaux : -

| | | | | |
|----|-----------------|---|----|---------------|
| 3: | | → | 3: | [[matrice_U]] |
| 2: | | | 2: | [[matrice_V]] |
| 1: | [[matrice_arg]] | | 1: | [vecteur_S] |

[[matrice_arg]] est de dimensions $l \times c$. [[matrice_U]] et [[matrice_V]] ont $l \times l$ et $c \times c$ pour dimensions respectives. [vecteur_S] correspond au vecteur renvoyé par SVL (voir SVL). On a :
 $[[matrice_arg]] = [[matrice_U]] \times [[matrice_DS]] \times [[matrice_V]]$. Ici, [[matrice_DS]] est une matrice $l \times c$ dont les éléments diagonaux sont ceux de [vecteur_S].

SVL

Décomposition en valeurs singulières

Accès (HP-48 G/GX) : [MTH] (MATR) (FACTR) [NXT] (SVL) Drapeaux : -

| | | | | |
|----|-----------------|---|----|-------------|
| 1: | [[matrice_arg]] | → | 1: | [vecteur_S] |
|----|-----------------|---|----|-------------|

[[matrice_arg]] est de dimensions $l \times c$. [vecteur_S] est un vecteur dont les éléments réels correspondent aux valeurs singulières de [[matrice_arg]] classées par ordre décroissant. Le nombre d'éléments de [vecteur_S] correspond à la plus petite dimension de [[matrice_arg]].

SWAP

Inversion des contenus des deux premiers niveaux de la pile

Accès : [\leftarrow] [\rightarrow] Drapeaux : -

| | | | | |
|----|---------|---|----|---------|
| 2: | objet_a | → | 2: | objet_b |
| 1: | objet_b | | 1: | objet_a |



SWAP inverse les contenus des deux premiers niveaux de la pile sans modifier les objets qui s'y trouvent. Voir aussi DUP et ROLL.

SYSEVAL

Evaluation d'un objet dont l'adresse est donnée en argument

Accès : [α] [α] [SIN] [+/-] [SIN] [E] [\sqrt{x}] [A] [NXT] → Drapeaux : -

| | | | | |
|----|--|---|----|--|
| 1: | | → | 1: | |
|----|--|---|----|--|

→ SYSEVAL évalue un objet en mémoire dont l'adresse est donnée sous la forme d'un entier binaire. Les adresses sont codées sur cinq quartets et s'étendent de #00000h à #FFFFFFh.

%T

Pourcentage

Accès (HP-48 G/GX) : [MTH] (REAL) (%T)

Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] (%T)

| | |
|----|------|
| 2: | réel |
| 1: | réel |



| | |
|----|------|
| 2: | |
| 1: | réel |

| | |
|----|--------------|
| 2: | 'expr_symb1' |
| 1: | 'expr_symb2' |



| | |
|----|-----------------------------|
| 2: | |
| 1: | '%T(expr_symb1,expr_symb2)' |



%T renvoie le pourcentage que représente la valeur du niveau 1 par rapport à celle du niveau 2. Ainsi, 2 .75 %T renvoie 37.5, c'est-à-dire 37,5%.

→TAG

Création d'un objet associé à une étiquette

Accès (HP-48 G/GX) : [PRG] (TYPE) (→TAG)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) (→TAG)

| | |
|----|-------|
| 2: | objet |
| 1: | tag |



| | |
|----|------------|
| 2: | |
| 1: | :tag:objet |



→TAG associe une étiquette tag à un objet objet. tag peut être un nombre réel, un nom (placé entre apostrophes) ou une chaîne de caractères (limitée à 255 caractères et entourée des délimiteurs " et ").

TAIL

Suppression du premier élément ou du premier caractère



Accès (HP-48 G/GX) : [PRG] (LIST) (ELEM) [NXT] (TAIL)

Drapeaux : -

| | |
|----|-------------------------|
| 1: | { elt1 elt2 ... elt n } |
|----|-------------------------|



| | |
|----|--------------------------|
| 1: | { elt2 elt 3 ... elt n } |
|----|--------------------------|

| | |
|----|----------|
| 1: | "abcdef" |
|----|----------|



| | |
|----|---------|
| 1: | "bcdef" |
|----|---------|

TAIL prend en argument une liste ou une chaîne de caractères. TAIL renvoie soit la liste-argument privée de son premier élément, soit la chaîne-argument privée de son premier caractère. Par exemple, { A B C D } TAIL renvoie { B C D }, de même, "ABCD" TAIL renvoie "BCD".

TAN

Tangente

Accès : [TAN]

Drapeaux : -22, -18, -17, -3

| | | | | |
|----|-------------|---|----|------------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'TAN(expr_symb)' |

TAN renvoie la tangente de son argument. L'unité de mesure d'angle courante (ou celle précisée par un suffixe si l'argument réel est associé à une unité) est utilisée.



TANH

Tangente hyperbolique

Accès : [MTH] (HYP) (TANH)

Drapeaux : -22, -18, -17, -3

| | | | | |
|----|-------------|---|----|-------------------|
| 1: | réel | → | 1: | réel |
| 1: | complexe | → | 1: | complexe |
| 1: | 'expr_symb' | → | 1: | 'TANH(expr_symb)' |

TANH renvoie la tangente hyperbolique de son argument.

TAYLR

Polynôme de Taylor

Accès : [↵] [9] (TAYLR)

Drapeaux : -

| | | | | |
|----|-------------|---|----|-------------|
| 3: | 'expr_symb' | → | 3: | |
| 2: | 'var' | | 2: | |
| 1: | ordre | | 1: | 'expr_symb' |

TAYLR prend en argument une expression symbolique, le nom 'var' de l'une des variables de cette expression symbolique ainsi qu'un entier représentant l'ordre du polynôme de Taylor à calculer. Celui-ci est renvoyé sous la forme d'une expression symbolique.

TDELTA

Différence de température



Accès (HP-48 G/GX) : [↵] [3] (UTILS) [NXT] (TDELTA) Drapeaux : -3



| | | |
|-----------------|---|------------------------------------|
| 2: réel | → | 2: réel |
| 1: réel | | 1: réel |
| 2: objet_unité | → | 2: objet_unité |
| 1: objet_unité | | 1: objet_unité |
| 2: 'expr_symb1' | → | 2: 'TDELTA(expr_symb1,expr_symb2)' |
| 1: 'expr_symb2' | | 1: 'TDELTA(expr_symb1,expr_symb2)' |

→ TDELTA calcule une différence de température. Au niveau 2 doit être placée la température finale alors que la température initiale doit être placée au niveau 1. TDELTA est particulièrement intéressant avec des objets unités lorsque les deux températures n'emploient pas la même unité (dans ce cas, l'unité de la température finale est utilisée pour le résultat).

TEACH

Création du répertoire **EXAMPLES**



Accès (HP-48 G/GX) : [α] [α] [COS] [E] [A] [C] [PRG] **Drapeaux** : -

TEACH provoque la création d'un répertoire **EXAMPLES** dans le répertoire HOME. Le répertoire **EXAMPLES** contient divers exemples facilitant l'apprentissage du fonctionnement des HP-48 G/GX. CLTEACH (voir CLTEACH) supprime le répertoire **EXAMPLES**.

TEXT

Affichage de la pile

Accès (HP-48 G/GX) : [PRG] [NXT] (OUT) (TEXT)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) [NXT] [NXT] [NXT] (TEXT)

TEXT provoque l'affichage de la pile pendant l'exécution d'un programme (voir aussi PVIEW).

THEN

En-tête de la procédure à exécuter si la condition requise est remplie

Accès (HP-48 G/GX) : [PRG] (BRCH) (IF) (THEN)

Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] (THEN)

THEN précède les instructions à exécuter si la condition requise est remplie, et ce, dans le cadre de structures débutant par IF (voir IF), IFERR (voir IFERR) ou CASE (voir CASE). Contrairement à THEN, ELSE précède les instructions à exécuter si la condition requise n'est pas satisfaite.



TICKS

Heure système

Accès (HP-48 G/GX) : [↵] [4] (TICKS)

Drapeaux : -

Accès (HP-48 S/SX) : [↵] [4] [NXT] (TICKS)

1: → 1: entier binaire

TICKS place dans la pile un entier binaire correspondant à l'heure système courante. Cette heure est exprimée en $1/8192^e$ de seconde.

TIME

Heure système

Accès (HP-48 G/GX) : [↵] [4] (TIME)

Drapeaux : -

Accès (HP-48 S/SX) : [↵] [4] [NXT] (TIME)

1: → 1: réel

TIME place dans la pile un réel correspondant à l'heure système courante. Cette heure est exprimée au format hh.mmss_{ssss} où hh représente les heures (format 24 heures), mm les minutes, ss les secondes et _{ssss} les fractions décimales de secondes ($1/10^e$, $1/100^e$, etc.). mm et ss sont exprimés sous forme sexagésimale.



→TIME

Heure système

Accès (HP-48 G/GX) : [↵] [4] (→TIM)

Drapeaux : -

Accès (HP-48 S/SX) : [↵] [4] (SET) (→TIM)

1: réel → 1:

→TIME détermine l'heure système à partir du réel donné en argument. Ce réel est au format hh.mmss_{ssss} (voir TIME).

TDELTA

Différence de température

→



Accès (HP-48 G/GX) : [↵] [3] (UTILS) [NXT] (TINC)

Drapeaux : -3

2: réel
1: réel → 2:
1: réel

| | | |
|------------------------------------|---|--|
| 2: objet_unité 1: objet_unité | → | 2: 1: objet_unité |
| 2: `expr_symb1` 1: `expr_symb2` | → | 2: 1: `TINC(expr_symb1,expr_symb2)` |

→ TINC ajoute une différence de température à une température initiale. Au niveau 2 doit être placée la température initiale alors que la différence de température doit être placée au niveau 1. TINC est particulièrement intéressant avec des objets unités lorsque les deux températures n'emploient pas la même unité.

TLINE

Tracé d'un segment de droite par inversion de l'état des pixels

Accès (HP-48 G/GX) : [PRG] (PICT) (TLINE) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (DSPL) (TLINE)

| | | |
|--------------------------|---|----------|
| 2: coord_1 1: coord_2 | → | 2: 1: |
|--------------------------|---|----------|

TLINE inverse l'état de chaque pixel du segment de droite dont les extrémités sont coord_1 et coord_2. Ces deux coordonnées peuvent être de la forme utilisateur (x,y) ou de la forme matérielle { #x #y } mais doivent toutefois être de la même forme. Voir aussi LINE.

TMENU

Menu temporaire

Accès (HP-48 G/GX) : [↵] [CST] (MENU) (TMEN) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [CST] [NXT] (TMEN)

| | | |
|------------|---|----|
| 1: num | → | 1: |
| 1: `nom` | → | 1: |
| 1: { def } | → | 1: |

TMENU affiche un menu défini par son numéro num, une liste { def } le définissant ou le nom `nom` d'une variable contenant une liste définissant un menu. Contrairement à MENU, TMENU ne modifie pas CST.

TOT

Somme des contenus de chaque colonne de ΣDAT

Accès (HP-48 G/GX) : [↵] [5] (1VAR) (TOT) Drapeaux : - →

→ Accès (HP-48 S/SX) : [←] [5] [NXT] (TOT)

| | | |
|----|---|---------------------------------|
| 1: | → | 1: réel |
| 1: | → | 1: { somm1 somm2 ... somm n } |

TOT calcule, pour chaque colonne de Σ_{DAT} , la somme du contenu de la colonne. Si Σ_{DAT} ne contient qu'une seule colonne, la somme de son contenu est renvoyée sous la forme d'un unique réel. Si Σ_{DAT} contient n colonnes, les sommes des contenus de chacune des n colonnes sont renvoyées sous la forme d'une liste de n réels.

TRACE

Trace d'une matrice carrée



Accès (HP-48 G/GX) : [MTH] (MATR) (NORM) [NXT] (TRACE) Drapeaux : -

| | | |
|--------------|---|---------|
| 1: [matrice] | → | 1: réel |
|--------------|---|---------|

TRACE calcule la trace d'une matrice (nécessairement carrée), autrement dit, la somme des éléments diagonaux.

Par exemple, [[1 2 3] [4 5 6] [7 8 9]] TRACE renvoie 15. 15 correspond à la somme des éléments composant la première diagonale, autrement dit $1 + 5 + 9 = 15$.

TRANSIO

Mode de traduction lors des transferts ASCII par Kermit

Accès (HP-48 G/GX) : [←] [1] (IOPAR) (TRAN) Drapeaux :

Accès (HP-48 S/SX) : [←] [PRG] (SETUP) (TRAN)

| | | |
|---------|---|----|
| 1: mode | → | 1: |
|---------|---|----|

TRANSIO détermine, en fonction de mode, le mode de traduction des données transmises par la voie série en mode ASCII à l'aide du protocole Kermit. Valeur possible pour l'entier mode :

- 0 → absence de traduction,
- 1 → selon le sens de transmission, le saut de ligne (LF) est automatiquement associé au retour chariot (CR), ou bien, le retour chariot est automatiquement supprimé à la suite d'un saut de ligne,
- 2 → les symboles codés par les valeurs comprises entre 128 et 159 sont adaptés,
- 3 → les symboles codés par les valeurs comprises entre 128 et 255 sont adaptés.

TRN

Matrice conjuguée (transposition de matrice)

Accès (HP-48 G/GX) : [MTH] (MATR) (MAKE) (TRN) Drapeaux : -

Accès (HP-48 S/SX) : [MTH] (MATR) (TRN)

| | | | | |
|----|-------------|---|----|-------------|
| 1: | [[matrice]] | → | 1: | [[matrice]] |
|----|-------------|---|----|-------------|

| | | | | |
|----|---------------|---|----|--|
| 1: | `nom_matrice` | → | 1: | |
|----|---------------|---|----|--|

TRN renvoie la matrice conjuguée calculée à partir de la matrice argument. TRN peut aussi prendre en argument le nom d'une variable contenant une matrice. Dans ce cas, la matrice est effectivement modifiée mais TRN ne place pas le résultat dans la pile. Par exemple [[1 2] [3 4] [5 6]] TRN renvoie [[1 3 5] [2 4 6]].



TRNC

Adaptation du format d'affichage des nombres

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] [NXT] (TRNC) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] [NXT] (TRNC)

| | | | | |
|----|------|---|----|------|
| 2: | réel | → | 2: | réel |
| 1: | mode | | 1: | |

| | | | | |
|----|----------|---|----|----------|
| 2: | complexe | → | 2: | complexe |
| 1: | mode | | 1: | |

| | | | | |
|----|-------------|---|----|-------------|
| 2: | [[tableau]] | → | 2: | [[tableau]] |
| 1: | mode | | 1: | |

| | | | | |
|----|------------|---|----|------------|
| 2: | réel_unité | → | 2: | réel_unité |
| 1: | mode | | 1: | |

| | | | | |
|----|-------------|---|----|------------------------|
| 2: | `expr_symb` | → | 2: | `TRNC(expr_symb,mode)` |
| 1: | mode | | 1: | |

TRNC modifie le format d'affichage, le nombre de décimales ou le nombre de chiffres significatifs des réels. TRNC prend en argument un réel ou des objets pouvant être ramenés à des réels (un complexe est un couple de réels). TRNC s'adapte donc aux réels, complexes, tableaux réels ou complexes, objets-unités et expressions symboliques. La modification du format est fonction de mode qui est un entier compris entre -11 et 11 ou une expression symbolique pouvant être ramenée à un entier. Valeurs de mode :

- 12 → activation du format courant,
- 0 à 11 → de 0 à 11 décimales (en fonction de mode),
- -1 à -11 → de 1 à 11 chiffres significatifs (en fonction de mode).

TRUTH

Mode de tracé TRUTH

Accès : [←] [8] (PTYPE) (TRUTH) Drapeaux : -

TRUTH active le mode graphique permettant la représentation graphique de fonctions booléennes de deux variables réelles.

TSTR

Création d'une chaîne de caractères à partir de la date et de l'heure

Accès (HP-48 G/GX) : [←] [4] [NXT] [NXT] (TSTR) Drapeaux : -41, -42

Accès (HP-48 S/SX) : [←] [4] [NXT] (TSTR)

| | | |
|---|---|--|
| 2: date 1: heure | → | 2: "chaîne" 1: "chaîne" |
|---|---|--|

TSTR se fonde sur une date, par exemple 7.141996 (14/7/1996 au format US), et sur une heure, par exemple 15.30 (15h30) pour construire une chaîne incluant la date et l'heure. Dans le cadre de cet exemple, 7.141996 15.30 TSTR renvoie "SUN 07/14/96 03:30:00P".



TVARS

Liste des variables d'un type donné dans le répertoire courant

Accès (HP-48 G/GX) : [←] [VAR] (DIR) (TVARS) Drapeaux : -

Accès (HP-48 S/SX) : [←] [VAR] [NXT] (TVARS)

| | | |
|---|---|---|
| 1: type | → | 1: { var1 var2 ... } |
| 1: { type1 type2 ... } | → | 1: { var1 var2 ... } |

type étant un entier représentant un type (voir TYPE), TVARS renvoie sous forme de liste les noms des variables du type spécifié présentes dans le répertoire courant. Il est possible de rechercher plusieurs types en les spécifiant à l'aide d'une liste.

TVM

Accès au menu du solveur TVM



Accès : [α] [α] [COS] [√x] ['] Drapeaux : -

TVM provoque l'affichage du menu relatif au solveur TVM. Voir aussi TVMBEG, TVMEND et TVMROOT.

TVMBEG



Paiements en début de période

Accès : [α] [α] [COS] [\sqrt{x}] ['] [B] [E] [MTH] **Drapeaux : -**

TVMBEG précise que, pour les calculs TVM, les paiements ont lieu au début de chaque période.

TVMEND



Paiements en fin de période

Accès : [α] [α] [COS] [\sqrt{x}] ['] [E] [STO] [D] **Drapeaux : -**

TVMEND précise que, pour les calculs TVM, les paiements ont lieu à la fin de chaque période.

TVMROOT



Solution TVM

Accès : [\leftarrow] [7] (TVM) (TVMR) **Drapeaux : -**

1: → 1:

TVM calcule la solution TVM à partir de la variable TVM dont le nom est donné est argument.

TYPE

Affichage du type d'un objet

Accès (HP-48 G/GX) : [PRG] (TEST) [NXT] (TYPE) **Drapeaux : -**

Accès (HP-48 S/SX) : [PRG] (TEST) (TYPE)

1: → 1:

TYPE prend n'importe quel objet en argument et renvoie son type sous la forme d'un entier représentant le type.

| Valeur de type | Type | Valeur de type | Type |
|----------------|---------------------------------|----------------|----------------------------|
| 0 | Reel | 14 | XLIB name |
| 1 | Complexe | 15 | (sous-) répertoire |
| 2 | Chaîne de caractères | 16 | Bibliothèque (library) |
| 3 | Matrice ou vecteur de réels | 17 | Objet-sauvegarde |
| 4 | Matrice ou vecteur de complexes | 18 | Fonction standard |
| 5 | Liste | 19 | Commande standard |
| 6 | Nom de variable globale | 20 | Nombre binaire système |
| 7 | Nom de variable locale | 21 | Nombre réel étendu |
| 8 | Programme RPL | 22 | Tableau lié |
| 9 | Expression algébrique | 23 | Caractère (hors chaîne) |
| 10 | Entier binaire | 24 | Code exécutable |
| 11 | GROB (objet graphique) | 25 | Contenu d'une bibliothèque |
| 12 | Objet-tag (objet identifié) | 26 | Objets dits «externes» |
| 13 | Objet-unité | >26 | |

UBASE

Conversion d'une unité en unité du système international

Accès (HP-48 G/GX) : [\leftarrow] [6] (UBASE) Drapeaux : -3Accès (HP-48 S/SX) : [\rightarrow] [6] (UBASE)

| | | | | |
|----|-------------|---|----|-------------|
| 1: | objet-unité | → | 1: | objet-unité |
|----|-------------|---|----|-------------|

UBASE prend en argument un objet-unité dont l'unité est quelconque et renvoie un objet-unité équivalent mais dont l'unité appartient au système international.

**UFACT**

Factorisation d'unités

Accès (HP-48 G/GX) : [\leftarrow] [6] (UFACT) Drapeaux : -Accès (HP-48 S/SX) : [\rightarrow] [6] (UFACT)

| | | | | |
|----|-------------|---|----|-------------|
| 2: | objet_unité | → | 2: | objet_unité |
| 1: | objet_unité | | 1: | objet_unité |

UFACT prend en argument deux objets-unités et renvoie un nouvel objet-unité dans lequel l'unité initialement placée au niveau 1 se retrouve en facteur dans une expression représentant l'unité de niveau 2. Par exemple $1_V \ 1_A \ UFACT$ renvoie $1_kg * m^2 / (A * s^3)$.

→UNIT

Création d'un objet-unité

Accès (HP-48 G/GX) : [PRG] (TYPE) (→UNIT) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] (→UNIT)

| | | | | |
|----|-------------|---|----|-------------|
| 2: | réel | → | 2: | objet_unité |
| 1: | objet_unité | | 1: | objet_unité |

→UNIT prend un réel et un objet-unité dont seule l'unité est significative. →UNIT crée un nouvel objet-unité dont la partie réelle est le réel donné en argument et dont l'unité est l'unité de l'objet-unité donné en argument.

**UNTIL**

Introduction du test dans une structure Do... Until... End

Accès (HP-48 G/GX) : [PRG] (BRCH) (DO) (UNTIL) Drapeaux : - →

Accès (HP-48 S/SX) : [PRG] (BRCH) [NXT] (UNTIL)

→ UNTIL se place avant le test (condition) intégré à une structure Do... Until... End. Voir DO.

UPDIR

Retour au répertoire-père

Accès : [←] [']

Drapeaux : -

UPDIR fait du répertoire-père le répertoire courant. Rappelons que le répertoire-père est le répertoire du chemin courant qui précède le répertoire courant.



UTPC

Distribution Khi carré à droite

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) [NXT] (UTPC) **Drapeaux :** -

Accès (HP-48 S/SX) : [MTH] (PROB) [NXT] (UTPC)

| | | | | |
|----|---------------|---|----|------|
| 2: | nb_degrés_lib | → | 2: | |
| 1: | réel | | 1: | réel |

UTPC calcule la probabilité que Khi carré (variable aléatoire) soit plus grande que le réel donné en argument tout en respectant le nombre de degrés de liberté spécifié au niveau 2.

UTPF

Distribution F de Snedecor à droite

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) [NXT] (UTPF) **Drapeaux :** -

Accès (HP-48 S/SX) : [MTH] (PROB) [NXT] (UTPF)

| | | | | |
|----|---------------------|---|----|------|
| 3: | nb_degrés_lib_num | → | 3: | |
| 2: | nb_degrés_lib_dénom | | 2: | |
| 1: | réel | | 1: | réel |

UTPF calcule la probabilité que F (variable aléatoire) soit plus grande que le réel donné en argument tout en respectant le nombre de degrés de liberté du numérateur et du dénominateur (pour la distribution F).

UTPN

Distribution normale à droite

→

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) [NXT] (UTPN) **Drapeaux** : -

Accès (HP-48 S/SX) : [MTH] (PROB) [NXT] (UTPN)

| | | | | |
|----|----------------------|---|----|------|
| 3: | moyenne | → | 3: | |
| 2: | distribution normale | | 2: | |
| 1: | réel | | 1: | réel |

→ UTPF calcule la probabilité qu'une variable aléatoire soit plus grande que le réel donné en argument tout en respectant la moyenne et la distribution normale données en arguments.

UTPT

Distribution t de Student à droite

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) [NXT] (UTPT) **Drapeaux** : -

Accès (HP-48 S/SX) : [MTH] (PROB) [NXT] (UTPT)

| | | | | |
|----|---------------|---|----|------|
| 2: | nb_degrés_lib | → | 2: | |
| 1: | réel | | 1: | réel |

UTPT calcule la probabilité que t (variable aléatoire de Student) soit plus grande que le réel donné en argument tout en respectant le nombre de degrés de liberté spécifié au niveau 2.

UFACT

Extraction de la partie réelle d'un objet-unité

Accès (HP-48 G/GX) : [←] [6] (UVAL) **Drapeaux** : -

Accès (HP-48 S/SX) : [→] [6] (UVAL)

| | | | | |
|----|-------------|---|----|------|
| 1: | objet-unité | → | 1: | réel |
|----|-------------|---|----|------|

UFACT renvoie le réel constituant la partie réelle de l'objet-unité donné en argument.

→V2

Création d'un vecteur du plan ou d'un complexe

Accès (HP-48 G/GX) : [MTH] (VECTR) (→V2) **Drapeaux** : -16, -19

Accès (HP-48 S/SX) : [MTH] (VECTR) [NXT] (→V2) →

| | | | | |
|----|--------|---|----|------------------|
| 2: | réel_1 | → | 2: | |
| 1: | réel_2 | | 1: | (réel_1, réel_2) |

| | | | | |
|----|--------|---|----|-------------------|
| 2: | réel_1 | → | 2: | |
| 1: | réel_2 | | 1: | [réel_1 réel_2] |

→ →V2 associe deux réels pour créer un vecteur (*flag* -19 désarmé) ou un nombre complexe (*flag* -19 armé). Si le *flag* -16 est armé, le mode polaire (coordonnées (r,θ)) est utilisé.

→ **V3**

Création d'un vecteur de l'espace

Accès (HP-48 G/GX) : [MTH] (VECTR) (→V3) Drapeaux : -16, -15

Accès (HP-48 S/SX) : [MTH] (VECTR) [NXT] (→V3) →



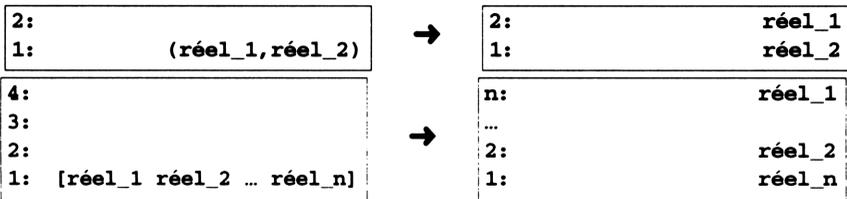
→V3 crée un vecteur de l'espace à partir de trois réels représentant les coordonnées dudit vecteur. Selon les valeurs des *flags* -15 et -16, le mode rectangulaire (coordonnées cartésiennes), le mode polaire cylindrique ou le mode polaire sphérique est utilisé pour les coordonnées.

V→

Séparation des éléments constituant un vecteur ou un complexe

Accès (HP-48 G/GX) : [MTH] (VECTR) (V→) Drapeaux : -15, -16

Accès (HP-48 S/SX) : [MTH] (VECTR) [NXT] (V→)



V→ renvoie dans la pile les parties réelle et imaginaire d'un complexe ou les différents éléments constituant un vecteur. Dans les deux cas, le mode angulaire (*flags* -15 et -16) est respecté.

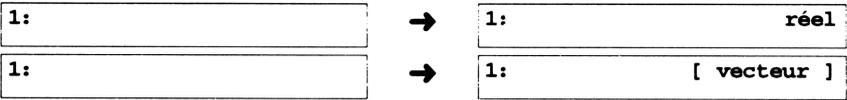


VAR

Variance



Accès (HP-48 G/GX) : [←] [5] (1VAR) [NXT] (VAR) Drapeaux : -



→

→ VAR calcule la variance de chaque colonne de Σ_{DAT} . Si Σ_{DAT} ne comporte qu'une seule colonne, seul un réel est renvoyé. Si Σ_{DAT} comporte plusieurs colonnes, un vecteur contenant les variances des différentes colonnes est renvoyé.

VARS

Liste des variables du répertoire courant

Accès (HP-48 G/GX) : [↵] [VAR] (DIR) (VARS) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [VAR] (VARS)

1: → 1: { liste }



VARS renvoie la liste des noms des variables placées dans le répertoire courant (il s'agit bien entendu de variables globales).

VERSION

Version de la calculatrice



Accès (HP-48 G/GX) : [α] [α] [√x] [E] [→] [SIN] [EVAL] [STO] Drapeaux : -

2: → 2: "chaîne"
1: → 1: "chaîne"

VERSION renvoie deux chaînes de caractères, l'une indique la version de la calculatrice, l'autre est un message de copyright.

VTYPE

Type de l'objet associé à la variable spécifié

Accès (HP-48 G/GX) : [PRG] (TYPE) [NXT] [NXT] (VTYPE) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (OBJ) [NXT] (VTYPE)

1: 'nom' → 1: type

VTYPE prend en argument le nom d'une variable globale et renvoie -1 si cette variable n'existe pas dans le répertoire courant ou affiche un entier déterminant le type de l'objet associé à la variable (voir TYPE pour connaître les valeurs possibles de type).

*W

Zoom horizontal

Accès (HP-48 G/GX) : [↵] [8] (PPAR) [NXT] (*W) Drapeaux : - →

Accès (HP-48 S/SX) : [\leftarrow] [8] (PLOT) [NXT] [NXT] (*W)

1: → 1:

→ *W conserve le centre de la représentation graphique mais modifie l'intervalle de représentation des abscisses en fonction du réel donné en argument qui joue le rôle de facteur d'agrandissement horizontal.

WAIT

Interruption temporaire de l'exécution du programme

Accès (HP-48 G/GX) : [PRG] [NXT] (IN) (WAIT) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (CTRL) [NXT] (WAIT)

1: → 1:

1: → 1:



Si le réel donné en argument est positif et non nul, il correspond au nombre de secondes d'interruption du programme (dans ce cas, WAIT ne renvoie rien). Si le réel donné en argument vaut -1 ou 0, WAIT attend une pression sur une touche autre que [α], [\leftarrow] et [\rightarrow] pour laisser le programme reprendre son exécution (dans ce cas, la touche pressée est renvoyée au format 1c.s, voir ASN). Si l'argument est -1, le menu courant est affiché.

WHILE

En-tête de structure While... Repeat... End

Accès (HP-48 G/GX) : [PRG] (BRCH) (WHILE) (WHILE) Drapeaux : -

Accès (HP-48 S/SX) : [PRG] (BRCH) (WHILE)



WHILE précède le test dans une structure de boucle While... Repeat... end, par exemple :

WHILE *condition à remplir*

REPEAT

instructions de la boucle

END

WIREFRAME



Mode graphique pour tracé de fonctions scalaires à 2 variables

Accès (HP-48 G/GX) : [\leftarrow] [8] [NXT] (3D) (PTYPE) (WIREF) Drapeaux : -

Mode pour vue en perspective de fonctions scalaires à deux variables.

WSLOG



Informations relatives aux quatre derniers redémarrages

Accès (HP-48 G/GX) : [α] [α] [y^x] [SIN] [NXT] [EVAL] [MTH] Drapeaux : -42

4:
3:
2:
1:



4: "code date heure"
3: "code date heure"
2: "code date heure"
1: "code date heure"



WSLOG renvoie quatre chaînes de caractères apportant des informations relatives aux quatre derniers redémarrages de la calculatrice. Chacune de ces chaînes contient la date et l'heure du redémarrage ainsi qu'un code représentant la raison du redémarrage (voir le chapitre *Codes WSLOG* du livre *HP-48 : faites vos jeux en assembleur* édité par DUNOD).

Σx

Somme des valeurs de la variable indépendante (statistiques)

Accès (HP-48 G/GX) : [↵] [5] (SUMS) (Σx) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [5] [NXT] [NXT] [NXT] [NXT] (Σx)

1:



1: réel

Σx renvoie la somme des valeurs prises par la variable indépendante dans ΣDAT (XCOL spécifie la colonne contenant les valeurs de la variable indépendante dans ΣDAT , voir XCOL).

Σx^2

Somme des carrés des valeurs de la variable indépendante (statistiques)

Accès (HP-48 G/GX) : [↵] [5] (SUMS) (Σx^2) Drapeaux : -

Accès (HP-48 S/SX) : [↵] [5] [NXT] [NXT] [NXT] [NXT] (Σx^2)

1:



1: réel

Σx^2 renvoie la somme des carrés des valeurs prises par la variable indépendante dans ΣDAT (XCOL spécifie la colonne contenant les valeurs de la variable indépendante dans ΣDAT , voir XCOL).

XCOL

Colonne de ΣDAT contenant les valeurs de la variable indépendante

Accès (HP-48 G/GX) : [↵] [5] (ΣPAR) (XCOL) Drapeaux : -



Accès (HP-48 S/SX) : [`←`] [5] [NXT] [NXT] (XCOL)

| | | |
|-----------------------------|---|----|
| 1: entier | → | 1: |
|-----------------------------|---|----|

→ XCOL prend en argument un entier correspondant à un numéro de colonne de Σ DAT. La colonne correspondant à ce numéro contient les valeurs de la variable indépendante.

XMIT

Emission d'une chaîne sans utiliser le protocole Kermit

Accès (HP-48 G/GX) : [`←`] [1] [NXT] (SERIA) (XMIT) **Drapeaux :** -33

Accès (HP-48 S/SX) : [`←`] [PRG] [NXT] [NXT] (XMIT)

| | | |
|-------------------------------|---|-------------------------------|
| 2: | → | 2: "chaîne" |
| 1: "chaîne" | | 1: 0 |

| | | |
|-------------------------------|---|------------------------|
| 2: | → | 2: |
| 1: "chaîne" | | 1: 1 |

XMIT émet une chaîne de caractères par la voie série sans faire appel au protocole Kermit. Si la transmission s'est déroulée normalement, 1 est renvoyé. Si ce n'est pas le cas, 0 est renvoyé au niveau 1 alors que le fragment de chaîne non transmis est renvoyé au niveau 2.

XOR

Ou logique exclusif

Accès (HP-48 G/GX) : [PRG] (TEST) [NXT] (XOR) **Drapeaux :** -3, -5 à -10

Accès (HP-48 S/SX) : [PRG] (TEST) (XOR)

| | | |
|---------------------------------------|---|---------------------------------------|
| 2: entier binaire 1 | → | 2: |
| 1: entier binaire 2 | | 1: entier binaire 3 |

| | | |
|---------------------------------|---|---------------------------------|
| 2: "chaîne 1" | → | 2: |
| 1: "chaîne 2" | | 1: "chaîne 3" |

| | | |
|--------------------------------|---|--------------------------------|
| 2: booléen_1 | → | 2: |
| 1: booléen_2 | | 1: booléen_3 |

| | | |
|-----------------------------------|---|--------------------------------|
| 2: 'expr_symb1' | → | 2: |
| 1: 'expr_symb2' | | 1: 'expr_symb1 XOR expr_symb2' |

XOR effectue une comparaison entre les bits de mêmes rangs de ses arguments et renvoie comme résultat de l'opération *Ou exclusif* un objet de même type (les bits armés du résultat ne doivent être armés que dans l'un des deux arguments). Les chaînes de caractères sont perçues comme des chaînes de bits (avec 8 bits pour chaque caractère).

XPON

Extraction de l'exposant du facteur 10

Accès (HP-48 G/GX) : [MTH] (REAL) [NXT] (XPON) **Drapeaux : -3**

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] [NXT] (XPON)

| | | | | |
|----|-------------|---|----|-------------------|
| 1: | réel | → | 1: | entier |
| 1: | 'expr_symb' | → | 1: | 'XPON(expr_symb)' |

XPON renvoie un entier représentant l'exposant du facteur 10 associé au réel donné en argument. Par exemple, $3E5$ représente 3×10^5 , pour extraire l'exposant du facteur 10 : $3E5$ XPON renvoie 5.

XRECV

Réception d'une variable via XMODEM

Accès (HP-48 G/GX) : [↵] [1] [NXT] (XRECV) **Drapeaux : -33, -36**

| | | | | |
|----|-------|---|----|--|
| 1: | 'nom' | → | 1: | |
|----|-------|---|----|--|

XRECV place la HP-48 en mode d'attente de réception d'une variable devant arriver par la voie série en faisant appel au protocole XMODEM. L'objet reçu est placé dans la variable nommée 'nom' du répertoire courant. Plusieurs noms de variables peuvent être placés dans une liste mais dans ce cas, une liste comportant autant de noms doit aussi être utilisée par l'émetteur (voir XSEND).

XRNG

Intervalle de représentation des abscisses

Accès (HP-48 G/GX) : [↵] [8] (PPAR) (XRNG) **Drapeaux : -**

Accès (HP-48 S/SX) : [↵] [8] (PLOT) (XRNG)

| | | | | |
|----|-----|---|----|--|
| 2: | min | → | 2: | |
| 1: | max | | 1: | |

XRNG introduit dans PPAR les valeurs extrêmes de l'intervalle de représentation des abscisses. min et max sont les deux bornes de cet intervalle.

**XROOT**

Racine x-ième

Accès : [↵] [√x] **Drapeaux : -3**

| | | |
|------------------------------------|---|---|
| 2: réel 1: réel | → | 2: 1: réel |
| 2: 'expr_symb1' 1: 'expr_symb2' | → | 2: 1: 'XROOT(expr_symb2,expr_symb1)' |

→ XROOT calcule la racine x-ième. Par exemple, pour calculer $a^{\sqrt{b}}$, placez b au niveau 2 et a au niveau 1.

XSEND S/SX

Emission d'une variable via XMODEM

Accès (HP-48 G/GX) : [←] [1] [NXT] (XSEN) Drapeaux : -33

| | | |
|----------|---|----|
| 1: 'nom' | → | 1: |
|----------|---|----|

XSEND émet la variable nommée 'nom' du répertoire courant par la voie série en employant le protocole XMODEM. Il est possible d'envoyer en une seule opération plusieurs variables du répertoire courant en plaçant leurs noms dans une liste. La HP-48 devant recevoir les variables émises doit se trouver en mode réception à l'aide de XRECV (voir XRECV).

XVOL S/SX

Largeur de vue volumique

Accès (HP-48 G/GX) : [←] [8] [NXT] (3D) (VPAR) (XVOL) Drapeaux : -

| | | |
|------------------------|---|----------|
| 2: gauche 1: droite | → | 2: 1: |
|------------------------|---|----------|

XVOL prend en arguments deux réels fixant les extrémités gauche et droite de la largeur de vue en volume dans le cadre des tracés en trois dimensions.

XXRNG S/SX

Intervalle du domaine de x pour les tracés 3D

Accès (HP-48 G/GX) : [←] [8] [NXT] (3D) (VPAR) (XXRN) Drapeaux : -

| | | |
|------------------|---|----------|
| 2: min 1: max | → | 2: 1: |
|------------------|---|----------|

XXRNG prend en arguments deux réels représentant les bornes de l'intervalle sur lequel varie x (zone d'entrée) pour les tracés 3D de type GRIDMAP et PARSURFACE.

$\Sigma x*y$

Somme des produits des variables indépendante et dépendante

Accès (HP-48 G/GX) : [←] [5] (SUMS) ($\Sigma x*y$) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] [NXT] ($\Sigma x*y$)

1: → 1: réel

$\Sigma x*y$ renvoie la somme des produits de la variable indépendante (voir XCOL) par la variable dépendante (voir YCOL) associée (dans le cadre de ΣDAT).

Σy

Somme des valeurs de la variable dépendante

Accès (HP-48 G/GX) : [←] [5] (SUMS) (Σy) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] [NXT] (Σy)

1: → 1: réel

Σy renvoie la somme des valeurs de la variable dépendante (voir YCOL) dans le cadre de ΣDAT .

Σy^2

Somme des carrés des valeurs de la variable dépendante

Accès (HP-48 G/GX) : [←] [5] (SUMS) (Σy^2) Drapeaux : -

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] [NXT] [NXT] (Σy^2)

1: → 1: réel

Σy^2 renvoie la somme des carrés des valeurs de la variable dépendante (voir YCOL) dans le cadre de ΣDAT .

YCOL

Définition de la colonne de ΣDAT contenant la variable dépendante

Accès (HP-48 G/GX) : [←] [5] (ΣPAR) (YCOL) Drapeaux :

Accès (HP-48 S/SX) : [←] [5] [NXT] [NXT] (YCOL)

1: entier → 1:

YCOL prend en argument un entier, celui-ci est un numéro de colonne de la matrice ΣDAT qui est définie comme contenant les valeurs prises par la variable dépendante (voir aussi XCOL).

YRNG**Intervalle de représentation des ordonnées**

Accès (HP-48 G/GX) : [←] [8] (PPAR) (YRNG) **Drapeaux** : -

Accès (HP-48 S/SX) : [←] [8] (PLOT) (YRNG)

| | | | |
|----|-----|---|----|
| 2: | min | → | 2: |
| 1: | max | | 1: |

YRNG introduit dans PPAR les valeurs extrêmes de l'intervalle de représentation des ordonnées. min et max sont les deux bornes de cet intervalle.

YSLICE**Activation du mode graphique YSLICE**

Accès (HP-48 G/GX) : [←] [8] [NXT] (3D) (PTYPE) (YSLIC) **Drapeaux** : -

YSLICE active le mode graphique permettant le tracé de fonctions scalaires de deux variables (avec vue en section).

YVOL**Profondeur de vue volumique**

Accès (HP-48 G/GX) : [←] [8] [NXT] (3D) (VPAR) (YVOL) **Drapeaux** : -

| | | | |
|----|-----------------|---|----|
| 2: | le plus proche | → | 2: |
| 1: | le plus éloigné | | 1: |

YVOL prend en arguments deux réels fixant l'ordonnée du point représenté le plus proche et celle du point représenté le plus éloigné pour une vue en volume dans le cadre des tracés en trois dimensions. Voir aussi XVOL.

YYRNG**Intervalle du domaine de y pour les tracés 3D**

Accès (HP-48 G/GX) : [←] [8] [NXT] (3D) (VPAR) (YYRN) **Drapeaux** : -

| | | | |
|----|-----------------|---|----|
| 2: | le plus proche | → | 2: |
| 1: | le plus éloigné | | 1: |

YYRNG prend en arguments deux réels représentant les bornes de l'intervalle sur lequel varie y (zone d'entrée) pour les tracés 3D de type GRIDMAP et PARSURFACE.

ZFACTOR



Facteur de correction de compressibilité d'un gaz

Accès (HP-48 G/GX) : [←] [3] (UTILS) (ZFACT) Drapeaux : -3

| | | | | |
|----|-----------|---|----|-----------|
| 2: | Temp_red | → | 2: | |
| 1: | Press_red | | 1: | fact_corr |

| | | | | |
|----|--------------|---|----|----------------------------------|
| 2: | 'expr_symb1' | → | 2: | |
| 1: | 'expr_symb2' | | 1: | 'ZFACTOR(expr_symb1,expr_symb2)' |

ZFACTOR renvoie *fact_corr* qui est un réel représentant le facteur de correction de compressibilité d'un gaz dans le cadre du comportement non idéal d'un hydrocarbure. *Temp_red*, la température réduite, correspond à la température réelle divisée par la température pseudocritique. *Press_red*, la pression réduite, correspond à la pression réelle divisée par la pression pseudocritique.

ZVOL



Hauteur de vue volumique

Accès (HP-48 G/GX) : [←] [8] [NXT] (3D) (VPAR) (ZVOL) Drapeaux : -

| | | | | |
|----|--------------|---|----|--|
| 2: | le plus bas | → | 2: | |
| 1: | le plus haut | | 1: | |

ZVOL prend en arguments deux réels fixant la cote du point représenté le plus bas et celle du point représenté le plus haut pour une vue en volume dans le cadre des tracés en trois dimensions. Voir aussi XVOL et YVOL.



Création de variables locales

Accès : [→] [0] Drapeaux : -

La flèche → est suivie de noms de variables locales (lettres minuscules) utilisée à partir du symbole < suivant.



*

Produit

Accès : [X] Drapeaux : -3, -5 à -10

| | | | | |
|----|------|---|----|------|
| 2: | réel | → | 2: | |
| 1: | réel | | 1: | réel |

* calcule le produit des objets placés aux niveaux 1 et 2 de la pile. * fonctionne non seulement avec les réels mais aussi avec les complexes, →



→ les objets-unités, les expressions algébriques, les matrices, les entiers binaires, etc. Des objets de types différents peuvent être multipliés les uns avec les autres.

+
Addition et concaténation

Accès : [+]

Drapeaux : -3, -5 à -10

| | |
|----|------|
| 2: | réel |
| 1: | réel |



| | |
|----|------|
| 2: | |
| 1: | réel |



Bien entendu, + effectue la somme de deux réels placés aux niveaux 1 et 2 de la pile mais + peut aussi être utilisé avec les complexes, les matrices, les expressions symboliques, les objets-unités, les entiers-binaires, etc. Appliqué à deux chaînes de caractères, + renvoie une nouvelle chaîne créée en juxtaposant les deux chaînes données en argument (concaténation de chaînes). De même, + est capable de juxtaposer deux listes. Appliqué à deux objets graphiques (GROB) de mêmes tailles, + est équivalent à un *ou* logique (voir OR).

-
Soustraction

Accès : [-]

Drapeaux : -3

| | |
|----|--------|
| 2: | réel_1 |
| 1: | réel_2 |



| | |
|----|------|
| 2: | |
| 1: | réel |



- effectue la soustraction $\text{réel}_1 - \text{réel}_2$. - peut aussi être utilisé avec les complexes, les matrices, les expressions symboliques, les objets-unités, les entiers-binaires, etc.

/
Quotient

Accès : [÷]

Drapeaux : -3

| | |
|----|--------|
| 2: | réel_1 |
| 1: | réel_2 |



| | |
|----|------|
| 2: | |
| 1: | réel |



/ effectue la division $\text{réel}_1 \div \text{réel}_2$. ÷ peut aussi être utilisé avec les complexes, les matrices, les expressions symboliques, les objets-unités, les entiers-binaires, etc.

=
Création d'une équation

Accès : [\leftarrow] [0] Drapeaux : -3

| | | |
|------------------------------------|---|----------------------------------|
| 2: réel_1 1: réel_2 | → | 2: 1: 'réel_1=réel_2' |
| 2: 'expr_symb1' 1: 'expr_symb2' | → | 2: 1: 'expr_symb1=expr_symb2' |

= réunit deux objets en une équation. L'objet de niveau 2 devient le membre de gauche de l'équation alors que l'objet de niveau 1 devient le membre de droite de l'équation. = fonctionne avec les réels, les objets-unités, les complexes et les expressions symboliques.

==
Test de l'égalité de deux objets

Accès (HP-48 G/GX) : [PRG] (TEST) (= =) Drapeaux : -3

Accès (HP-48 S/SX) : [PRG] (TEST) [NXT] (= =)

| | | |
|--------------------------|---|------------------|
| 2: objet_1 1: objet_2 | → | 2: 1: égalité |
|--------------------------|---|------------------|

== compare deux objets et renvoie 1 si les deux objets sont identiques ou 0 s'ils ne le sont pas.

^
Exponentiation

Accès : [y^x] Drapeaux : -3, -1

| | | |
|------------------------|---|---------------|
| 2: réel_1 1: réel_2 | → | 2: 1: réel |
|------------------------|---|---------------|

^ élève réel_1 à la puissance réel_2. ^ peut aussi être utilisé avec les complexes, les expressions symboliques, les objets-unités ou les entiers-binaires.



!
Exponentielle

Accès (HP-48 G/GX) : [MTH] [NXT] (PROB) (!) Drapeaux : -3, -21, -20

Accès (HP-48 S/SX) : [MTH] (PROB) (!) →

| | | |
|-----------|---|-----------|
| 1: entier | → | 1: entier |
|-----------|---|-----------|

→ ! calcule la factorielle de son argument entier. Si l'argument est un réel x , ! renvoie $\Gamma(x+1)$.

%

Pourcentage

Accès (HP-48 G/GX) : [MTH] (REAL) (%) Drapeaux : -3

Accès (HP-48 S/SX) : [MTH] (PARTS) [NXT] (%)

| | | | | |
|----|--------|---|----|------|
| 2: | réel_1 | → | 2: | réel |
| 1: | réel_2 | | 1: | |



% renvoie en pourcentage la proportion que représente réel_1 par rapport à réel_2.

< > ≤ ≥

Test de supériorité et d'infériorité strictes ou larges

Accès (HP-48 G/GX) : [PRG] (TEST) ... Drapeaux : -3

Accès (HP-48 G/GX) : [PRG] (TEST) [NXT]...

| | | | | |
|----|---------|---|----|---------|
| 2: | objet_1 | → | 2: | égalité |
| 1: | objet_2 | | 1: | |

<, ≥, > et ≤ comparent deux objets de types compatibles (réels, entiers binaires, chaînes de caractères, objets-unités, expressions symboliques) et renvoie 1 si l'infériorité (ou la supériorité, selon le cas) de l'un des objets par rapport à l'autre est effective. Si ce n'est pas le cas, 0 est renvoyé.

**—
Création d'un objet-unité**

Accès : [⌈] [x] Drapeaux : -

| | | | | |
|----|---------|---|----|-------------|
| 2: | réel | → | 2: | objet-unité |
| 1: | 'unité' | | 1: | |

_ associe un réel (placé au niveau 2 de la pile) à une unité présentée sous la forme d'une expression symbolique et placée au niveau 1 de la pile. _ crée ainsi un objet-unité.

Par exemple, 5 'm' _ crée l'objet-unité '5_m'. On peut obtenir le même résultat avec 5 '1_m' _ ou même avec 5 1_m _.

\neq
Test de l'inégalité de deux objets

Accès (HP-48 G/GX) : [PRG] (TEST) (\neq) Drapeaux : -3

Accès (HP-48 S/SX) : [PRG] (TEST) [NXT] (\neq)

| | | | | |
|----|---------|---|----|-----------|
| 2: | objet_1 | → | 2: | inégalité |
| 1: | objet_2 | | 1: | |

\neq compare deux objets et renvoie 1 si les deux objets sont inégaux ou 0 s'ils ne le sont pas.

π
Constante π

Accès : [\leftarrow] [SPC] Drapeaux : -2, -3

| | | | | |
|----|--|---|----|-----|
| 1: | | → | 1: | `π` |
| 1: | | → | 1: | π |

π renvoie l'expression symbolique `π` ou la valeur 3,14159... de π en fonction des *flags* -2 et -3.

$\sqrt{\quad}$
Racine carrée

Accès : [\sqrt{x}] Drapeaux : -1, -3

| | | | | |
|----|------|---|----|------|
| 1: | réel | → | 1: | réel |
|----|------|---|----|------|

$\sqrt{\quad}$ calcule la racine carrée de son argument qui peut être réel (éventuellement, un objet-unité ou une expression symbolique) ou complexe.



Drapeaux, indicateurs ou flags

1

Notion d'indicateur

■ Qu'est-ce qu'un indicateur ?



Un indicateur, aussi appelé *flag* ou drapeau, est un marqueur spécialisé dont l'état (armé ou désarmé) indique l'existence ou la non existence d'une condition particulière. Plus simplement, un indicateur peut être considéré comme une lampe témoin pouvant être allumée ou éteinte : si la lampe est allumée la caractéristique associée à l'indicateur est disponible, si la lampe est éteinte, cette caractéristique est indisponible. Par exemple, l'indicateur -3 correspond à l'activation de l'évaluation numérique. Ainsi, lorsque l'indicateur -3 est armé, les expressions symboliques sont évaluées et perdent leur caractère symbolique pour devenir des nombres.



Attention ! Il est question dans le livre *HP-48 : faites vos jeux en assembleur* (DUNOD) d'indicateurs accessibles en assembleur. Ces indicateurs n'ont rien de commun avec ceux que nous manipulons en RPL.

La HP-48 dispose de 128 indicateurs :

- 64 indicateurs-système (*flags système*),
- 64 indicateurs-utilisateur (*flags utilisateur*).

Les 64 indicateurs-utilisateur sont à la disposition du programmeur qui peut les utiliser comme bon lui semble afin de contrôler l'exécution de ses

programmes. Les 64 indicateurs-utilisateur sont, par défaut, désarmés. Les 64 indicateurs-système peuvent être consultés et modifiés par l'utilisateur mais leurs rôles ne peuvent pas être redéfinis.



Un indicateur peut être armé (actif) ou désarmé (inactif). Ces deux états peuvent être représentés par un booléen (*TRUE/FALSE*) ou par un bit. L'usage veut que l'état d'un indicateur soit représenté par un bit :

- un bit valant 0 représente un indicateur désarmé,
- un bit valant 1 représente un indicateur armé.

Chaque indicateur pouvant être représenté par un bit, les 64 indicateurs-système peuvent être représentés par un mot binaire de 64 bits dans lequel chaque bit représente un indicateur. De la même manière, les 64 indicateurs-utilisateur peuvent être représentés par un mot binaire de 64 bits. Un nombre hexadécimal (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E ou F) représente quatre bits. Il est donc très commode de remplacer chaque mot de 64 bits par un mot de 16 nombres hexadécimaux.

Les 64 indicateurs-système et les 64 indicateurs-utilisateur ne sont pas numérotés de la même façon :

- les indicateurs-système sont numérotés de -1 à -64,
- les indicateurs-utilisateur sont numérotés de 1 à 64.



Nous allons voir qu'il est très facile de récupérer dans la pile le mot de 64 bits représentant les indicateurs-système ou le mot de 64 bits représentant les indicateurs-utilisateur. Placés dans la pile sous forme d'entiers binaires, ces mots binaires peuvent être utilisés avec dans des opérations logiques : AND, OR, XOR... Il est ensuite possible de modifier les états des indicateurs à partir d'entiers binaires placés dans la pile.

■ Récupérer les drapeaux et leur affecter des valeurs

La fonction *RCLF* renvoie dans la pile une liste de deux entiers binaires comportant 64 bit. Le premier entier binaire représente les indicateurs-système, alors que le second représente les indicateurs-utilisateur. Ainsi, on obtient par exemple :

```
{ #600000FF0h #0h }
```

Ici, #600000FF0h représente les indicateurs-système. Notez que les bits de poids forts nuls ne sont pas affichés (les 0 de gauche sont omis). #0h représente les indicateurs-utilisateur. Ici, #0h signale que tous les indicateurs utilisateurs sont désarmés (ils sont tous représentés par 0). Bien

entendu, les entiers binaires sont représentés à l'aide de la base active (voir BIN, OCT, DEC et HEX). Ils peuvent donc apparaître sous la forme `#nb` (binaire), `#no` (octal), `#nd` (décimal) ou `#nh` (hexadécimal). Une fois récupérés dans la pile grâce à `RCLF`, les indicateurs peuvent être utilisés dans des opérations logiques et arithmétiques.



Très important ! Le bit de poids faible d'un entier binaire représentant les états des indicateurs correspond à l'indicateur -1 ou 1. Les indicateurs suivants sont représentés par les bits de poids plus forts. Par exemple, si `RCLF` renvoie `{ #1h #1h }`, cela signifie que l'indicateur -1 (système) et que l'indicateur 1 (utilisateur) sont armés (actifs).

La fonction `STOF` (*store flag*) permet de modifier les états des indicateurs en fondant cette modification sur le ou les entiers binaires donnés en arguments. `STOF` peut prendre deux types d'arguments :

- un entier binaire représentant les états des indicateurs système,
- une liste contenant deux entiers binaires. Le premier représente les états des indicateurs système alors que le second représente les indicateurs utilisateur.

Par exemple `#5h STOF` arme (active) les indicateurs système -1 et -3. En effet, `#5h = #101b` ; le 1 de droite (bit de plus faible poids) représente l'indicateur -1, le 0 représente l'indicateur -2 alors que le 1 de gauche représente l'indicateur -3. Si la taille du mot binaire courant (définie avec `STWS`) est inférieure à 64 bits, les indicateurs dont les états ne peuvent être précisés compte tenu de la longueur du mot binaire sont désarmés.

De même `{ #0h #4h } STOF` désarme tous les indicateurs système (`#0h`) et arme l'indicateur utilisateur 3 (`#4h = #100b`, le troisième indicateur est donc armé).

■ Armer et désarmer un indicateur isolé



Les fonctions `RCLF` et `STOF` imposent la manipulation de l'ensemble des indicateurs et compliquent l'armement ou le désarmement d'un indicateur isolé. En effet, avec `RCLF` et `STOF` la manipulation d'un indicateur isolé impose l'emploi des opérations logiques AND, OR, XOR... Par exemple, armons l'indicateur -2 :

- à l'aide de `RCLF`, on peut récupérer l'entier binaire de 64 bits codant les 64 indicateurs système. Admettons pour l'exemple que l'entier binaire obtenu vaille `#6FAB000h`,
- l'opération logique `#6FAB000h #10b OR` renvoie `#6FAB002h`, l'analyse bit par bit de cet entier binaire montre

que le second bit (compté à partir du bit de plus faible poids) vaut 1, l'indicateur -2 sera donc armé,

- #6FAB002h STOF modifie ensuite les indicateurs système.



Les fonctions SF et CF permettent d'armer (SF) ou de désarmer (CF) un indicateur isolé. SF et CF prennent un réel (de -64 à -1 et de 1 à 64) en argument.

Par exemple, -40 SF arme l'indicateur -40. Armé, cet indicateur provoque l'affichage permanent de l'horloge dans la zone d'état de l'écran. De la même façon, on peut armer un indicateur utilisateur. Par exemple 1 SF arme l'indicateur utilisateur 1.

Pour exprimer les dates au format américain (mois/jour/année), il convient de désarmer l'indicateur système -42. Pour cela il suffit d'utiliser -42 CF.

■ Tester l'état d'un indicateur



Les fonctions FS? et FC? testent l'état d'un indicateur et renvoient 0 ou 1 selon l'état de l'indicateur dont le numéro est donné en argument. FS? teste l'armement alors que FC? teste le désarmement.

FS? renvoie :

- 0 si l'indicateur est désarmé,
- 1 si l'indicateur est armé.

FC? renvoie :

- 0 si l'indicateur est armé,
- 1 si l'indicateur est désarmé.

Par exemple, 1 FS? renvoie 1 si l'indicateur utilisateur 1 est armé. On peut tout aussi simplement tester le désarmement des indicateurs. Ainsi, -31 FC? renvoie 1 si l'indicateur système -31 est désarmé (le remplissage des représentations graphiques est alors actif).



FC?C et FS?C combinent les effets de FC? ou de FS? avec ceux de CF et prennent en argument un numéro d'indicateur. Ainsi, FS?C renvoie :

- 0 si l'indicateur est désarmé,
- 1 si l'indicateur est armé puis désarme l'indicateur testé.

De même, FC?C renvoie :

- 0 si l'indicateur est armé puis désarme l'indicateur testé,
- 1 si l'indicateur est désarmé.

2

Les indicateurs-système

Les 64 indicateurs système sont représentés par leurs numéros compris entre -1 et -64. Certains indicateurs système n'ont pas de rôle prédéterminé, ce sont les indicateurs :

- -4, -13, -14, -27, -28, -29, -54 sur HP-48 S/SX,
- -4, -13 et -30 sur HP-48 G/GX.

■ Le calcul symbolique : les indicateurs -1 à -3

Indicateur -1, solution principale ou ensemble des solutions possibles :

- armé, les instructions ISOL et QUAD n'affichent que la solution principale,
- désarmé, les instructions ISOL et QUAD fournissent l'ensemble des solutions.

Indicateur -2, évaluation des constantes symboliques :

- armé, les constantes MINR, MAXR, π , e et i s'affichent sous leurs formes numériques, et ce, quel que soit l'état de l'indicateur -3,
- désarmé, ces constantes s'affichent sous leurs formes symboliques à moins que l'indicateur -3 soit armé.



Indicateur -3, évaluation des arguments symboliques :

- armé, les constantes exprimées sous forme symbolique et les arguments symboliques sont, dans la mesure du possible, remplacés par leurs valeurs numériques,
- désarmé, les arguments symboliques gardent leur caractère symbolique et ne sont pas évalués sous forme numérique.

■ Les entiers binaires : les indicateurs -5 à -12



Les six indicateurs -5 à -10 déterminent la taille, en nombre de bits, des entiers binaires manipulés par l'utilisateur (en interne, la HP-48 travaille toujours sur 64 bits). La manipulation directe de ces 6 indicateurs n'est pas utile. En effet, RCWS (*recall binary word size*) renvoie la taille du mot binaire courant alors que STWS (*store binary word size*) permet de déterminer la taille du mot binaire.



Les indicateurs **-11** et **-12** déterminent la base de numération utilisée pour la représentation des entiers binaires dans la pile. L'utilisateur peut choisir entre la base 2 (binaire), la base 8 (octal), la base 10 (décimal) et la base 16 (hexadécimal). Les instructions BIN, OCT, DEC et HEX permettent de fixer la base de numération utilisée pour la représentation des entiers binaires, la manipulation directe des indicateurs **-11** à **-12** est donc inutile.

| Base | Indicateur -11 | Indicateur -12 |
|----------------------|----------------|----------------|
| Base 2 (#nb) | désarmé | armé |
| Base 8 (#no) | armé | désarmé |
| Base 10 (#nd) | désarmé | désarmé |
| Base 16 (#nh) | armé | armé |

■ **Dates de paiement : l'indicateur -14**

Uniquement sur HP-48 G/GX, l'indicateur **-14** permet de choisir la date de paiement dans le cadre des calculs TVM :

- armé, les paiements sont effectués en début de période,
- désarmé, les paiements sont effectués en fin de période.

■ **Mesures d'angle et des coordonnées : les indicateurs -15 à -18**



Les indicateurs **-15** et **-16** déterminent le mode de représentation des coordonnées (rectangulaires, cylindriques, sphériques). RECT, CYLIN et SPHERE sont les instructions permettant d'activer un mode de représentation des coordonnées sans manipuler directement les indicateurs **-15** et **-16**.

| Coordonnées | Indicateur -15 | Indicateur -16 |
|------------------------------|----------------|----------------|
| Rectangulaires | désarmé | désarmé |
| Polaires cylindriques | désarmé | armé |
| Polaires sphériques | armé | armé |

| Unité | Indicateur -17 | Indicateur -18 |
|----------------------------|----------------|----------------|
| Degré | désarmé | désarmé |
| Radian (témoin RAD) | armé | désarmé |
| Grade (témoin GRAD) | désarmé | armé |



Les indicateurs **-17** et **-18** déterminent l'unité de mesure d'angle par défaut. C'est l'unité de mesure d'angle par défaut qui est utilisée lorsqu'aucun suffixe (objet unité) ne suit une mesure d'angle. L'unité de mesure d'angle par défaut permet de donner des réels en argument aux

fonctions trigonométriques et n'impose donc pas l'emploi d'objets-unité (*réel_unité*). Les instructions DEG, GRAD et RAD activent l'une des trois unités (degré, radian ou grade) sans qu'il soit nécessaire de manipuler directement les indicateurs -17 et -18.

■ Bascule mode vecteur/mode complexe : l'indicateur -19

L'indicateur -19 détermine la fonction de $\rightarrow V2$:

- -19 étant armé, $\rightarrow V2$ prend deux réels aux deux premiers niveaux de la pile et renvoie un nombre complexe,
- -19 étant désarmé, $\rightarrow V2$ prend deux réels dans la pile et renvoie un vecteur du plan.

■ Dépassement de capacité : les indicateurs -20 à -26



En cas de dépassement des capacités de calcul de la HP-48, les indicateurs -20, -21 et -22 permettent de choisir entre le déclenchement d'une erreur et la poursuite des calculs. Si vous choisissez la poursuite des calculs, les indicateurs -23, -24, -25 et -26 garderont une trace de la nature du dépassement de capacité.

L'indicateur -20 traite les dépassements de capacité inférieure, autrement dit, les cas où les données tendent vers 0^- ou 0^+ :

- armé, le dépassement de capacité provoque une erreur,
- désarmé, le dépassement de capacité provoque le renvoi de 0 dans la pile et l'armement de l'indicateur -23 en cas de dépassement de capacité inférieure négative (0^-) ou l'armement de l'indicateur -24 en cas de dépassement de capacité inférieure positive (0^+).

L'indicateur -21 traite les dépassements de capacité rencontrés lorsque les valeurs manipulées provoquent le calcul de résultats inférieurs à MINR ou supérieurs à MAXR :

- armé, le dépassement de capacité provoque une erreur,
- désarmé, le dépassement de capacité renvoie la valeur numérique de MINR ou de MAXR (selon le cas) et arme l'indicateur -25.

L'indicateur -22 traite les cas où les résultats tendent vers $+\infty$ ou $-\infty$:

- armé, le dépassement de capacité renvoie la valeur numérique de MINR ou de MAXR (selon le cas) et arme l'indicateur -26,
- désarmé, le dépassement de capacité provoque une erreur.

**Rappel :**

- MINR correspond à $-9,9999999999 \times 10^{499}$,
- MAXR correspond à $9,9999999999 \times 10^{499}$.

■ Mise en forme des complexes : l'indicateur -27

L'indicateur **-27** permet, sur HP-48 G/GX uniquement, de mettre en forme les nombres complexes incorporés aux expressions symboliques :

- armé, les nombres complexes placés dans les expressions symboliques ne sont pas modifiés,
- désarmé, ' $a+b*i$ ' est automatiquement transformé en ' (a,b) '.

■ Tracés simultanés et représentation des axes du repère : les indicateurs -28 et -29

Lorsque plusieurs représentations graphiques sont enregistrées (contenues dans la variable réservée EQ), l'indicateur **-28** permet, uniquement sur HP-48 G/GX de choisir entre les tracés simultanés et les tracés séquentiels :

- armé, tout le contenu de EQ est représenté graphiquement simultanément,
- désarmé, les courbes sont tracées une par une les unes après les autres.



L'indicateur **-29** permet, sur HP-48 G/GX uniquement, d'activer ou de désactiver la représentation des axes du repère dans le cadre des tracés en deux dimensions et des représentations graphiques statistiques :

- armé, les axes ne sont pas tracés,
- désarmé, les axes sont tracés.

■ Représentation de $y=f(x)$: l'indicateur -30

Sur HP-48 S/SX uniquement, l'indicateur **-30** détermine le traitement de $y=f(x)$ lors d'une représentation graphique :

- armé, le tracé d'une équation du type $y=f(x)$ entraîne en fait le tracé de deux courbes différentes, l'une représentant y , l'autre représentant $f(x)$,
- désarmé, le tracé d'une équation du type $y=f(x)$ entraîne le tracé d'une courbe représentant $f(x)$.

■ Tracé des points intermédiaires et pointeur : les indicateurs -31 et -32

L'indicateur -31 permet d'opter pour le tracé de segments de droite entre les points calculés des représentations graphiques :



- armé, les points de la courbe dont les coordonnées sont calculées ne sont pas reliés entre eux à l'aide de segments de droite, d'où l'apparition de pointillés mais aussi la disparition d'asymptotes indésirables (lors du tracé de la fonction tangente par exemple),
- désarmé, les points calculés sont reliés entre eux.

L'indicateur -32 détermine l'aspect du pointeur graphique :

- armé, le pointeur graphique apparaît toujours en inverse vidéo et reste donc visible quel que soit le fond,
- désarmé, le pointeur graphique est noir quel que soit le fond.

■ Communication série : les indicateurs -33 à -39



L'indicateur -33 détermine le port (câble série ou diodes infrarouges) utilisé pour la communication avec un périphérique autre qu'une imprimante :

- armé, les informations transitent par le port infrarouge,
- désarmé, les informations passent par le câble.

L'indicateur -34 détermine le port (câble série ou diodes infrarouges) utilisé pour la communication avec une imprimante :

- armé, les informations passent par le câble,
- désarmé, les informations passent par le port infrarouge.

L'indicateur -35 détermine le format des données :

- armé, les informations sont transmises bit par bit (*binary*),
- désarmé, les bits d'information sont groupés sous la forme d'octets (*ASCII*).

L'indicateur -36 permet d'autoriser ou d'interdire l'écrasement des variables homonymes :

- armé, une variable reçue portant le même nom qu'une variable du répertoire courant écrase cette dernière,
- désarmé, en cas d'homonymie, la variable reçue voit un numéro ajouté à son nom.

L'indicateur -37 détermine la hauteur de l'interligne pour l'imprimante :

- armé, l'interligne est double,
- désarmé, l'interligne reste normal.

L'indicateur **-38** permet d'ajouter des sauts de ligne à la suite des lignes imprimées :

- armé, aucun saut de ligne n'est ajouté,
- désarmé, un saut de ligne est ajouté à la suite de chaque ligne imprimée.

L'indicateur **-39** permet d'interdire l'affichage des messages relatifs à la communication série :

- armé, les messages ne sont pas affichés,
- désarmé, les messages sont affichés.

■ **Date et heure : les indicateurs -40 à -44**

L'indicateur **-40** fixe les conditions d'affichage de l'horloge :

- armé, l'heure est affichée dans la zone d'état,
- désarmé, l'heure n'est affichée que pendant l'affichage du menu TIME.

L'indicateur **-41** fixe le format d'affichage de l'heure :

- armé, l'heure est affichée au format 24 heures,
- désarmé, l'heure est affichée au format 12 heures AM/PM.

L'indicateur **-42** fixe le format d'affichage de la date :

- armé, l'heure est affichée au format jour/mois/année,
- désarmé, l'heure est affichée au format mois/jour/année.

L'indicateur **-43** permet de répéter les alarmes échues non validées :

- armé, les alarmes échues non validées ne sont pas répétées,
- désarmé, les alarmes échues non validées sont répétées.

L'indicateur **-44** permet de mémoriser les alarmes échues et validées :

- armé, les alarmes échues et validées sont mémorisées,
- désarmé, les alarmes échues et validées sont supprimées.

■ **Formats des nombres et affichage : les indicateurs -45 à -54**

Les indicateurs **-45 à -48** codent le nombre de décimales. Ils sont contrôlés par les instructions FIX, SCI et ENG.

Les indicateurs **-49** et **-50** fixent, quant à eux, le mode d'affichage des nombres et permettent d'opter pour le mode scientifique, le mode ingénierie ou l'affichage avec un nombre fixe et déterminé de décimales.

| Format | Indicateur -49 | Indicateur -50 |
|--------------------------------|----------------|----------------|
| Normal (STD) | désarmé | désarmé |
| Nombre fixe de décimales (FIX) | armé | désarmé |
| Affichage scientifique (SCI) | désarmé | armé |
| Affichage ingénierie (ENG) | armé | armé |



L'indicateur **-51** détermine l'aspect du séparateur décimal :

- armé, une virgule sépare la partie entière de la partie décimale,
- désarmé, un point sépare la partie entière de la partie décimale.

L'indicateur **-52** détermine l'affichage du contenu du niveau 1 de la pile :

- armé, l'objet placé au niveau 1 de la pile est représenté sur une seule ligne quelle que soit sa longueur,
- désarmé, l'objet placé au niveau 1 de la pile peut, si besoin est, occuper jusqu'à quatre lignes de la pile.

L'indicateur **-53** permet d'autoriser la simplification des expressions algébriques :

- armé, toutes les parenthèses saisies apparaissent,
- désarmé, les parenthèses inutiles disparaissent.

Uniquement sur HP-48 G/GX, l'indicateur **-54** permet d'autoriser le remplacement des très petites valeurs par 0 :

- armé, les très petites valeurs ne sont pas remplacées et DET n'arrondit pas le résultat renvoyé,
- désarmé, les très petites valeurs sont remplacées par 0, DET arrondit le résultat renvoyé.

■ Mémorisation des derniers arguments : l'indicateur -55

L'indicateur **-55** permet d'activer ou de désactiver la mémorisation des derniers arguments utilisés :



- armé, les derniers arguments ne sont pas sauvegardés d'où un gain d'espace en mémoire,
- désarmé, les derniers arguments sont sauvegardés.

■ Les signaux sonores : les indicateurs -56 et -57

L'indicateur **-56** permet d'activer ou de désactiver le «bip» émis en cas d'erreur ou par l'instruction BEEP :

- armé, les signaux sonores sont désactivés,
- désarmé, les signaux sonores sont actifs.

L'indicateur -57 permet d'activer ou de désactiver le «bip» émis par le déclenchement des alarmes :

- armé, les signaux sonores sont désactivés,
- désarmé, les signaux sonores sont actifs.

■ Affichage des messages : les indicateurs -58 et -59

L'indication -58 détermine le mode d'affichage des messages émanant du système :

- armé, l'affichage automatique des messages-système est désactivé,
- désarmé, les messages-système sont affichés normalement.

L'indication -59 détermine, sur HP-48 S/SX, le mode d'affichage des équations, messages et menus (PLOT et PLOTR, SOLVE et SOLVR) :

- armé, seuls les noms des équations sont affichés,
- désarmé, les équations et leurs noms sont affichés.

L'indication -59 détermine, sur HP-48 G/GX, le mode d'affichage des variables dans le cadre du gestionnaire de variables :

- armé, seuls les noms des variables sont affichés,
- désarmé, chaque nom de variable est suivi du début du contenu de ladite variable.



Armer l'indicateur -59 fait perdre une certaine convivialité mais accélère le fonctionnement des environnements concernés.

■ Blocage du mode α et mode utilisateur : les indicateurs -60 et -61

L'indicateur -60 détermine le nombre de pressions sur $[\alpha]$ entraînant le verrouillage du mode alphabétique :

- armé, une seule pression sur $[\alpha]$ bloque le clavier en mode alphabétique,
- désarmé, une unique pression sur $[\alpha]$ active le mode alphabétique pour la prochaine pression sur le clavier, deux pressions sur $[\alpha]$ bloquent le clavier en mode alphabétique,

L'indicateur -61 contrôle l'accès aux modes USER et USR1 :

- armé, le mode USR1 est indisponible et $[\leftarrow]$ $[\alpha]$ donne accès au mode USER,
- désarmé, $[\leftarrow]$ $[\alpha]$ donne accès au mode USR1 alors que $[\leftarrow]$ $[\alpha]$ $[\leftarrow]$ $[\alpha]$ donne accès au mode USER.

L'indicateur **-62** active ou désactive le mode USER :

- armé, le mode USER est activé,
- désarmé, le mode USER est désactivé.

■ **Affectation de la touche ENTER : l'indicateur -63**

L'indicateur **-63** détermine le rôle de la touche [ENTER] :

- armé, la touche [ENTER] déclenche l'action définie par l'utilisateur,
- désarmé, la touche [ENTER] fonctionne normalement et provoque l'évaluation du contenu de la ligne de commande.

■ **Retour au premier élément une fois la fin de l'élément composite atteinte : l'indicateur -64**



L'indicateur **-64** autorise ou interdit le retour au premier élément après l'utilisation du dernier élément d'un objet composite (liste, tableau, etc.) par GETI ou PUTI :

- armé, le retour au premier élément est automatique une fois atteinte la fin de l'objet composite,
- désarmé, GETI ou PUTI ne reviennent pas automatiquement au premier élément d'un objet composite une fois le dernier élément utilisé.

Variables réservées

1

Variables réservées

?

Une simple pression sur [VAR] provoque l'affichage des variables du répertoire courant dans la barre des menus. L'utilisateur est susceptible de placer un objet de son choix dans une variable qu'il est libre de nommer à sa guise. Cependant, certains noms de variables correspondent, aux yeux de la HP-48, à des variables particulières dont le contenu est exploité par certaines applications intégrées ou fonctions intégrées (communication, graphisme, statistiques, etc.).

D'une façon générale, les variables réservées contiennent les paramètres dont un environnement (graphisme, communication, etc.) a besoin. Les variables réservées n'ont cependant aucun rapport avec les *flags* qui, eux aussi, paramètrent le fonctionnement de la calculatrice.



Les variables réservées de la HP-48 : ALRMDAT, CST, der (...), EQ, EXPR, IOPAR, MHpar, Mpar, Nmines, nn, PPAR, PRTPAR, sn, Σ DAT, Σ PAR, VPAR et ZPAR. Les variables portant ces noms ont un rôle précis et ne peuvent contenir qu'un objet d'un type et d'une nature déterminés. Notez que les noms des fonctions intégrées (SIN, CRDIR...) peuvent aussi être considérés comme des noms réservés. Par défaut, l'emploi d'une fonction (ou d'un environnement) faisant appel à une variable réservée provoque l'emploi de ladite variable réservée présente dans le répertoire courant. Si cette variable réservée n'existe pas, elle est généralement créée, et ce, dans le répertoire courant et avec des paramètres par défaut.

Seules les variables réservées `ALRMDAT`, `IOPAR` et `PRTPAR` ne peuvent pas se trouver dans un répertoire quelconque et n'existe qu'en un seul exemplaire (dans le répertoire `HOME`, sauf pour `ALRMDAT`). Les autres variables réservées, par exemple `ΣDAT` ou `EQ`, peuvent être présentes avec des contenus différents dans chacun des répertoires de l'arborescence.



La description précise de toutes les formes possibles de toutes les variables réservées nous semble peu utile dans la mesure où la HP-48 est capable de générer automatiquement toutes ces variables. Une fois créée, l'une de ces variables peut être incorporée à un programme si celui-ci a besoin de ladite variable. Toutefois, le lecteur pourra consulter l'*Advanced user's reference manual (Hewlett-Packard)* ou le *Manuel de référence avancé (POLE)* s'il souhaite disposer de plus d'informations à propos des variables réservées.

2

Structure des variables réservées



Mis à part quelques cas particuliers, les variables réservées peuvent contenir un nom d'une variable dont le contenu sera considéré comme étant celui de la variable réservée. Ainsi, `EQ`, `ΣDAT` ou `EXPR` peuvent contenir de simples noms de variables.

■ **ALRMDAT**

`ALRMDAT` est une variable très particulière puisqu'elle n'existe dans aucun répertoire. `ALRMDAT` est chargée de conserver toutes les données relatives aux alarmes programmées. Compte tenu de sa nature très particulière, `ALRMDAT` ne peut pas être éditée directement. Les instructions `STOALARM` (*store alarm*) et `RCLALARM` (*recall alarm*) permettent de travailler à partir de `ALRMDAT` depuis un programme RPL. Pour chaque alarme programmée, `ALRMDAT` conserve :

- la date et l'heure,
- une chaîne de caractères à afficher ou un objet à évaluer au moment du déclenchement de l'alarme,
- le délai d'attente avant la répétition de l'alarme en cas de non validation de celle-ci.

■ CST

La variable réservée `CST` (*custom*, personnalisation) correspond à la liste des objets disponibles *via* le menu `CST`. `CST` contient donc la liste des objets accessibles à l'aide des articles du menu affiché à l'aide de `[CST]`. La variable réservée `CST` correspond à une liste contenant les noms des articles (ce sont des chaînes de caractères) tous suivis par le nom de l'objet dont l'évaluation sera déclenchée par la sélection de l'article.

■ der(...)

Les noms de variables débutant par *der* correspondent à des variables créées par ∂ lors du calcul d'une dérivée à partir d'une fonction dont la fonction dérivée n'est pas présente en mémoire morte. *der* est suivi par le nom de la fonction dont la dérivée est calculée.

■ EQ

`EQ` contient généralement une expression algébrique mais peut aussi contenir une liste d'expressions algébriques, voire, d'autres types d'objets (un réel, un programme, un nom de variable...). Le contenu de `EQ`, modifiable avec `STEQ`, définit la représentation graphique pour les modes :

- `CONIC` (sections coniques),
- `DIFFEQ` (équations différentielles),
- `FUNCTION` (fonctions cartésiennes, c'est-à-dire du type $y=f(x)$),
- `PARAMETRIC` (équations paramétriques),
- `POLAR` (fonctions polaires),
- `TRUTH` (fonction booléenne de deux variables réelles),



Lorsque l'un des modes graphiques cités est actif, l'exécution de `DRAW` provoque la représentation graphique du contenu de la variable `EQ` appartenant au répertoire courant. Si `EQ` contient une expression algébrique ou une équation (les listes d'expressions algébriques ne sont pas tolérées dans ce second cas), `ROOT` permet la recherche de racines à partir du contenu de `EQ`.

■ EXPR

`EXPR` contient une expression algébrique servant de base de travail aux outils associés à l'application `SYMBOLIC`.

■ IOPAR

IOPAR est une liste de réels (qui sont tous des entiers naturels) déterminant les paramètres employés pour la communication par la voie série. IOPAR contient :

- la vitesse de transmission en bits par seconde,
- le mode de contrôle de parité,
- le paramètre XON/XOFF,
- le type de somme de contrôle,
- le mode de conversion des caractères.



IOPAR est créé ou mis à jour automatiquement lors d'une opération de communication. IOPAR ne peut demeurer que dans HOME.

■ MHPAR

Sur HP-48 G/GX, MHPAR (*MineHunt parameter*) conserve l'état d'une partie de démineur interrompue. MHPAR n'est pas créée automatiquement (il est nécessaire de presser la touche [STO] après avoir quitter le jeu).

■ MPAR

MPAR conserve les données requises par le solveur d'équations multiples. MINIT récupère les équations présentes dans la pile et crée ainsi MPAR. MPAR est une variable réservée considérée comme un ensemble de données de bibliothèque (*library data*). MPAR ne peut donc être consultée ou modifiée sans faire appel à des instructions dédiées.

■ N_{mines}

Cette variable correspond à un entier compris entre 1 et 64. Cet entier définit, sur HP-48 G/GX, le nombre de mines composant le champ de mises du démineur (lancé avec MINEHUNT).

■ n_n

Les variables n_1, n_2, n_3, \dots sont des entiers que créent ISOL et QUAD afin de représenter des résultats renvoyés par des calculs symboliques. Le *flag* système -1 provoque le renvoi dans la pile de la seule solution principale, les entiers arbitraires sont alors nuls.

■ PPAR

Comme EQ, une variable PPAR peut être présente dans chaque répertoire. C'est donc la variable PPAR du répertoire courant qui est utilisée par défaut (PPAR est créée dans le répertoire courant si elle n'existe pas). Utilisée par DRAW et modifiée par AUTO, PPAR définit entre autres les intervalles de représentation des axes. PPAR contient une liste :

- un nombre complexe définit, en coordonnées-utilisateur, les coordonnées de l'angle inférieur gauche de l'écran,
- de même, un nombre complexe définit, en coordonnées-utilisateur, les coordonnées de l'angle supérieur droit de l'écran,
- le nom de la variable indépendante est précisé,
- l'intervalle entre deux valeurs calculées de la variable indépendante est codé sous la forme d'un entier binaire,
- un nombre complexe définit, en coordonnées-utilisateur, les coordonnées du point d'intersection des deux axes du repère (par défaut, (0, 0)),
- l'intervalle entre deux graduations sur l'axe des abscisses est exprimé au moyen d'un réel ou d'un entier binaire,
- de même, l'intervalle entre deux graduations sur l'axe des ordonnées est exprimé au moyen d'un réel ou d'un entier binaire,
- le nom de l'axe des abscisses est précisé sous la forme d'une chaîne (par exemple, "X"),
- de même, le nom de l'axe des ordonnées est précisé sous la forme d'une chaîne (par exemple, "Y"),
- le type de tracé est précisé : FUNCTION, CONIQ, HISTOGRAM, GRIDMAP, POLAR, etc.
- le nom de la variable indépendante est précisé. Il peut être remplacé par une liste contenant des informations spécifiques.



PPAR définit les intervalles de représentation des axes à l'aide des coordonnées des extrémités de la seconde diagonale de l'écran. Par défaut, les abscisses sont représentées sur $[-^{13}_2, ^{13}_2]$ alors que les ordonnées le sont sur $[-^{31}_{10}, ^{16}_5]$.

■ PRTPAR

PRTPAR est une liste contenant les différents paramètres d'impression : délai d'attente entre deux lignes imprimées, chaîne définissant le mappage des caractères à partir du code ASCII #128d, nombre de caractères par ligne pour une imprimante reliée par câble, caractère de fin de ligne (CR ou CR/LF). PRTPAR ne peut exister que dans HOME.

■ s_n

Les variables réservées s_1, s_2, s_3, \dots correspondent à des signes arbitraires générés par ISOL ou QUAD lorsque la recherche de la solution principale est désactivée.

■ ΣDAT

ΣDAT est une matrice contenant les données exploitées par les fonctions statistiques. Il est possible de préciser le numéro de la colonne de ΣDAT contenant les valeurs de la variable indépendante et celui de celle contenant les valeurs de la variable dépendante (voir ΣPAR). Le nombre de colonnes de ΣDAT n'est pas limité. Il peut exister une matrice ΣDAT dans chaque répertoire (ces matrices pouvant bien sûr être différentes les unes des autres). Voir les instructions $\text{STO}\Sigma, \text{CL}\Sigma$.

■ ΣPAR

ΣPAR est une liste fixant les paramètres associés à la matrice ΣDAT du même répertoire. ΣPAR contient : le numéro de la colonne de ΣDAT contenant les valeurs de la variable indépendante, le numéro de la colonne de ΣDAT contenant les valeurs de la variable dépendante, le coefficient d'intersection, la pente de la droite de régression et le mode de régression.

■ VPAR

VPAR (*volumic parameter*) est une liste contenant les paramètres associés à la visualisation des tracés 3D : largeur de vue volumique, profondeur de vue volumique, hauteur de vue volumique, position de l'observateur, intervalles pour les abscisses et les ordonnées, largeur et profondeur de zone.

■ ZPAR

ZPAR (*zoom parameter*) est une liste contenant les paramètres de zoom exploités par toutes les représentations graphiques en deux dimensions. Cette liste comprend : le facteur d'agrandissement horizontal, le facteur d'agrandissement vertical, le recentrage automatique ou non autour du pointeur graphique et, éventuellement, la copie de PPAR .

Codes clavier



Pour chaque combinaison de touches voici les codes à utiliser avec des instructions telles que ASN, DELKEYS, STOKEYS et RCLKEYS dans le cadre de la personnalisation du clavier (évaluation d'un objet donné associée à la saisie d'une séquence de touches).

| Combinaison de touches | Code | Combinaison de touches | Code |
|------------------------|------|------------------------|------|
| [A] | 11.0 | [←] [C] | 13.2 |
| [A] | 11.1 | [→] [C] | 13.3 |
| [←] [A] | 11.2 | [α] [C] | 13.4 |
| [→] [A] | 11.3 | [α] [←] [C] | 13.5 |
| [α] [A] | 11.4 | [α] [→] [C] | 13.6 |
| [α] [←] [A] | 11.5 | [D] | 14.0 |
| [α] [→] [A] | 11.6 | [D] | 14.1 |
| [B] | 12.0 | [←] [D] | 14.2 |
| [B] | 12.1 | [→] [D] | 14.3 |
| [←] [B] | 12.2 | [α] [D] | 14.4 |
| [→] [B] | 12.3 | [α] [←] [D] | 14.5 |
| [α] [B] | 12.4 | [α] [→] [D] | 14.6 |
| [α] [←] [B] | 12.5 | [E] | 15.0 |
| [α] [→] [B] | 12.6 | [E] | 15.1 |
| [C] | 13.0 | [←] [E] | 15.2 |
| [C] | 13.1 | [→] [E] | 15.3 |

| Combinaison de touches | Code | Combinaison de touches | Code |
|--------------------------------------|------|---|------|
| [α] [E] | 15.4 | [VAR] | 24.1 |
| [α] [\leftarrow] [E] | 15.5 | [\leftarrow] [VAR] | 24.2 |
| [α] [\rightarrow] [E] | 15.6 | [\rightarrow] [VAR] | 24.3 |
| [F] | 16.0 | [α] [VAR] | 24.4 |
| [F] | 16.1 | [α] [\leftarrow] [VAR] | 24.5 |
| [\leftarrow] [F] | 16.2 | [α] [\rightarrow] [VAR] | 24.6 |
| [\rightarrow] [F] | 16.3 | [\uparrow] | 25.0 |
| [α] [F] | 16.4 | [\uparrow] | 25.1 |
| [α] [\leftarrow] [F] | 16.5 | [\leftarrow] [\uparrow] | 25.2 |
| [α] [\rightarrow] [F] | 16.6 | [\rightarrow] [\uparrow] | 25.3 |
| [MTH] | 21.0 | [α] [\uparrow] | 25.4 |
| [MTH] | 21.1 | [α] [\leftarrow] [\uparrow] | 25.5 |
| [\leftarrow] [MTH] | 21.2 | [α] [\rightarrow] [\uparrow] | 25.6 |
| [\rightarrow] [MTH] | 21.3 | [NXT] | 26.0 |
| [α] [MTH] | 21.4 | [NXT] | 26.1 |
| [α] [\leftarrow] [MTH] | 21.5 | [\leftarrow] [NXT] | 26.2 |
| [α] [\rightarrow] [MTH] | 21.6 | [\rightarrow] [NXT] | 26.3 |
| [PRG] | 22.0 | [α] [NXT] | 26.4 |
| [PRG] | 22.1 | [α] [\leftarrow] [NXT] | 26.5 |
| [\leftarrow] [PRG] | 22.2 | [α] [\rightarrow] [NXT] | 26.6 |
| [\rightarrow] [PRG] | 22.3 | ['] | 31.0 |
| [α] [PRG] | 22.4 | ['] | 31.1 |
| [α] [\leftarrow] [PRG] | 22.5 | [\leftarrow] ['] | 31.2 |
| [α] [\rightarrow] [PRG] | 22.6 | [\rightarrow] ['] | 31.3 |
| [CST] | 23.0 | [α] ['] | 31.4 |
| [CST] | 23.1 | [α] [\leftarrow] ['] | 31.5 |
| [\leftarrow] [CST] | 23.2 | [α] [\rightarrow] ['] | 31.6 |
| [\rightarrow] [CST] | 23.3 | [STO] | 32.0 |
| [α] [CST] | 23.4 | [STO] | 32.1 |
| [α] [\leftarrow] [CST] | 23.5 | [\leftarrow] [STO] | 32.2 |
| [α] [\rightarrow] [CST] | 23.6 | [\rightarrow] [STO] | 32.3 |
| [VAR] | 24.0 | [α] [STO] | 32.4 |

| Combinaison de touches | Code | Combinaison de touches | Code |
|--|------|---|------|
| [α] [\leftarrow] [STO] | 32.5 | [\leftarrow] [SIN] | 41.2 |
| [α] [\rightarrow] [STO] | 32.6 | [\rightarrow] [SIN] | 41.3 |
| [EVAL] | 33.0 | [α] [SIN] | 41.4 |
| [EVAL] | 33.1 | [α] [\leftarrow] [SIN] | 41.5 |
| [\leftarrow] [EVAL] | 33.2 | [α] [\rightarrow] [SIN] | 41.6 |
| [\rightarrow] [EVAL] | 33.3 | [COS] | 42.0 |
| [α] [EVAL] | 33.4 | [COS] | 42.1 |
| [α] [\leftarrow] [EVAL] | 33.5 | [\leftarrow] [COS] | 42.2 |
| [α] [\rightarrow] [EVAL] | 33.6 | [\rightarrow] [COS] | 42.3 |
| [\leftarrow] | 34.0 | [α] [COS] | 42.4 |
| [\leftarrow] | 34.1 | [α] [\leftarrow] [COS] | 42.5 |
| [\leftarrow] [\leftarrow] | 34.2 | [α] [\rightarrow] [COS] | 42.6 |
| [\rightarrow] [\leftarrow] | 34.3 | [TAN] | 43.0 |
| [α] [\leftarrow] | 34.4 | [TAN] | 43.1 |
| [α] [\leftarrow] [\leftarrow] | 34.5 | [\leftarrow] [TAN] | 43.2 |
| [α] [\rightarrow] [\leftarrow] | 34.6 | [\rightarrow] [TAN] | 43.3 |
| [\downarrow] | 35.0 | [α] [TAN] | 43.4 |
| [\downarrow] | 35.1 | [α] [\leftarrow] [TAN] | 43.5 |
| [\leftarrow] [\downarrow] | 35.2 | [α] [\rightarrow] [TAN] | 43.6 |
| [\rightarrow] [\downarrow] | 35.3 | [\sqrt{x}] | 44.0 |
| [α] [\downarrow] | 35.4 | [\sqrt{x}] | 44.1 |
| [α] [\leftarrow] [\downarrow] | 35.5 | [\leftarrow] [\sqrt{x}] | 44.2 |
| [α] [\rightarrow] [\downarrow] | 35.6 | [\rightarrow] [\sqrt{x}] | 44.3 |
| [\rightarrow] | 36.0 | [α] [\sqrt{x}] | 44.4 |
| [\rightarrow] | 36.1 | [α] [\leftarrow] [\sqrt{x}] | 44.5 |
| [\leftarrow] [\rightarrow] | 36.2 | [α] [\rightarrow] [\sqrt{x}] | 44.6 |
| [\rightarrow] [\rightarrow] | 36.3 | [y^x] | 45.0 |
| [α] [\rightarrow] | 36.4 | [y^x] | 45.1 |
| [α] [\leftarrow] [\rightarrow] | 36.5 | [\leftarrow] [y^x] | 45.2 |
| [α] [\rightarrow] [\rightarrow] | 36.6 | [\rightarrow] [y^x] | 45.3 |
| [SIN] | 41.0 | [α] [y^x] | 45.4 |
| [SIN] | 41.1 | [α] [\leftarrow] [y^x] | 45.5 |

| Combinaison de touches | Code | Combinaison de touches | Code |
|--|------|---|------|
| [α] [\rightarrow] [y^x] | 45.6 | [\rightarrow] [DEL] | 54.3 |
| [$1/x$] | 46.0 | [α] [DEL] | 54.4 |
| [$1/x$] | 46.1 | [α] [\leftarrow] [DEL] | 54.5 |
| [\leftarrow] [$1/x$] | 46.2 | [α] [\rightarrow] [DEL] | 54.6 |
| [\rightarrow] [$1/x$] | 46.3 | [\Leftarrow] | 55.0 |
| [α] [$1/x$] | 46.4 | [\Leftarrow] | 55.1 |
| [α] [\leftarrow] [$1/x$] | 46.5 | [\leftarrow] [\Leftarrow] | 55.2 |
| [α] [\rightarrow] [$1/x$] | 46.6 | [\rightarrow] [\Leftarrow] | 55.3 |
| [ENTER] | 51.0 | [α] [\Leftarrow] | 55.4 |
| [ENTER] | 51.1 | [α] [\leftarrow] [\Leftarrow] | 55.5 |
| [\leftarrow] [ENTER] | 51.2 | [α] [\rightarrow] [\Leftarrow] | 55.6 |
| [\rightarrow] [ENTER] | 51.3 | [α] | 61.0 |
| [α] [ENTER] | 51.4 | [α] | 61.1 |
| [α] [\leftarrow] [ENTER] | 51.5 | [\leftarrow] [α] | 61.2 |
| [α] [\rightarrow] [ENTER] | 51.6 | [\rightarrow] [α] | 61.3 |
| [+/.] | 52.0 | [α] [α] | 61.4 |
| [+/.] | 52.1 | [α] [\leftarrow] [α] | 61.5 |
| [\leftarrow] [+/.] | 52.2 | [α] [\rightarrow] [α] | 61.6 |
| [\rightarrow] [+/.] | 52.3 | [7] | 62.0 |
| [α] [+/.] | 52.4 | [7] | 62.1 |
| [α] [\leftarrow] [+/.] | 52.5 | [\leftarrow] [7] | 62.2 |
| [α] [\rightarrow] [+/.] | 52.6 | [\rightarrow] [7] | 62.3 |
| [EEX] | 53.0 | [α] [7] | 62.4 |
| [EEX] | 53.1 | [α] [\leftarrow] [7] | 62.5 |
| [\leftarrow] [EEX] | 53.2 | [α] [\rightarrow] [7] | 62.6 |
| [\rightarrow] [EEX] | 53.3 | [8] | 63.0 |
| [α] [EEX] | 53.4 | [8] | 63.1 |
| [α] [\leftarrow] [EEX] | 53.5 | [\leftarrow] [8] | 63.2 |
| [α] [\rightarrow] [EEX] | 53.6 | [\rightarrow] [8] | 63.3 |
| [DEL] | 54.0 | [α] [8] | 63.4 |
| [DEL] | 54.1 | [α] [\leftarrow] [8] | 63.5 |
| [\leftarrow] [DEL] | 54.2 | [α] [\rightarrow] [8] | 63.6 |

| Combinaison de touches | Code | Combinaison de touches | Code |
|------------------------|------|------------------------|------|
| [9] | 64.0 | [α] [5] | 73.4 |
| [9] | 64.1 | [α] [←] [5] | 73.5 |
| [←] [9] | 64.2 | [α] [→] [5] | 73.6 |
| [→] [9] | 64.3 | [6] | 74.0 |
| [α] [9] | 64.4 | [6] | 74.1 |
| [α] [←] [9] | 64.5 | [←] [6] | 74.2 |
| [α] [→] [9] | 64.6 | [→] [6] | 74.3 |
| [÷] | 65.0 | [α] [6] | 74.4 |
| [÷] | 65.1 | [α] [←] [6] | 74.5 |
| [←] [÷] | 65.2 | [α] [→] [6] | 74.6 |
| [→] [÷] | 65.3 | [×] | 75.0 |
| [α] [÷] | 65.4 | [×] | 75.1 |
| [α] [←] [÷] | 65.5 | [←] [×] | 75.2 |
| [α] [→] [÷] | 65.6 | [→] [×] | 75.3 |
| [←] | 71.0 | [α] [×] | 75.4 |
| [←] | 71.1 | [α] [←] [×] | 75.5 |
| [←] [←] | 71.2 | [α] [→] [×] | 75.6 |
| [→] [←] | 71.3 | [→] | 81.0 |
| [α] [←] | 71.4 | [→] | 81.1 |
| [α] [←] [←] | 71.5 | [←] [→] | 81.2 |
| [α] [→] [←] | 71.6 | [→] [→] | 81.3 |
| [4] | 72.0 | [α] [→] | 81.4 |
| [4] | 72.1 | [α] [←] [→] | 81.5 |
| [←] [4] | 72.2 | [α] [→] [→] | 81.6 |
| [→] [4] | 72.3 | [1] | 82.0 |
| [α] [4] | 72.4 | [1] | 82.1 |
| [α] [←] [4] | 72.5 | [←] [1] | 82.2 |
| [α] [→] [4] | 72.6 | [→] [1] | 82.3 |
| [5] | 73.0 | [α] [1] | 82.4 |
| [5] | 73.1 | [α] [←] [1] | 82.5 |
| [←] [5] | 73.2 | [α] [→] [1] | 82.6 |
| [→] [5] | 73.3 | [2] | 83.0 |

| Combinaison de touches | Code | Combinaison de touches | Code |
|------------------------|------|------------------------|------|
| [2] | 83.1 | [0] | 92.1 |
| [←] [2] | 83.2 | [←] [0] | 92.2 |
| [→] [2] | 83.3 | [→] [0] | 92.3 |
| [α] [2] | 83.4 | [α] [0] | 92.4 |
| [α] [←] [2] | 83.5 | [α] [←] [0] | 92.5 |
| [α] [→] [2] | 83.6 | [α] [→] [0] | 92.6 |
| [3] | 84.0 | [.] | 93.0 |
| [3] | 84.1 | [.] | 93.1 |
| [←] [3] | 84.2 | [←] [.] | 93.2 |
| [→] [3] | 84.3 | [→] [.] | 93.3 |
| [α] [3] | 84.4 | [α] [.] | 93.4 |
| [α] [←] [3] | 84.5 | [α] [←] [.] | 93.5 |
| [α] [→] [3] | 84.6 | [α] [→] [.] | 93.6 |
| [-] | 85.0 | [SPC] | 94.0 |
| [-] | 85.1 | [SPC] | 94.1 |
| [←] [-] | 85.2 | [←] [SPC] | 94.2 |
| [→] [-] | 85.3 | [→] [SPC] | 94.3 |
| [α] [-] | 85.4 | [α] [SPC] | 94.4 |
| [α] [←] [-] | 85.5 | [α] [←] [SPC] | 94.5 |
| [α] [→] [-] | 85.6 | [α] [→] [SPC] | 94.6 |
| [ON] | 91.0 | [+] | 95.0 |
| [ON] | 91.1 | [+] | 95.1 |
| [←] [ON] | 91.2 | [←] [+] | 95.2 |
| [→] [ON] | 91.3 | [→] [+] | 95.3 |
| [α] [ON] | 91.4 | [α] [+] | 95.4 |
| [α] [←] [ON] | 91.5 | [α] [←] [+] | 95.5 |
| [α] [→] [ON] | 91.6 | [α] [→] [+] | 95.6 |
| [0] | 92.0 | | |

Codes menus



L'instruction **MENU** affiche un menu intégré défini par un réel. Si celui-ci vaut 0, **MENU** renvoie le dernier menu affiché. Les codes des menus des HP-48 G et GX sont donnés par le tableau ci-dessous. Les codes utilisables sur HP-48 S et SX sont fournis par Hewlett-Packard à la page 724 du manuel (annexe D). Les menus sont classés en fonction de la position des touches y donnant accès : de haut en bas et de gauche à droite.

| Combinaison de touches | Code |
|----------------------------|------|
| [MTH] | 3 |
| [MTH] [NXT] | 3.2 |
| [MTH] (VECTR) | 4 |
| [MTH] (VECTR) [NXT] | 4.2 |
| [MTH] (MATR) | 5 |
| [MTH] (MATR) [NXT] | 5.2 |
| [MTH] (MATR) (MAKE) | 6 |
| [MTH] (MATR) (MAKE) [NXT] | 6.2 |
| [MTH] (MATR) (NORM) | 7 |
| [MTH] (MATR) (NORM) [NXT] | 7.2 |
| [MTH] (MATR) (FACTR) | 8 |
| [MTH] (MATR) (FACTR) [NXT] | 8.2 |
| [MTH] (MATR) (COL) | 9 |
| [MTH] (MATR) (ROW) | 10 |

| Combinaison de touches | Code |
|----------------------------|------|
| [MTH] (MATR) (ROW) [NXT] | 10.2 |
| [MTH] (LIST) | 11 |
| [MTH] (HYP) | 12 |
| [MTH] (HYP) [NXT] | 12.2 |
| [MTH] (REAL) | 14 |
| [MTH] (REAL) [NXT] | 14.2 |
| [MTH] (REAL) [NXT] [NXT] | 14.3 |
| [MTH] (BASE) | 15 |
| [MTH] (BASE) [NXT] | 15.2 |
| [MTH] (BASE) [NXT] (LOGIC) | 16 |
| [MTH] (BASE) [NXT] (BIT) | 17 |
| [MTH] (BASE) [NXT] (BYTE) | 18 |
| [MTH] [NXT] (PROB) | 13 |
| [MTH] [NXT] (FFT) | 19 |

| Combinaison de touches | Code | Combinaison de touches | Code |
|--------------------------------|------|-----------------------------|------|
| [MTH] [NXT] (CMPL) | 20 | [PRG] (GROB) | 37 |
| [MTH] [NXT] (CMPL) [NXT] | 20.2 | [PRG] (GROB) [NXT] | 37.2 |
| [MTH] [NXT] (CONS) | 21 | [PRG] (PICT) | 38 |
| [MTH] [NXT] (CONS) [NXT] | 21.2 | [PRG] (PICT) [NXT] | 38.2 |
| [←] [PRG] | 62 | [PRG] [NXT] (IN) | 39 |
| [←] [PRG] [NXT] | 62.2 | [PRG] [NXT] (IN) [NXT] | 39.2 |
| [PRG] | 22 | [PRG] [NXT] (OUT) | 40 |
| [PRG] [NXT] | 22.2 | [PRG] [NXT] (OUT) [NXT] | 40.2 |
| [PRG] (BRCH) | 23 | [PRG] [NXT] (RUN) | 41 |
| [PRG] (BRCH) [NXT] | 23.2 | [PRG] [NXT] (RUN) [NXT] | 41.2 |
| [PRG] (BRCH) (IF) | 24 | [PRG] [NXT] (ERROR) | 61 |
| [PRG] (BRCH) (CASE) | 25 | [PRG] [NXT] (ERROR) (IFERR) | 60 |
| [PRG] (BRCH) (START) | 26 | [CST] | 1 |
| [PRG] (BRCH) (FOR) | 27 | [←] [CST] | 63 |
| [PRG] (BRCH) (DO) | 29 | [←] [CST] (FMT) | 64 |
| [PRG] (BRCH) (WHILE) | 31 | [←] [CST] (ANGL) | 65 |
| [PRG] (TEST) | 32 | [←] [CST] (FLAG) | 66 |
| [PRG] (TEST) [NXT] | 32.2 | [←] [CST] (FLAG) [NXT] | 66.2 |
| [PRG] (TEST) [NXT] [NXT] | 32.3 | [←] [CST] (KEYS) | 67 |
| [PRG] (TEST) [NXT] [NXT] [NXT] | 32.4 | [←] [CST] (MENU) | 68 |
| [PRG] (TYPE) | 33 | [←] [CST] (MISC) | 69 |
| [PRG] (TYPE) [NXT] | 33.2 | [←] [CST] (MISC) [NXT] | 69.2 |
| [PRG] (TYPE) [NXT] [NXT] | 33.3 | [VAR] | 2 |
| [PRG] (LIST) | 34 | [←] [VAR] | 70 |
| [PRG] (LIST) (ELEM) | 35 | [←] [VAR] [NXT] | 70.2 |
| [PRG] (LIST) (ELEM) [NXT] | 35.2 | [←] [VAR] (DIR) | 71 |
| [PRG] (LIST) (PROC) | 36 | [←] [VAR] (ARITH) | 72 |
| [PRG] (LIST) (PROC) [NXT] | 36.2 | [←] [↑] | 73 |

| Combinaison de touches | Code |
|------------------------------------|------|
| [←] [↑] [NXT] | 73.2 |
| [←] [+/-] | 28 |
| [←] [7] | 74 |
| [←] [7] (ROOT) | 75 |
| [←] [7] (ROOT) (SOLVR) | 30 |
| [←] [7] (DIFFE) | 76 |
| [←] [7] (POLY) | 77 |
| [←] [7] (SYS) | 78 |
| [←] [7] (TVM) | 79 |
| [←] [7] (TVM) (SOLVR) | 80 |
| [←] [8] | 81 |
| [←] [8] [NXT] | 81.2 |
| [←] [8] (PTYPE) | 82 |
| [←] [8] (PPAR) | 83 |
| [←] [8] (PPAR) [NXT] | 83.2 |
| [←] [8] (PPAR) [NXT] [NXT] | 83.3 |
| [←] [8] [NXT] (3D) | 84 |
| [←] [8] [NXT] (3D) (PTYPE) | 85 |
| [←] [8] [NXT] (3D) (VPAR) | 86 |
| [←] [8] [NXT] (STAT) | 87 |
| [←] [8] [NXT] (STAT) (PTYPE) | 88 |
| [←] [8] [NXT] (STAT) (DATA) | 91 |
| [←] [8] [NXT] (STAT) (ΣPAR) | 89 |
| [←] [8] [NXT] (STAT) (ΣPAR) (MODL) | 90 |
| [←] [8] [NXT] (FLAG) | 92 |
| [←] [9] | 93 |
| [←] [9] [NXT] | 93.2 |
| [←] [9] [NXT] [NXT] | 93.3 |

| Combinaison de touches | Code |
|----------------------------------|-------|
| [←] [4] | 94 |
| [←] [4] [NXT] | 94.2 |
| [←] [4] [NXT] [NXT] | 94.3 |
| [←] [4] (ALRM) | 95 |
| [←] [5] | 96 |
| [←] [5] (DATA) | 97 |
| [←] [5] (ΣPAR) | 98 |
| [←] [5] (ΣPAR) (MODL) | 99 |
| [←] [5] (1VAR) | 100 |
| [←] [5] (1VAR) [NXT] | 100.2 |
| [←] [5] (PLOT) | 101 |
| [←] [5] (FIT) | 102 |
| [←] [5] (FIT) [NXT] | 102.2 |
| [←] [5] (SUMS) | 103 |
| [←] [6] | 59 |
| [→] [6] | 42 |
| [→] [6] [NXT] | 42.2 |
| [→] [6] [NXT] [NXT] | 42.3 |
| [→] [6] (LENG) | 43 |
| [→] [6] (LENG) [NXT] | 43.2 |
| [→] [6] (LENG) [NXT] [NXT] | 43.3 |
| [→] [6] (LENG) [NXT] [NXT] [NXT] | 43.4 |
| [→] [6] (AREA) | 44 |
| [→] [6] (AREA) [NXT] | 44.2 |
| [→] [6] (VOL) | 45 |
| [→] [6] (VOL) [NXT] | 45.2 |
| [→] [6] (VOL) [NXT] [NXT] | 45.3 |
| [→] [6] (VOL) [NXT] [NXT] [NXT] | 45.4 |

| Combinaison de touches | Code |
|--|------|
| [↵] [6] (<i>TIME</i>) | 46 |
| [↵] [6] (<i>SPEED</i>) | 47 |
| [↵] [6] (<i>SPEED</i>) [NXT] | 47.2 |
| [↵] [6] (<i>MASS</i>) | 48 |
| [↵] [6] (<i>MASS</i>) [NXT] | 48.2 |
| [↵] [6] (<i>MASS</i>) [NXT] [NXT] | 48.3 |
| [↵] [6] [NXT] (<i>FORCE</i>) | 49 |
| [↵] [6] [NXT] (<i>ENRG</i>) | 50 |
| [↵] [6] [NXT] (<i>ENRG</i>) | 50.2 |
| [↵] [6] [NXT] (<i>POWR</i>) | 51 |
| [↵] [6] [NXT] (<i>PRESS</i>) | 52 |
| [↵] [6] [NXT] (<i>PRESS</i>) [NXT] | 52.2 |
| [↵] [6] [NXT] (<i>TEMP</i>) | 53 |
| [↵] [6] [NXT] (<i>ELEC</i>) | 54 |
| [↵] [6] [NXT] (<i>ELEC</i>) [NXT] | 54.2 |
| [↵] [6] [NXT] [NXT] (<i>ANGL</i>) | 55 |
| [↵] [6] [NXT] [NXT] (<i>LIGHT</i>) | 56 |
| [↵] [6] [NXT] [NXT] (<i>LIGHT</i>) [NXT] | 56.2 |
| [↵] [6] [NXT] [NXT] (<i>RAD</i>) | 57 |
| [↵] [6] [NXT] [NXT] (<i>RAD</i>) [NXT] | 57.2 |

| Combinaison de touches | Code |
|---|-------|
| [↵] [6] [NXT] [NXT] (<i>VISC</i>) | 58 |
| [↵] [1] | 104 |
| [↵] [1] [NXT] | 104.2 |
| [↵] [1] (<i>SRVR</i>) | 105 |
| [↵] [1] (<i>IOPAR</i>) | 106 |
| [↵] [1] (<i>IOPAR</i>) [NXT] | 106.2 |
| [↵] [1] (<i>PRINT</i>) | 107 |
| [↵] [1] (<i>PRINT</i>) [NXT] | 107.2 |
| [↵] [1] (<i>PRINT</i>) (<i>PRTPA</i>) | 108 |
| [↵] [1] [NXT] (<i>SERIA</i>) | 109 |
| [↵] [2] | 110 |
| [↵] [2] [NXT] | 110.2 |
| [↵] [2] (<i>PORTS</i>) | 111 |
| [↵] [2] (<i>PORTS</i>) (:0:) | 112 |
| [↵] [3] | 113 |
| [↵] [3] (<i>EQLIB</i>) | 114 |
| [↵] [3] (<i>COLIB</i>) | 115 |
| [↵] [3] (<i>MES</i>) | 116 |
| [↵] [3] (<i>UTIL</i>) | 117 |

Codes ASCII



Cette liste présente les caractères disponibles sur HP-48. Quand elle existe, la combinaison de touches donnant accès à un caractère est présentée. Quand ces combinaisons de touches n'existent pas, il suffit de donner le code ASCII du caractère en argument à CHR pour obtenir une chaîne contenant le caractère recherché. Les équivalences hexadécimales sont données à la suite de la liste des caractères.

| Car. | Combinaison de touches | Code | Car. | Combinaison de touches | Code |
|------|-----------------------------|------|------|------------------------------------|------|
| ... | <i>néant (utilisez CHR)</i> | 31 | + | [α] [+] | 43 |
| | [SPC] | 32 | , | [←] [.] | 44 |
| ! | [α] [←] [DEL] | 33 | - | [α] [-] | 45 |
| " | [α] [↵] [-] | 34 | . | [.] (avec <i>flag -51</i> désarmé) | 46 |
| # | [↵] [÷] | 35 | / | [α] [÷] | 47 |
| \$ | [α] [←] [4] | 36 | 0 | [0] | 48 |
| % | [α] [↵] [TAN] | 37 | 1 | [1] | 49 |
| & | [α] [←] [ENTER] | 38 | 2 | [2] | 50 |
| ' | [α] [↵] ['] | 39 | 3 | [3] | 51 |
| (| [←] [÷] | 40 | 4 | [4] | 52 |
|) | [←] [÷] | 41 | 5 | [5] | 53 |
| * | [α] [×] | 42 | 6 | [6] | 54 |

| Car. | Combinaison de touches | Code | Car. | Combinaison de touches | Code |
|------|------------------------------|------|------|-----------------------------|------|
| 7 | [7] | 55 | O | [α] [EVAL] | 79 |
| 8 | [8] | 56 | P | [α] [←] | 80 |
| 9 | [9] | 57 | Q | [α] [↓] | 81 |
| : | [α] [↵] [+] | 58 | R | [α] [→] | 82 |
| ; | [↵] [.] (avec flag -51 armé) | 59 | S | [α] [SIN] | 83 |
| < | [α] [↵] [2] | 60 | T | [α] [COS] | 84 |
| = | [α] [↵] [0] | 61 | U | [α] [TAN] | 85 |
| > | [α] [↵] [2] | 62 | V | [α] [√x] | 86 |
| ? | [α] [↵] [⇐] | 63 | W | [α] [y ^x] | 87 |
| @ | [α] [↵] [ENTER] | 64 | X | [α] [1/x] | 88 |
| A | [α] [A] | 65 | Y | [α] [+/-] | 89 |
| B | [α] [B] | 66 | Z | [α] [EEX] | 90 |
| C | [α] [C] | 67 | [| [↵] [x] | 91 |
| D | [α] [D] | 68 | \ | <i>néant (utilisez CHR)</i> | 92 |
| E | [α] [E] | 69 |] | [↵] [x] | 93 |
| F | [α] [F] | 70 | ^ | [y ^x] | 94 |
| G | [α] [MTH] | 71 | _ | [↵] [x] | 95 |
| H | [α] [PRG] | 72 | ` | <i>néant (utilisez CHR)</i> | 96 |
| I | [α] [CST] | 73 | a | [α] [↵] [A] | 97 |
| J | [α] [VAR] | 74 | b | [α] [↵] [B] | 98 |
| K | [α] [↑] | 75 | c | [α] [↵] [C] | 99 |
| L | [α] [NXT] | 76 | d | [α] [↵] [D] | 100 |
| M | [α] ['] | 77 | e | [α] [↵] [E] | 101 |
| N | [α] [STO] | 78 | f | [α] [↵] [F] | 102 |

| Car. | Combinaison de touches | Code | Car. | Combinaison de touches | Code |
|------|---|------|----------------|---|------|
| g | [α] [\leftarrow] [MTH] | 103 | ⊗ | <i>néant (utilisez CHR)</i> | 127 |
| h | [α] [\leftarrow] [PRG] | 104 | \angle | [\rightarrow] [SPC] | 128 |
| i | [α] [\leftarrow] [CST] | 105 | \bar{x} | [α] [\rightarrow] [$1/x$] | 129 |
| j | [α] [\leftarrow] [VAR] | 106 | ∇ | <i>néant (utilisez CHR)</i> | 130 |
| k | [α] [\leftarrow] [\uparrow] | 107 | $\sqrt{\quad}$ | [\sqrt{x}] | 131 |
| l | [α] [\leftarrow] [NXT] | 108 | \int | [\rightarrow] [COS] | 132 |
| m | [α] [\leftarrow] ['] | 109 | Σ | [\rightarrow] [TAN] | 133 |
| n | [α] [\leftarrow] [STO] | 110 | ♦ | <i>néant (utilisez CHR)</i> | 134 |
| o | [α] [\leftarrow] [EVAL] | 111 | π | [\leftarrow] [SPC] | 135 |
| p | [α] [\leftarrow] [\leftarrow] | 112 | ∂ | [\rightarrow] [SIN] | 136 |
| q | [α] [\leftarrow] [\downarrow] | 113 | \leq | [α] [\leftarrow] [3] | 137 |
| r | [α] [\leftarrow] [\rightarrow] | 114 | \geq | [α] [\rightarrow] [3] | 138 |
| s | [α] [\leftarrow] [SIN] | 115 | \neq | [α] [\rightarrow] [1] | 139 |
| t | [α] [\leftarrow] [COS] | 116 | α | [α] [\rightarrow] [A] | 140 |
| u | [α] [\leftarrow] [TAN] | 117 | \rightarrow | [\rightarrow] [0] | 141 |
| v | [α] [\leftarrow] [\sqrt{x}] | 118 | \leftarrow | [α] [\rightarrow] [\leftarrow] | 142 |
| w | [α] [\leftarrow] [y^x] | 119 | \downarrow | [α] [\rightarrow] [\downarrow] | 143 |
| x | [α] [\leftarrow] [$1/x$] | 120 | \uparrow | [α] [\rightarrow] [\uparrow] | 144 |
| y | [α] [\leftarrow] [$+/-$] | 121 | γ | [α] [\rightarrow] [MTH] | 145 |
| z | [α] [\leftarrow] [EEX] | 122 | δ | [α] [\rightarrow] [D] | 146 |
| { | [\leftarrow] [+] | 123 | ε | [α] [\rightarrow] [E] | 147 |
| | [α] [\rightarrow] [VAR] | 124 | η | [α] [\rightarrow] [PRG] | 148 |
| } | [\leftarrow] [+] | 125 | θ | [α] [\rightarrow] [F] | 149 |
| ~ | [α] [\rightarrow] [\sqrt{x}] | 126 | λ | [α] [\rightarrow] [NXT] | 150 |

| Car. | Combinaison de touches | Code | Car. | Combinaison de touches | Code |
|-----------|--|------|---------|---|------|
| φ | [α] [\rightarrow] [\rightarrow] | 151 | - | <i>néant (utilisez CHR)</i> | 175 |
| σ | [α] [\rightarrow] [SIN] | 152 | \circ | [α] [\rightarrow] [6] | 176 |
| τ | [α] [\rightarrow] [COS] | 153 | \pm | [α] [\rightarrow] [+/-] | 177 |
| ω | [α] [\rightarrow] [y^*] | 154 | 2 | <i>néant (utilisez CHR)</i> | 178 |
| Δ | [α] [\rightarrow] [C] | 155 | 3 | <i>néant (utilisez CHR)</i> | 179 |
| Π | [α] [\rightarrow] [+/-] | 156 | , | <i>néant (utilisez CHR)</i> | 180 |
| Ω | [α] [\rightarrow] [EVAL] | 157 | μ | [α] [\rightarrow] [STO] | 181 |
| ■ | <i>néant (utilisez CHR)</i> | 158 | η | <i>néant (utilisez CHR)</i> | 182 |
| ∞ | [α] [\rightarrow] [CST] | 159 | • | <i>néant (utilisez CHR)</i> | 183 |
| | <i>néant (utilisez CHR)</i> | 160 | , | <i>néant (utilisez CHR)</i> | 184 |
| i | [α] [\rightarrow] [DEL] | 161 | 1 | <i>néant (utilisez CHR)</i> | 185 |
| € | [α] [\rightarrow] [4] | 162 | 0 | <i>néant (utilisez CHR)</i> | 186 |
| £ | [α] [\leftarrow] [5] | 163 | » | [α] [\leftarrow] [-] | 187 |
| ¤ | [α] [\leftarrow] [6] | 164 | $1/4$ | <i>néant (utilisez CHR)</i> | 188 |
| ¥ | [α] [\rightarrow] [5] | 165 | $1/2$ | <i>néant (utilisez CHR)</i> | 189 |
| ‡ | <i>néant (utilisez CHR)</i> | 166 | $3/4$ | <i>néant (utilisez CHR)</i> | 190 |
| § | <i>néant (utilisez CHR)</i> | 167 | ¿ | [α] [\rightarrow] [\Leftarrow] | 191 |
| ¨ | <i>néant (utilisez CHR)</i> | 168 | À | [α] [A] [α] [\leftarrow] [7] | 192 |
| ⊕ | <i>néant (utilisez CHR)</i> | 169 | Á | [α] [A] [α] [\rightarrow] [7] | 193 |
| ð | <i>néant (utilisez CHR)</i> | 170 | Â | [α] [A] [α] [\leftarrow] [8] | 194 |
| « | [α] [\leftarrow] [-] | 171 | Ã | [α] [A] [α] [\rightarrow] [8] | 195 |
| ¬ | <i>néant (utilisez CHR)</i> | 172 | Ä | [α] [A] [α] [\leftarrow] [9] | 196 |
| - | <i>néant (utilisez CHR)</i> | 173 | Å | [α] [A] [α] [\rightarrow] [9] | 197 |
| ⊗ | <i>néant (utilisez CHR)</i> | 174 | Æ | [α] [E] [α] [\rightarrow] [9] | 198 |

| Car. | Combinaison de touches | Code | Car. | Combinaison de touches | Code |
|------|-----------------------------|------|------|----------------------------|------|
| Ç | [α] [C] [α] [↵] [9] | 199 | β | [α] [↵] [B] | 223 |
| È | [α] [E] [α] [↵] [7] | 200 | à | [α] [↵] [A] [α] [↵] [7] | 224 |
| É | [α] [E] [α] [↵] [7] | 201 | á | [α] [↵] [A] [α] [↵] [7] | 225 |
| Ê | [α] [E] [α] [↵] [8] | 202 | â | [α] [↵] [A] [α] [↵] [8] | 226 |
| Ë | [α] [E] [α] [↵] [9] | 203 | ã | [α] [↵] [A] [α] [↵] [8] | 227 |
| Ì | [α] [CST] [α] [↵] [7] | 204 | ä | [α] [↵] [A] [α] [↵] [9] | 228 |
| Í | [α] [CST] [α] [↵] [7] | 205 | å | [α] [↵] [A] [α] [↵] [9] | 229 |
| Î | [α] [CST] [α] [↵] [8] | 206 | æ | [α] [↵] [E] [α] [↵] [9] | 230 |
| Ï | [α] [CST] [α] [↵] [9] | 207 | ç | [α] [↵] [C] [α] [↵] [9] | 231 |
| Ð | [α] [D] [α] [↵] [9] | 208 | è | [α] [↵] [E] [α] [↵] [7] | 232 |
| Ñ | [α] [STO] [α] [↵] [8] | 209 | é | [α] [↵] [E] [α] [↵] [7] | 233 |
| Ò | [α] [EVAL] [α] [↵] [7] | 210 | ê | [α] [↵] [E] [α] [↵] [8] | 234 |
| Ó | [α] [EVAL] [α] [↵] [7] | 211 | ë | [α] [↵] [E] [α] [↵] [9] | 235 |
| Ô | [α] [EVAL] [α] [↵] [8] | 212 | ì | [α] [↵] [CST] [α] [↵] [7] | 236 |
| Õ | [α] [EVAL] [α] [↵] [8] | 213 | í | [α] [↵] [CST] [α] [↵] [7] | 237 |
| Ö | [α] [EVAL] [α] [↵] [9] | 214 | î | [α] [↵] [CST] [α] [↵] [8] | 238 |
| × | <i>néant (utilisez CHR)</i> | 215 | ï | [α] [↵] [CST] [α] [↵] [9] | 239 |
| Ø | [α] [EVAL] [α] [↵] [9] | 216 | đ | [α] [↵] [D] [α] [↵] [9] | 240 |
| Ù | [α] [TAN] [α] [↵] [7] | 217 | ñ | [α] [↵] [STO] [α] [↵] [8] | 241 |
| Ú | [α] [TAN] [α] [↵] [7] | 218 | ò | [α] [↵] [EVAL] [α] [↵] [7] | 242 |
| Û | [α] [TAN] [α] [↵] [8] | 219 | ó | [α] [↵] [EVAL] [α] [↵] [7] | 243 |
| Ü | [α] [TAN] [α] [↵] [9] | 220 | ô | [α] [↵] [EVAL] [α] [↵] [8] | 244 |
| Ý | [α] [+/.] [α] [↵] [7] | 221 | õ | [α] [↵] [EVAL] [α] [↵] [8] | 245 |
| ÿ | [α] [←] [α] [↵] [9] | 222 | ö | [α] [↵] [EVAL] [α] [↵] [9] | 246 |

Codes des messages



Cette liste rassemble plus de 110 messages (erreurs et informations) émis par le système. Ces messages sont classés selon leurs numéros. Ceux-ci sont des entiers binaires qui peuvent être donnés en argument à DOERR afin d'afficher le message correspondant.

| Numéro | Message | Domaine |
|--------|-----------------------|------------|
| #001h | Insufficient Memory | Mémoire |
| #002h | Directory Recursion | Répertoire |
| #003h | Undefined Local Name | Mémoire |
| #004h | Undefined XLIB Name | Mémoire |
| #005h | Memory Clear | Mémoire |
| #006h | Power Lost | Général |
| #008h | Invalid Card Data | Mémoire |
| #009h | Object In use | Général |
| #00Ah | Port Not available | Mémoire |
| #00Bh | No Room in Port | Mémoire |
| #00Ch | Object Not in Port | Mémoire |
| #101h | No Room to Save Stack | Mémoire |
| #102h | Can't Edit Null Char. | Général |
| #103h | Invalid User Function | Général |
| #104h | No Current Equation | Général |
| #106h | Invalid Syntax | Général |

| Numéro | Message | Domaine |
|---------------|----------------------------|----------------------|
| #124h | LAST STACK Disabled | Général |
| #125h | LAST CMD Disabled | Général |
| #126h | HALT Not Allowed | Général |
| #128h | Wrong Argument Count | Général |
| #129h | Circular Reference | Général |
| #12Ah | Directory Not Allowed | Répertoire |
| #12Bh | Non-Empty Directory | Répertoire |
| #12Ch | Invalid Definition | Général |
| #12Eh | Invalid PPAR | Général |
| #12Fh | Non-Real Result | Général |
| #130h | Unable to isolate | Général |
| #131h | No Room to show Stack | Général |
| #135h | Out of Memory | Mémoire |
| #13Ch | Name Conflict | Mémoire |
| #201h | Too Few Arguments | Arguments |
| #202h | Bad Argument Type | Arguments |
| #203h | Bad Argument Value | Arguments |
| #204h | Undefined Name | Mémoire |
| #205h | LASTARG disabled | Général |
| #206h | Incomplete Subexpression | EquationWriter |
| #207h | Implicit () off | EquationWriter |
| #208h | Implicit () on | EquationWriter |
| #301h | Positive Underflow | Dépassement |
| #302h | Negative Underflow | Dépassement |
| #303h | Overflow | Dépassement |
| #304h | undefined Result | Dépassement |
| #305h | Infinite result | Dépassement |
| #501h | Invalid Dimension | Tableau |
| #502h | Invalid Array Element | Tableau |
| #503h | Deleting Row | Tableau |
| #504h | Deleting Column | Tableau |
| #505h | Inserting Row | Tableau |
| #506h | Inserting Column | Tableau |
| #601h | Invalid Σ Data | Données statistiques |
| #602h | Nonexistent Σ DAT | Données statistiques |
| #603h | Insufficient Σ Data | Données statistiques |

| Numéro | Message | Domaine |
|--------|--|------------------------|
| #604h | Invalid Σ PAR | Données statistiques |
| #605h | Invalid Σ Data LN(Neg) | Données statistiques |
| #606h | Invalid Σ Data LN(0) | Données statistiques |
| #607h | Invalid EQ | Graphisme |
| #608h | Current equation: | Equation courante |
| #609h | No current equation. | Equation courante |
| #60Ah | Enter eqn, press NEW | Equation courante |
| #60Bh | Name the equation, press ENTER | Equation courante |
| #60Ch | Select plot type | Graphisme |
| #60Dh | Empty catalog | Alarme, équation, stat |
| #60Fh | No statistics data to plot | Données statistiques |
| #610h | Autoscaling | Graphisme |
| #614h | Select a model | Données statistiques |
| #619h | Acknowledged | Alarme |
| #61Ah | Enter alarm, press SET | Alarme |
| #61Bh | Select repeat interval | Alarme |
| #61Ch | I/O setup menu | Liaison série |
| #61Dh | Plot type: | Graphisme |
| #61Eh | "" | Général |
| #61Fh | (OFF SCREEN) | Graphisme |
| #620h | Invalid PTYPE | Graphisme |
| #621h | Name the stat data, press ENTER | Données statistiques |
| #622h | Enter value (zoom out if > 1), press ENTER | Graphisme |
| #623h | Copied to stack | Equation courante |
| #624h | x axis zoom w/auto | Graphisme |
| #625h | x axis zoom. | Graphisme |
| #626h | y axis zoom. | Graphisme |
| #627h | x and y-axis zoom. | Graphisme |
| #A01h | Bad Guess(es) | Equation courante |
| #A02h | Constant ? | Equation courante |
| #A03h | Interrupted | Equation courante |
| #A04h | Zero | Equation courante |
| #A05h | Sign reversal | Equation courante |
| #A06h | Extremum | Equation courante |
| #B01h | Invalid Unit | Unités |
| #B02h | Inconsistent Units | Unités |

| Numéro | Message | Domaine |
|---------------|------------------------|----------------|
| #C01h | Bad Packet Block check | Liaison série |
| #C02h | Timeout | Liaison série |
| #C03h | Receive Error | Liaison série |
| #C04h | Receive Buffer Overrun | Liaison série |
| #C05h | Parity Error | Liaison série |
| #C06h | Transfer Failed | Liaison série |
| #C07h | Protocol Error | Liaison série |
| #C08h | Invalid Server Cmd | Liaison série |
| #C09h | Port Closed | Liaison série |
| #C0Ah | Connecting | Liaison série |
| #C0Bh | Retry # | Liaison série |
| #C0Ch | Awaiting Server Cmd. | Liaison série |
| #C0Dh | Sending | Liaison série |
| #C0Eh | Receiving | Liaison série |
| #C0Fh | Object Discarded | Liaison série |
| #C10h | Packet # | Liaison série |
| #C11h | Processing Command | Liaison série |
| #C12h | Invalid IOPAR | Liaison série |
| #C13h | Invalid PRTPAR | Liaison série |
| #C14h | I/O: Batt Too Low | Liaison série |
| #C15h | Empty Stack | Arguments |
| #C17h | Invalid Name | Liaison série |
| #D01h | Invalid Date | Date et heure |
| #D02h | Invalid Time | Date et heure |
| #D03h | Invalid Repeat | Date et heure |
| #D04h | Nonexistent Alarm | Date et heure |

3

RPL-système



Si vous maîtrisez le RPL et avez bien compris les échanges de données *via* la pile ainsi que les structures de base du RPL, vous pouvez aborder le RPL-système. Le premier chapitre de cette partie vous expose le principe de la programmation en RPL-système, le second vous présente la structure des objets élémentaires, enfin, les autres chapitres sont des listes d'adresses (utilisables avec `SYSEVAL` pour la plupart) à utiliser dans vos programmes écrits en RPL-système.



Attention ! La programmation en RPL-système suppose la maîtrise des fonctions logiques, de la numération binaire et de la numération hexadécimale. Ces notions ainsi que la structure de la mémoire ont été décrites dans le livre *HP-48 faites vos jeux en assembleur* (DUNOD).

Programmer en RPL-système



Chaque instruction RPL peut être décomposée en sous-instructions appelées primitives de l'instruction. Chacune de ces primitives est en fait un mini programme présent en mémoire morte. La programmation en RPL système repose sur l'exécution de chacun de ces petits programmes. Chaque primitive correspondant à une adresse de programme en mémoire morte, un programme écrit en RPL-système en avant tout une succession d'adresses. Celles-ci s'étendent de #00000h à #FFFFFFh.

1

Créer un programme en RPL-système

■ Emploi de l'instruction SYSEVAL



L'instruction `SYSEVAL` évalue un objet dont l'adresse en mémoire est donnée en argument sous la forme d'un entier binaire. `SYSEVAL` permet d'évaluer des objets présents en mémoire morte depuis un programme RPL classique. `SYSEVAL` est donc le moyen le plus simple d'aborder la programmation en RPL-système qui repose sur le rappel d'adresses d'objets présents en mémoire morte.

Considérons par exemple le programme RPL :

```
« 360 .5 BEEP »
```

Ce programme extrêmement simple produit pendant 0,5 seconde un son dont la fréquence est 360 Hz.

Les trois objets **360**, **.5** et **BEEP** sont disponibles à travers la mémoire morte :

- **#65126h** est une adresse qui, lorsqu'elle est appelée, place le réel **360** dans la pile,
- de même, **#650BDh** place **.5** dans la pile,
- **#1A5C4h** correspond à l'adresse de l'instruction **BEEP**.

Nous avons donc les équivalences :



| | | |
|-------------|---|----------------|
| 360 | → | #65126h |
| .5 | → | #650BDh |
| BEEP | → | #1A5C4h |

Le programme « **360 .5 BEEP** » devient donc :

```

«
#65126h SYSEVAL
#650BDh SYSEVAL
#1A5C4h SYSEVAL
»
    
```

Ce programme place trois entiers binaires dans la pile. Chacun de ces entiers binaires est une adresse en mémoire morte. L'instruction SYSEVAL évalue l'objet dont l'adresse est donnée en argument.

- CRC : #7688h / 56.5

... ou, plus élégamment :

```

«
1 3
FOR i
{
#65126h
#650BDh
#1A5C4h
}
i
GET
SYSEVAL
NEXT
»
    
```

Le compteur i de la boucle FOR... NEXT varie entre 1 et 3.

Les trois adresses exprimées par des entiers binaires sont rassemblées dans une liste délimitée par { et }.

GET extrait le i-ème élément de la liste et le place dans la pile pour que SYSEVAL puisse le prendre en argument.

SYSEVAL évalue l'adresse extraite de la liste.

- CRC : #1063h / 78

Jusqu'ici, tous nos programmes sont de véritables programmes RPL classiques. Seule la présence de SYSEVAL nous permet de faire un premier pas vers le RPL système.



Attention ! SYSEVAL prend en argument une adresse codée sur 5 quarts (5 nombres hexadécimaux) mais aucune vérification n'est faite quant à la validité de cette adresse. L'emploi de certaines adresses peut avoir un effet dévastateur. Des pertes de mémoire sont à craindre si vous fournissez une adresse quelconque à SYSEVAL.

■ Création d'un objet programme RPL système



Attention ! L'emploi du RPL système peut, en cas d'erreur dans un programme, bloquer la machine ou effacer sa mémoire. Deux précautions élémentaires sont donc à prendre :

- sauvegardez tout le contenu de la mémoire (sur un micro-ordinateur *via* la liaison série, sur une autre HP-48 *via* le port infrarouge, etc.),
- souvenez-vous que la touche [ON] peut être désactivée, et donc, incapable d'interrompre un programme. Le seul moyen de débloquer la machine consiste à utiliser la combinaison de touches [ON]-[C].

Les listings des programmes RPL-système se présentent sous la forme de suites de chiffres hexadécimaux (0, 1, 2, 3, 4, 5, 6, 7, 8, 8, A, B, C, D, E et F) placés dans des chaînes de caractères. Assimilables à des chaînes de caractères et non à des programmes RPL, les listings RPL-système sont placés entre " et " et non entre « et ».

Ci-dessous, un exemple de listing de programme RPL-système :

```
"D9D20E163293632B2130"
```



La chaîne de caractères contenant le listing ne permet pas d'utiliser le programme. Pour obtenir un programme exécutable, il faut traiter la chaîne de caractères avec le programme ASS présenté ci-dessous :

```
<<
DUP SIZE
2 / .5 +
IP
"GROB 1 "
SWAP
+ " "
```

```
+ SWAP
+ OBJ→
#4017h SYSEVAL #56B6h SYSEVAL
DROP
NEWOB
»
```

- CRC : DA72h / 104.5



Stockez le programme ci-dessus sous le nom ASS. Nous utiliserons ce programme pour convertir les chaînes hexadécimales en programmes. Par exemple, introduisez la chaîne :

```
"D9D20B2130"
```

Validez-la avec [ENTER] puis exécutez ASS. Cette chaîne code un programme RPL minimal et se résume en fait aux délimiteurs d'un programme RPL. ASS renvoie donc dans la pile un programme vide (le niveau 1 semble vide mais ne l'est pas). Appuyez sur [EVAL] pour exécuter ce programme vide qui, vous vous en doutez, ne produit rien...

Regardons de plus près la chaîne de caractères **"D9D20B2130"**. Cette chaîne contient deux champs. Le premier correspond à 02D9D (prologue d'une structure RPL) et le second à 0321B (épilogue d'une structure RPL). Retenez que tous les champs doivent être inversés, ainsi :

- 02D9D devient D9D20,
- 0321B devient B2130.

Nous avons vu, au début de ce chapitre, le programme :

```
« #65126h SYSEVAL
#650BDh SYSEVAL
#1A5C4h SYSEVAL »
```

Inversons chacune des trois adresses qu'il contient :

- 65126 devient 62156,
- 650BD devient DB056,
- 1A5C4 devient 4C5A1.

Plaçons ces trois adresses inversées entre les délimiteurs inversés **D9D20** et **B2130** :

```
"D9D2062156DB0564C5A1B2130"
```

- CRC : #91E1h / 30

Cette chaîne étant au niveau 1 de la pile, exécutez ASS. ASS renvoie alors 360 .5 BEEP. Stockez ce programme dans la variable de votre choix puis évaluez cette variable. Le programme doit normalement s'exécuter et produire le même effet que le programme « 360 .5 BEEP ».

A retenir :



- le contenu de chaque champ (qu'il s'agisse ou non d'une adresse) doit être inversé,
- un listing d'un programme RPL-système est une chaîne de chiffres hexadécimaux délimitée par " et ",
- le programme ASS transforme cette chaîne en programme.



Bien que correspondant respectivement au prologue et à l'épilogue d'un programme, 02D9D et 0312B ne représentent pas les délimiteurs « et ». Ceux-ci sont représentés par 2361E et 23639. Invertissons ces adresses et insérons-les dans le programme que nous venons de réaliser :

"D9D20 E1632 62156 DB056 4C5A1 93632 B2130"

- CRC : 585Bh / 40



Attention ! Les espaces n'ont été ajoutés à la chaîne qu'afin d'accroître sa lisibilité. Ne les saisissez surtout pas car ils interdiraient l'interprétation par ASS de la chaîne hexadécimale.

La chaîne **"D9D20E163262156DB0564C5A193632B2130"** étant saisie, validée avec [ENTER] et placée au niveau 1, exécutez ASS. Vous obtenez le programme « 360 .5 BEEP » qu'il suffit de stocker dans une variable puis d'exécuter en évaluant ladite variable.

■ Création d'objets avec ASS

A partir de chaînes hexadécimales, ASS permet de créer tous les types d'objets. Les chaînes que nous abordons maintenant placent au niveau 1 de la pile un objet d'un type courant. Ces chaînes pourront être utilisées dans des programmes RPL-système afin d'insérer dans la pile les arguments demandés par une instruction ou une fonction.

Plaçons dans la pile le réel $3,14 \times 10^{-3}$. Le codage d'un réel fait intervenir quatre champs :

- le prologue 02933,
- l'exposant du facteur 10 (3 quartets DCB),
- la mantisse (12 quartets DCB),
- le signe (1 quartet).

Quand l'exposant du facteur 10 est négatif, on le remplace par 1000 moins l'exposant. Le quartet codant le signe vaut 0 si le nombre est positif, 9 s'il est positif. On obtient les quatre champs :

- 02933 (prologue signalant un réel),
- 997 (exposant -3, codé par 1000-3),
- 314000000000 (mantisse, la partie entière se limite toujours à un chiffre),
- 0 (signe, le nombre étant positif).

On inverse les quatre champs, on obtient :

- 33920,
- 799,
- 000000000413,
- 0.

Juxtaposons ces champs inversés dans une chaîne de caractères :

"33920799000000004130"

Traitée par ASS, cette chaîne place au niveau 1 le nombre réel $3,14 \times 10^{-3}$. Cette chaîne peut être incorporée à un programme RPL-système.

Introduisons maintenant dans la pile le caractère **a**. Un caractère est un objet très simple codé sur deux champs :

- le prologue 029BF,
- le code ASCII du caractère en hexadécimal sur deux quartets.

Le caractère **a** est donc codé par :

- 029BF,
- 61 (code ASCII de **a** en base 16).

On inverse les champs, on obtient :

- FB920,
- 16.

Juxtaposons ces champs, on obtient la chaîne :

"FB92016"

Traitée avec ASS, cette chaîne crée un objet de type caractère.

Créons maintenant la chaîne de caractères "HP". Une chaîne fait intervenir de nombreux champs :

- le prologue 02A2C,
- la longueur hors prologue exprimée en hexa sur 5 quartets,

- un champ de deux quartets pour chacun des caractères. Un champ correspond au code ASCII hexadécimal d'un caractère.

On obtient donc les champs :

- 02A2C,
- 00009 (longueur totale hors prologue),
- 48 (code ASCII hexadécimal de H),
- 50 (code ASCII hexadécimal de P).

Inversés, les champs deviennent :

- C2A20,
- 90000,
- 84,
- 05.

En juxtaposant les champs, on obtient :

"C2A20900008405"

Traitée par ASS, cette chaîne renvoie **"HP"** au niveau 1 de la pile.

2

Exemples



Les structures des programmes RPL-système sont exactement les mêmes que celles des programmes RPL classique, les instructions sont juste traduites. Cependant, un programme RPL-système peut contenir des objets aussi performants que des programmes assembleur.

■ Traiter les variables locales

Nous allons ici créer un programme équivalent à :

```
«
→ b c
« b c +»
»
```

Ce programme prend deux réels dans la pile et les ajoute l'un à l'autre. Une structure de variable locale fait intervenir le symbole → ainsi que « et » chargés de délimiter la portion de programme où les variables locales sont visibles :

- → introduisant des noms de variables locales se trouve à l'adresse #234C1h,
- les délimiteurs « et » de visibilité de variables locales se trouvent respectivement aux adresses #2361Eh et #235FEh.

Construisons la version RPL-système du programme RPL :

| <i>RPL</i> | <i>Champs</i> | <i>Champs inversés</i> |
|-----------------|--------------------|------------------------|
| | 02D9D | D9D20 |
| < | 2361E | E1632 |
| → | 234C1 | 1C432 |
| b | 02E6D 01 62 | D6E20 10 26 |
| c | 02E6D 01 63 | D6E20 10 36 |
| < | 2361E | E1632 |
| b | 02E6D 01 62 | D6E20 10 26 |
| c | 02E6D 01 63 | D6E20 10 36 |
| + | 1AB67 | 76BA1 |
| > | 235FE | EF532 |
| >> | 0312B | B2130 |

Ici **D6E20 10 26** code le nom local 'b'. Nous avons volontairement oublié 23639 (>>) avant B1230 (fin d'objet RPL)... mais le programme fonctionne quand même ! On juxtapose les champs inversés au sein d'une chaîne de caractères :

```
"D9D20E16321C432D6E201026D6E201036E1632D6E201026D6E20
103676BA1EF532B2130"
```

- CRC : #9F24h / 76

Placez cette chaîne dans la pile, exécutez ASS puis stockez le résultat (un programme) dans une variable. Placez ensuite deux réels dans la pile puis évaluez cette variable. Le programme renvoie la somme des deux réels.

■ **Calcul du PGCD de deux entiers**

Issu du livre "HP-48 : Maths en prépa" (DUNOD), ce programme calcule le PGCD de deux entiers (algorithme d'Euclide binaire). Saisissez la chaîne :

```
"D9D20F1AA19F345DBBF1F1AA19F345CCD208B00014713717915
371001351748FB9760147137179157797A17AFE97A96AE16B10"
```

12010182281C832606410B6510182281C832CD1111181088228
 1E8324F1189FA50AFEB7297A606EDF969C0A74A6D64FF159F8F
 2D760142164808CD5345B2130"

- CRC : 2757h / 234

Cette chaîne doit être traitée par ASS. Le résultat renvoyé par ASS est le programme calculant le PGCD. Placez deux entiers naturels dans la pile puis exécutez le programme. Celui-ci renvoie le PGCD (plus grand commun diviseur) des entiers donnés en arguments.

Ce programme doit son exceptionnelle rapidité à l'emploi d'une portion de programme écrite en assembleur. Commentons le listing :

| | |
|--------------|-----------------|
| 02D9D | <i>Prologue</i> |
| 1AA1F | ABS |
| 543F9 | R→B |
| 1FBBD | SWAP |
| 1AA1F | ABS |
| 543F9 | R→B |

Début de la partie assembleur

| | | |
|----------------|---|---|
| 02DCC | <i>Prologue code exécutable (programme en assembleur)</i> | |
| 000B8 | <i>Taille (184 quartets) de la partie assembleur</i> | |
| 147 | C=DAT1 A | <i>C1=adresse de l'objet 1</i> |
| 137 | CD1EX | <i>D1=adresse de l'objet 1</i> |
| 179 | D1=D1+10 | <i>Saute le prologue</i> |
| 1537 | A=DAT1 W | <i>A=entier 1</i> |
| 100 | R0=A | <i>R0=entier 1</i> |
| 135 | D1=C | <i>D1 pointe sur le niveau 1</i> |
| 174 | D1=D1+5 | <i>D1 pointe sur le niveau 2</i> |
| 8FB9760 | GOSBVL 0679B | <i>Sauve les registres D0, D1, B, D</i> |
| 147 | C=DAT1 A | <i>C=adresse de l'objet 2</i> |
| 137 | CD1EX | <i>D1=adresse de l'objet 2</i> |
| 179 | D1=D1+10 | <i>Saute le prologue</i> |
| 1577 | C=DAT1 W | <i>C=entier 2</i> |
| 97A17 | ?C=0 W | <i>entier 2 = 0 ?</i> |
| | GOYES @1 | <i>si oui alors PGCD = entier 1</i> |
| AFF | ACEX W | <i>A=entier 2, C=entier 1</i> |
| 97A96 | ?C=0 W | <i>entier 1 = 0 ?</i> |
| | GOYES @1 | <i>si oui alors pgcd = entier 2</i> |
| AE1 | B=0 B | <i>Initialiser B à 0</i> |

Si $p=2p'$ et $q=2q'$, on a $\text{PGCD}(p,q)=2*\text{PGCD}(p',q')$. B va servir à compter combien de fois on peut ainsi mettre 2 en facteur.

| | | | |
|--------------|--------------|-----------|--------------------------|
| 6B10 | GOTO | @2 | aller en @2 |
| | | | |
| | | @5 | |
| 120 | AR0EX | | A=entier 1 |
| 101 | R1=A | | R1=entier 1 |
| 822 | SB=0 | | vider SB |
| 81C | ASRB | | A=(entier 1)/2 |
| 83260 | ?SB=0 | | L'entier 2 est-il pair ? |
| | GOYES | @3 | Si oui aller en @3 |
| 6410 | GOTO | @4 | Si non aller en @4 |

@3

B65 **B=B+1** **B**

Si on arrive à @3, cela signifie qu'on a pu mettre 2 en facteur dans les deux entiers. On incrémente donc B et on continue à essayer de diviser par 2.

| | | | |
|--------------|--------------|-----------|--------------------------|
| | | @2 | |
| 101 | R1=A | | R1=entier 2 |
| 822 | SB=0 | | Vider SB |
| 81C | ASRB | | A=(entier 2)/2, SB=reste |
| 832CD | ?SB=0 | | L'entier 2 est-il pair ? |
| | GOYES | @5 | Si oui tester l'entier 1 |

@4

111 **A=R1**

118 **C=R0**

Quand on arrive ici, on sait que les entiers sont de la forme $2^n p$ et $2^n q$, avec p et q dans A et C, et n dans B. De plus p au moins est impair.

| | | | |
|--------------|--------------|-----------|--------------------|
| | | @6 | |
| 108 | R0=C | | R0 = q |
| 822 | SB=0 | | Vider SB |
| 81E | CSRB | | C=q/2 et SB=reste |
| 8324F | ?SB=0 | | q est-il pair ? |
| | GOYES | @8 | Si oui aller en @6 |

Si $q=2q'$, comme p est impair, on a $\text{pgcd}(p,q)=\text{pgcd}(p,q')$. Donc on peut diviser q par 2 tant que c'est possible.

118 **C=R0** C = q

A ce stade, p et q sont tous deux impairs.

| | | | |
|--------------|-----------------|-----------|---|
| 9FA50 | ?C>=A | W | <i>Est-ce que $q \geq p$?</i> |
| | GOYES | @7 | <i>Si oui, va en @7</i> |
| AFE | ACEX | W | <i>Sinon échange p et q</i> |

@7

Ici, $C = \max(p, q)$ et $A = \min(p, q)$.

| | | | |
|--------------|--------------|-----------|---|
| B72 | C=C-A | W | <i>$C = \text{ABS}(p-q)$</i> |
| 97A60 | ?C=0 | W | <i>Est-ce que $p=q$?</i> |
| | GOYES | @8 | <i>Si oui alors $\text{pgcd}(p, q) = p = q$ est dans A</i> |

Comme $\text{PGCD}(u, v) = \text{PGCD}(u-v, v)$, on remplace p par $\text{abs}(p-q)$ et q par $\min(p, q)$ sans changer le PGCD. Comme p et q étaient impairs, le nouveau p est pair et q est impair, donc on continue à diminuer p en le divisant par 2.

| | | |
|-------------|-------------|-----------|
| 6EDF | GOTO | @6 |
|-------------|-------------|-----------|

A chaque parcours de la boucle, on remplace p et q par des nombres positifs qui ont le même PGCD, et tels que $p+q$ diminue strictement. La boucle va donc nécessairement se terminer.

@8

A ce stade, on sait que le PGCD cherché est $2B \times A$.

| | | | |
|--------------|--------------|-----------|-------------------------------------|
| 969C0 | ?B=0 | B | <i>Si $B=0$, termine</i> |
| | GOYES | @1 | |
| A74 | A=A+A | W | <i>Sinon, multiplie A par 2</i> |
| A6D | B=B-1 | B | <i>et diminue B de 1</i> |
| 64FF | GOTO | @8 | |

@1

| | | | |
|----------------|---------------|--------------|---|
| 159D | DAT1=A | 16 | <i>sauve le PGCD à la place de l'entier 2</i> |
| 8F2D760 | GOSBVL | 067D2 | <i>Récupère D, B, D1 et D0</i> |
| 142 | A=DAT0 | A | <i>Fin du programme</i> |

Fin de la partie assembleur

| | |
|-------------|----------------|
| 164 | D0=D0+5 |
| 808C | PC=(A) |

@_n représente une adresse en mémoire. A, B, C, D... sont des registres de la HP-48. La deuxième colonne de la partie assembleur présente des mnémoniques assembleur (voir HP-48 : *Faites vos jeux en assembleur*, DUNOD)

15

Objets

1

Introduction



Aux yeux de la HP-48, toutes les entités stockées en mémoire sont des objets. Ainsi, un nombre réel ou un code exécutable (programme) est placé en mémoire sous forme d'objet. Il existe différents types de données (des nombres, des chaînes de caractères, des programmes, etc.), il y aura donc en mémoire plusieurs types d'objets. L'objet est donc une sorte de « capsule » dans laquelle on place une donnée à stocker en mémoire. L'objet va être rangé à une adresse en mémoire. Cette adresse correspond à celle du premier quartet de l'objet, les autres quartets de l'objet étant placés aux adresses suivantes (écriture séquentielle de l'objet en mémoire). Le prologue d'un objet est un nombre hexadécimal à 5 chiffres occupant les 5 premiers quartets de l'objet. Le prologue définit le type de l'objet. Notez qu'il existe 28 types d'objets reconnus par la HP-48 dont une partie seulement est utilisable *via* le RPL classique, les autres n'étant accessibles à l'utilisateur qu'au moyen du RPL système ou de l'assembleur.

Ce chapitre est une version très allégée du chapitre *Les objets en mémoire* du livre *HP-48 : Faites vos jeux en assembleur* (DUNOD) mais il suffit pour développer des applications intéressantes en RPL système.

Nous présenterons l'implantation en mémoire de chaque objet. Nous noterons « @ » l'adresse de l'objet qui correspond à l'adresse du premier quartet du prologue de l'objet considéré. Un objet pouvant être décomposé en champs, chaque champ aura une adresse notée sous la forme « @ + #nh » où #nh est un nombre hexadécimal ($\#nh = n_{16}$).



Prenez garde ! La HP-48 inverse le contenu de chaque champ d'un objet. Etudiez attentivement l'implantation générique de l'objet que nous présentons sous forme de tableau ainsi que les exemples proposés afin de bien comprendre le mécanisme d'inversion des quartets d'un objet.

Un objet doit être saisi sous la forme d'une chaîne de caractères ne contenant que des chiffres hexadécimaux. C'est à l'aide d'un programme tel que ASS que l'on convertit une chaîne de caractères en l'objet qu'elle code.



La commande TYPE renvoie un type pour chaque objet placé au niveau 1 de la pile. Le tableau ci-dessous présente chaque objet avec son prologue et son type.

| Nom de l'objet | Prologue | Type |
|--|--------------|-------|
| Entier système (<i>system binary</i>) | 02911 | 20 |
| Nombre réel (<i>real</i>) | 02933 | 0 |
| Nombre réel double précision (<i>long real</i>) | 02955 | 21 |
| Nombre complexe (<i>complex</i>) | 02977 | 1 |
| Nombre complexe double précision (<i>long complex</i>) | 0299D | 22 |
| Caractère (<i>character</i>) | 029BF | 24 |
| Tableau (<i>array</i>) | 029E8 | 3 / 4 |
| Tableau indexé (<i>linked array</i>) | 02A0A | 23 |
| Chaîne de caractères (<i>string</i>) | 02A2C | 2 |
| Entier binaire (<i>binary integer</i>) | 02A4E | 10 |
| Liste (<i>list</i>) | 02A74 | 5 |
| Répertoire (<i>directory</i>) | 02A96 | 15 |
| Expression algébrique (<i>algebraic</i>) | 02AB8 | 9 |
| Objet avec unité (<i>unit</i>) | 02ADA | 13 |
| Objet « taggué » (<i>tagged</i>) | 02AFC | 12 |
| Objet graphique (<i>graphic</i>) | 02B1E | 11 |
| Bibliothèque (<i>library</i>) | 02B40 | 16 |
| Sauvegarde (<i>backup</i>) | 02B62 | 17 |
| Données de bibliothèque (<i>library data</i>) | 02B88 | 26 |
| Pointeur étendu (<i>extended pointer</i>) | 02BAA | 27 |
| Objet réservé 1 (<i>reserved 1</i>) | 02BCC | 27 |
| Objet réservé 2 (<i>reserved 2</i>) | 02BEE | 27 |
| Objet réservé 3 (<i>reserved 3</i>) | 02C10 | 27 |
| Programme RPL (<i>program</i>) | 02D9D | 8 |
| Code exécutable (<i>code</i>) | 02DCC | 25 |
| Nom global (<i>global name</i>) | 02E48 | 6 |
| Nom local (<i>local name</i>) | 02E6D | 7 |
| Accès à une bibliothèque (<i>XLIB name</i>) | 02E92 | 14 |

**Notations utilisées :**

- @ est l'adresse de l'objet (adresse de son premier quartet),
- #nh est un nombre hexadécimal (en base 16).

Les tailles de chaque champ sont exprimées en nombre de quartets. Le quartet étant, rappelons-le, l'unité d'information au sein de la HP-48.

Pour chaque objet un tableau standardisé récapitule ce que vous devez savoir de l'objet étudié :



| | | |
|--|------------------------------------|------------------------|
| Nom de l'objet (<i>nom de l'objet dans la documentation américaine</i>) | | |
| Prologue : 02911 | Taille totale : 10 quartets | |
| Type : 20 | Nombre de champs : 2 | |
| Classe : donnée | Prologue : pas de prologue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02911 | @ | 5 quartets |
| Codage du nombre | @ + #5h | 5 quartets |
| <i>Fin de l'objet</i> | @ + #Ah | |

Les rubriques du tableau contiennent les informations suivantes :

- le nom de l'objet suivi de l'appellation américaine placée entre parenthèses,
- le prologue constitué de 5 chiffres hexadécimaux,
- la taille totale de l'objet exprimée en quartets,
- le type renvoyé par la fonction TYPE,
- le nombre de champs qui constituent l'objet,
- la classe de l'objet,
- l'éventuel épilogue marquant la fin de l'objet,
- un tableau d'implantation en mémoire donnant l'adresse et la taille de chaque champ (@ étant l'adresse du prologue et de l'objet).



Attention ! La taille totale annoncée pour les objets de longueur variable est la plus petite taille possible de l'objet considéré.



Nous n'abordons ici qu'une dizaine de types d'objets choisis parmi les plus simples et les plus utiles au programmeur. Une liste exhaustive des types d'objets est proposée dans *HP-48 : faites vos jeux en assembleur* (DUNOD).

2

Entier système



| Entier système (<i>system binary</i>) | | |
|---|-----------------------------|------------------------|
| Prologue : 02911 | Taille totale : 10 quartets | |
| Type : 20 | Nombre de champs : 2 | |
| Classe : donnée | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02911 | @ | 5 quartets |
| Codage du nombre | @ + #5h | 5 quartets |
| Fin de l'objet | @ + #Ah | |



L'objet entier système est réservé au système et correspond à un nombre entier codé sur 5 quartets (20 bits). Il est utilisé directement par la HP-48 et ne doit pas être confondu avec un autre entier binaire (dont le prologue est 02A4E) plutôt destiné à l'utilisateur.

Les 5 quartets de « données utiles » représentent un nombre exprimé dans la base de numération active (base 2, 8, 10 ou 16). L'entier binaire système pourra, par exemple, représenter un prologue ou une adresse, ou encore, servir au transfert de paramètres entre deux programmes.

■ Quelques exemples

En mémoire, les dix quartets #1192009876h représentent un objet de type entier système dont la valeur est #67890h (à adapter à la base de numération active). Le microprocesseur SATURN inverse le contenu de chaque champ, le prologue #02911h est donc devenu #11920h alors que la donnée #67890h a été codée #09876h. Pour coder l'entier binaire valant #A2B3Ch, il suffit de juxtaposer le prologue « inversé » (#11920h) à la donnée inversée (#C3B2Ah). On obtient en mémoire les dix quartets 11920C3B2A codant l'entier binaire valant #A2B3Ch.

3

Nombre réel

| | | |
|----------------------------------|------------------------------------|------------------------|
| Nombre réel (real) | | |
| Prologue : 02933 | Taille totale : 21 quartets | |
| Type : 0 | Nombre de champs : 4 | |
| Classe : donnée | Epilogue : pas d'epilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02933 | @ | 5 quartets |
| Exposant (EEX) | @ + #5h | 3 quartets |
| Mantisse (MANT) | @ + #8h | 12 quartets |
| Signe (SIGN) | @ + #14h | 1 quartet |
| Fin de l'objet | @ + #15h | |



Cet objet est le nombre réel classique que manipule l'utilisateur. La donnée utile est divisée en trois champs que sont la mantisse, son signe et son exposant. L'exposant est en fait celui d'un facteur 10 associé à la mantisse. Le signe est codé sur un quartet qui vaut #0h si le nombre est positif ou #9h si le nombre est négatif. La mantisse comporte 12 chiffres représentés directement en base 10 par codage en DCB. L'exposant est codé sur trois chiffres en DCB. Si l'exposant est positif, ses 3 quartets DCB représentent directement l'exposant en base 10. Par contre, si l'exposant est négatif, on le remplace par « 1000 moins la valeur absolue de l'exposant ». Les exposants du facteur 10 sont compris entre -499 et 499.

■ Exemple

Quel objet représente en mémoire le réel $3,14159 \times 10^{-3}$? Respectant l'ordre des champs, juxtaposons le prologue inversé #33920h, l'exposant inversé #799h (l'exposant du facteur 10 étant négatif, on effectue « 1000 moins la valeur absolue de l'exposant » ce qui donne 997), la mantisse inversée #000000951413h (codage sur 12 quartets) et le signe #0h. On obtient en mémoire les 21 quartets #339207990000009514130h.

4

Réel long

| Long réel (<i>long real ou extended real</i>) | | |
|---|-----------------------------|-----------------|
| Prologue : 02955 | Taille totale : 26 quartets | |
| Type : 21 | Nombre de champs : 4 | |
| Classe : donnée | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02955 | @ | 5 quartets |
| Exposant (EEX) | @ + #5h | 5 quartets |
| Mantisse (MANT) | @ + #Ah | 15 quartets |
| Signe (SIGN) | @ + #19h | 1 quartet |
| Fin de l'objet | @ + #1Ah | |



Contrairement au nombre réel dont il vient d'être question, le nombre réel dit long n'est normalement pas accessible à l'utilisateur et n'est utilisé que pour les résultats intermédiaires d'un calcul en cours. Le codage de cet objet est très proche de celui du nombre réel classique, la seule différence réside en un plus grand nombre de quartets affectés à chaque champ. La donnée utile est divisée en trois champs que sont la mantisse, son signe et l'exposant du facteur 10. Le signe est codé sur un quartet qui vaut #0h si le nombre est positif, ou #9h si le nombre est négatif. La mantisse est codée sur 15 quartets DCB. L'exposant du facteur 10 est codé sur 5 quartets DCB. Si l'exposant est positif, ses 5 quartets DCB représentent directement l'exposant en base 10 (jusqu'à +49999). Par contre, si l'exposant est négatif, on le remplace par « 100000 moins la valeur absolue de l'exposant », le plus petit exposant du facteur 10 est donc -49999. Bien entendu, chaque champ est inversé.

■ Exemple

Reprenant le principe de codage des réels, -456×10^{-19999} sera représenté par les 26 quartets #55920 10008 0000000000000654 9h.

5

Nombre complexe

| | | |
|----------------------------------|------------------------------------|------------------------|
| Nombre complexe (complex) | | |
| Prologue : 02977 | Taille totale : 37 quartets | |
| Type : 1 | Nombre de champs : 7 | |
| Classe : donnée | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02977 | @ | 5 quartets |
| Exposant (réel) | @ + #5h | 3 quartets |
| Mantisse (réel) | @ + #8h | 12 quartets |
| Signe (réel) | @ + #14h | 1 quartet |
| Exposant (imaginaire) | @ + #15h | 3 quartets |
| Mantisse (imaginaire) | @ + #18h | 12 quartets |
| Signe (imaginaire) | @ + #24h | 1 quartet |
| <i>Fin de l'objet</i> | @ + #25h | |



Cet objet correspond au nombre complexe manipulable par l'utilisateur. La représentation d'un nombre complexe n'est autre que la juxtaposition de deux nombres réels. Le premier représente la partie réelle du complexe alors que le second représente la partie imaginaire du complexe. Comme toujours, le prologue occupe 5 quartets, il est ici suivi de deux nombres réels codés sur 16 quartets chacun.

■ Exemple

Pour représenter le complexe $1000\ 0000\ 0000 + i.2000\ 0000\ 0000$ on utilise l'objet #77920 110 0000000000001 0 110 0000000000002 0h où chaque groupe de quartets correspond respectivement au prologue, à l'exposant de la partie réelle (exposant du facteur 10), à la mantisse de la partie réelle (DCB), au signe de la mantisse réelle, à l'exposant de la partie imaginaire (exposant du facteur 10), à la mantisse de la partie imaginaire (DCB) et au signe de la mantisse imaginaire.

6

Complexe long



| Nombre complexe long (<i>long complex/extended complex</i>) | | |
|---|-----------------------------|-----------------|
| Prologue : 0299D | Taille totale : 47 quartets | |
| Type : 22 | Nombre de champs : 7 | |
| Classe : donnée | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 0299D | @ | 5 quartets |
| Exposant (réel) | @ + #5h | 5 quartets |
| Mantisse (réel) | @ + #Ah | 15 quartets |
| Signe (réel) | @ + #19h | 1 quartet |
| Exposant (imaginaire) | @ + #1Ah | 5 quartets |
| Mantisse (imaginaire) | @ + #1Fh | 15 quartets |
| Signe (imaginaire) | @ + #2Eh | 1 quartet |
| <i>Fin de l'objet</i> | @ + #2Fh | |



La représentation du nombre complexe destiné à l'utilisateur était fondé sur la juxtaposition de deux réels. La représentation de ce nombre complexe « long » destiné aux calculs intermédiaires effectués par la HP-48 est fondée sur la juxtaposition de deux nombres réels double précision.

■ Exemple

Nous coderons $-10 + i.20$ sous la forme d'un objet de type nombre complexe double précision. On obtient les 47 quartets suivants : #0299D 10000 1000000000000000 9 10000 2000000000000000 0h. Bien entendu, le DCB est utilisé et tous les champs sont inversés. Notez que les mantisses des deux nombres réels double précision comportent une partie réelle limitée à 1 chiffre d'où l'emploi de facteurs valant 10^1 . Par exemple, pour représenter une mantisse égale à 457 896 125 548 124, il faut lui associer un exposant 10^{14} .

7

Caractère

| Caractère (<i>character</i>) | | |
|--------------------------------|----------------------------|------------------------|
| Prologue : 029BF | Taille totale : 7 quartets | |
| Type : 24 | Nombre de champs : 2 | |
| Classe : donnée | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 029BF | @ | 5 quartets |
| Code ASCII | @ + #5h | 2 quartets |
| Fin de l'objet | @ + #7h | |



Cet objet est l'un des plus simples : le prologue est plus long que les données utiles ! Ne confondez pas l'objet caractère (caractère isolé) avec l'objet chaîne de caractères.

L'objet caractère est accessible à l'utilisateur. Il s'agit tout simplement de l'un des caractères que peut afficher la HP-48. La HP-48 dispose d'un code ASCII étendu comprenant quelques caractères particuliers. Vous pouvez visualiser l'ensemble des caractères affichables par la HP-48 en utilisant la commande CHARS (HP-48 G/GX).

La table ASCII de la HP-48 comporte 256 caractères comme tout ASCII étendu ; notez qu'il y a 127 symboles - codables sur 7 bits - dans la table ASCII de base. Chaque caractère est donc codé par deux quartets, c'est-à-dire un octet. On a 16 × 16 possibilités, donc 256 caractères.

■ Exemple

Pour coder un caractère, il suffit de se reporter à la table ASCII (voir le chapitre consacré à l'ASCII) et de considérer le code hexadécimal associé au caractère. Par exemple, 'Z' a pour code #5Ah. Il est donc codé en mémoire par l'objet #FB920 A5h (n'oubliez pas d'inverser chaque champ).



Chaîne de caractères



| Chaîne de caractères (<i>string ou character string</i>) | | |
|--|--------------------------------------|-----------------|
| Prologue : 02A2C | Taille totale : au moins 12 quartets | |
| Type : 2 | Nombre de champs : au moins 3 | |
| Classe : donnée | Exécution indirecte : DOCSTR | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02A2C | @ | 5 quartets |
| Longueur hors prologue L_{t-p} | @ + #5h | 5 quartets |
| Caractère 1 | @ + #Ah | 2 quartets |
| ... | ... | ... |
| Caractère n | @ + #A+(n×5)h | 2 quartets |
| Fin de l'objet | @ + #A+5+(n×5)h | |



Les chaînes de caractères servent à la représentation de messages. Elles consistent en une juxtaposition de codes ASCII sur 2 quartets, chaque code représentant un caractère. Une chaîne de caractères est définie par son prologue, sa longueur totale hors prologue suivie des codes de tous les caractères de la chaîne.

Un vecteur (objet tableau à une dimension) contenant des caractères peut être assimilé à une chaîne facilitant la recherche d'un n -ième caractère au sein de la chaîne.

■ Exemple

La chaîne de caractères "HP-48" sera codée de la façon suivante : le prologue #02A2Ch, la longueur totale hors prologue #0000Fh puis les codes ASCII des caractères #48h, #50h, #2Dh, #34h et #38h. En inversant chaque champ puis en les juxtaposant, on obtient en mémoire l'objet #C2A20 F0000 84 05 D2 43 83h.

9

Entier binaire



| Entier binaire (<i>Binary integer</i>) | | |
|--|--------------------------------------|-----------------|
| Prologue : 02A4E | Taille totale : au moins 11 quartets | |
| Type : 10 | Nombre de champs : 3 | |
| Classe : donnée | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02A4E | @ | 5 quartets |
| Longueur hors prologue | @ + #5h | 5 quartets |
| Chiffre 1 | @ + #Ah | 1 quartet |
| ... | ... | ... |
| Chiffre n | @ + #5 + (n×5)h | 1 quartet |
| Fin de l'objet | @ + #5 + 5 + (n×5)h | |

Cet objet correspond à l'entier binaire auquel a normalement accès l'utilisateur. Il est représenté par des chiffres précédés par # et suivis d'une lettre minuscule déterminant la base de numération dans laquelle est exprimée l'entier binaire.



Attention ! La base active (voir le menu BASE accessible par [MTH]) s'applique aux objets de type entier binaire au détriment de la base utilisée lors de leur saisie. La conversion est automatique.

Notez qu'il faut déclarer un quartet par chiffre et que les problèmes d'exposant de facteur 10 n'interviennent pas ici. Le nombre de chiffres hexadécimaux (codage sur un quartet pour chacun) est illimité, on constitue ainsi des chaînes hexadécimales d'une longueur adaptée aux besoins.

■ Exemple

L'entier binaire valant #1Ah est codé par les 12 quartets #E4A20 70000 A1h représentant le prologue, la longueur hors prologue et #1Ah.

10

Liste



| | | |
|----------------------------------|---|------------------------|
| Liste (<i>list</i>) | | |
| Prologue : 02A74 | Taille totale : au moins 10 quartets | |
| Type : 5 | Nombre de champs : au moins 2 | |
| Classe : donnée | Epilogue : 0312B | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02A74 | @ | 5 quartets |
| Objet 1 | @ + #5h | <i>selon l'objet</i> |
| ... | ... | ... |
| Objet n | @ + #(...)h | <i>selon l'objet</i> |
| Epilogue = 0312B | @ + #5 + (...)h | 5 quartets |
| <i>Fin de l'objet</i> | @ + #A + (...)h | |

Cet objet contient une liste d'objets de types différents et quelconques. La liste sert donc à encapsuler une série d'objets divers. Cet objet est borné par un prologue et par un épilogue. Entre le prologue et l'épilogue on trouve des objets complets avec leurs propres prologues. Il est possible de remplacer un objet de la liste par son adresse. Cette facilité est très intéressante dans le cadre de l'utilisation de programmes en mémoire morte.



Notez qu'une liste n'occupant que 10 quartets (pour les deux champs que sont le prologue et l'épilogue) est une liste vide.

■ Quelques exemples

Soit une liste { #1h A "HP" } comportant un entier binaire, un caractère et une chaîne de caractères. On se contente de coder ces trois objets déjà connus et de les encadrer avec un prologue et un épilogue. On obtient #47A20 E4A20 60000 1 FB920 14 C2A20 90000 84 05h où #E4A20 60000 1h code l'entier binaire, #FB920 14h code le caractère et #C2A20 90000 84 05h code la chaîne des caractères.

11

Expression algébrique



| Expression algébrique (<i>algebraic ou symbolic</i>) | | |
|--|--------------------------------------|------------------------|
| Prologue : 02AB8 | Taille totale : au moins 10 quartets | |
| Type : 9 | Nombre de champs : au moins 2 | |
| Classe : donnée | Epilogue : 0312B | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02AB8 | @ | 5 quartets |
| Objet n°1 | @ + #5h | selon objet n°1 |
| ... | ... | ... |
| Objet n°n | ... | selon objet n°n |
| Epilogue = 0312B | ... | 5 quartets |
| <i>Fin de l'objet</i> | selon les tailles des objets | |

Cet objet correspond à l'expression algébrique couramment manipulée en RPL. Il est délimité par un prologue (au début) et par un épilogue (à la fin). Cet objet contient lui-même de nombreux objets constituant l'expression algébrique codée en notation polonaise inversée même si la saisie et la restitution de l'objet dans la pile utilisent la notation algébrique classique. Remarquez que la notation polonaise inversée (RPN), ou notation postfixe, n'utilise aucune parenthèse. Notez que les opérations intervenant dans l'expression algébrique sont codées par l'adresse (sur 5 quartets) du « programme » placé en ROM qui correspond à l'opération.

■ Exemple

Codons l'expression algébrique '3xA'. En RPN, '3xA' devient « 3 A × ». L'objet commence par le prologue 8BA20, suivi du premier objet (un réel) codé par 33920 000 000000000003 0, suivi du deuxième objet (un nom global) codé par 84E20 10 16, suivi du troisième objet (adresse #1ADEEh en ROM) codé par EEDA1. Une fois le prologue ajouté, on obtient l'objet #8BA20 33920 000 000000000003 0 84E20 10 16 EEDA1 B2130h.

12

Objet avec unité

| | | |
|--|--|--|
| Objet avec unité (unit) | | |
| Prologue : 02ADA | Taille totale : selon l'objet et l'unité | |
| Type : 13 | Nombre de champs : au moins 4 | |
| Classe : donnée | Epilogue : 0312B | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02ADA Objet Unité associée à l'objet Epilogue = 0312B <i>Fin de l'objet</i> | @ @ + #5h selon l'objet selon l'objet et l'unité <i>selon l'objet et l'unité</i> | 5 quartets selon l'objet selon l'unité selon l'objet et l'unité |



L'objet muni d'une unité est délimité par un prologue (au début) et par un épilogue (à la fin). Il contient une suite d'objets dont le premier est l'objet à munir d'une unité (un réel par exemple) et les suivants sont le codage de l'unité proprement dite à l'aide de chaînes de caractères et d'opérations spécifiques de type produit codées par leurs adresses en ROM (la multiplication est à l'adresse #10B86h en ROM, la division en #10B68h et l'exponentiation en #10B72h).

■ Exemple

Codons 10 m.s⁻¹, c'est-à-dire 10 m/s. L'objet commence par le prologue ADA20, suivi de l'objet réel codant 10, c'est-à-dire 33920 100 000000000001 0. On code ensuite l'unité « m/s » qui devient « m s / » en notation polonaise inversée (notation postfixe). « m » et « s » sont des chaînes de caractères codées respectivement par C2A20 70000 D6 et par C2A20 70000 37. On code la division par 86B01 puis la multiplication (de l'unité par l'objet à munir de l'unité) par 68B01. Une fois l'épilogue B2130 ajouté, on obtient l'objet #ADA20 33920 100 000000000001 0 C2A20 70000 D6 C2A20 70000 37 86B01 68B01 B2130h sur 65 quartets.

13

Objet graphique

| Objet graphique (<i>Graphic ou GROB</i>) | | |
|--|--------------------------------------|------------------|
| Prologue : 02B1E | Taille totale : au moins 22 quartets | |
| Type : 11 | Nombre de champs : au moins 5 | |
| Classe : donnée | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02B1E | @ | 5 quartets |
| Taille hors prologue | @ + #5h | 5 quartets |
| Nombre de lignes | @ + #Ah | 5 quartets |
| Nombre de colonnes | @ + #Fh | 5 quartets |
| Pixels en haut à gauche | @ + #14h | codage par octet |
| ... | ... | ... |
| Pixels en bas à droite | selon GROB | codage par octet |
| <i>Fin de l'objet</i> | selon GROB | |



Une image est une juxtaposition de pixels (*picture element* ou points de l'écran). L'objet est codé par un prologue (02B1E) suivi de la taille hors prologue de l'objet, du nombre de lignes (nombre de pixels verticaux), du nombre de colonnes (nombre de pixels horizontaux), puis de la description pixel par pixel de l'image en partant de l'angle supérieur gauche de l'image jusqu'à l'angle inférieur droit (on parcourt tous les pixels un à un de haut en bas et de gauche à droite). A chaque pixel correspond un bit (un bit à 1 égale un pixel allumé). Le codage s'effectue par octets. Chaque octet représente huit points placés horizontalement et se divise en deux quartets lus à l'envers. Même si l'image ne comporte pas un nombre horizontal de points multiple de huit, nous sommes obligés de coder les pixels « inexistants ». Par exemple, codons l'objet graphique  qui comporte 6 pixels dont 1 est « allumé ». Traduite en bits, l'image devient 0100 00, qui devient l'octet 0100 0000 dont les deux derniers chiffres ne sont pas significatifs. Chaque quartet étant lu à l'envers, 0100 0000₂ devient 0010 0000₂ puis 20₁₆ après une conversion en hexadécimal. On obtient l'objet #02B1E 11000 10000 60000 20h sur 27 quartets.

14

Programme RPL

| Programme RPL (<i>program</i>) | | |
|----------------------------------|--------------------------------------|-----------------|
| Prologue : 02D9D | Taille totale : au moins 20 quartets | |
| Type : 8 | Nombre de champs : au moins 4 | |
| Classe : donnée | Epilogue : 0312B | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02D9D | @ | 5 quartets |
| « | @ + #5h | 5 quartets |
| objets divers... | @ + #Ah | 5 quartets |
| ... | ... | ... |
| » | @ _f - #Ah | 5 quartets |
| Epilogue = 0312B | @ _f - #5h | 5 quartets |
| <i>Fin de l'objet</i> | @ _f | |



La structure de l'objet programme RPL est très proche de celle de la liste. Un épilogue et un prologue encadrent une suite d'objets. Ceux-ci peuvent être remplacés par leurs adresses en mémoire. C'est notamment le cas des fonctions de la HP-48. Par exemple, l'adresse #1AB67h désigne la fonction d'addition.

Les guillemets « et » encadrent les objets (des fonctions et leurs arguments) d'un programme RPL. Ces guillemets peuvent aussi être utilisés (toujours par paires) pour délimiter des procédures indépendantes au sein d'un programme RPL. Les symboles « et » se trouvent respectivement aux adresses #2361Eh et #23639h.

Un programme RPL se compose d'une foule d'objets : des fonctions qui seront appelées par leur adresse en mémoire morte et les arguments de celles-ci (des nombres, les listes, des chaînes de caractères ,etc.).

15

Nom global

| | | |
|--|--|------------------------|
| Nom global (global name) | | |
| Prologue : 02E48 | Taille totale : au moins 7 quartets | |
| Type : 6 | Nombre de champs : au moins 2 | |
| Classe : identificateur | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02E48 | @ | 5 quartets |
| N_c | @ + #5h | 2 quartets |
| Caractère n°1 | @ + #5+2×1h | 2 quartets |
| Caractère n°2 | @ + #5+2×2h | 2 quartets |
| Caractère n°3 | @ + #5+2×3h | 2 quartets |
| Caractère n°4 | @ + #5+2×4h | 2 quartets |
| Caractère n°5 | @ + #5+2×5 | 5 quartets |
| ... | ... | ... |
| Caractère n° N_c | @ + #5+2× N_c h | 2 quartets |
| <i>Fin de l'objet</i> | @ + #7+2× N_c h | |
| NB : | | |
| N_c est le nombre de caractères ASCII constituant le nom global. | | |

Un nom global est placé entre des délimiteurs ' et '. Un exemple de nom global est 'HP-48'. Les caractères du nom sont codés en ASCII.

16

Nom local

Seul le prologue distingue le nom local du nom global. Un nom local n'est accessible que dans un environnement particulier.



| | | |
|--------------------------------------|------------------------------------|------------------------|
| Nom local (<i>local name</i>) | | |
| Prologue : 02E6D | Taille totale : variable | |
| Type : 7 | Nombre de champs : variable | |
| Classe : identificateur | Epilogue : pas d'épilogue | |
| Implantation en mémoire : | | |
| Champ | Adresse | Taille du champ |
| Prologue = 02E48 | @ | 5 quartets |
| Nombre de caractères N_c | @ + #5h | 2 quartets |
| Caractère n°1 | @ + #5+2×1h | 2 quartets |
| Caractère n°2 | @ + #5+2×2h | 2 quartets |
| ... | ... | ... |
| Caractère n° N_c | @ + #5+2× N_c h | 2 quartets |
| <i>Fin de l'objet</i> | @ + #7+2× N_c h | |

17

Remarques importantes

■ Les délimiteurs d'objets

Les délimiteurs des objets ne sont pas codés. C'est le prologue qui détermine les délimiteurs affichés lors de l'édition. Cela dit les délimiteurs « et » d'un programme RPL doivent être codés.

■ Taille fixe ou variable des objets

Certains objets ont une taille fixe (le réel par exemple). Le prologue détermine donc la longueur de l'objet, et par conséquent, l'adresse de son dernier quartet. Dans le cas d'un objet de taille variable, la longueur et l'adresse de fin de l'objet peuvent être indiquées à la HP-48 de deux façons différentes : soit un épilogue borne l'objet, soit la longueur totale hors prologue de l'objet est placée juste après le prologue en tête de l'objet.

■ Codage de la mantisse

Attention ! Dans le cas des réels, complexes, réels longs et complexes longs, la mantisse est de la forme 1,234567890 et non 1234567890. La mantisse a donc une partie entière limitée à un chiffre. Ainsi, pour représenter 12,34567890, il faudra que l'exposant du facteur 10 (appelé couramment « exposant ») vaille 1... Car il faut multiplier la mantisse par $10^1=10...$

Ainsi, pour faire passer tous les chiffres de la mantisse (nombres réels et complexes) devant la virgule, il faudra que l'exposant vaille 11 (en fait, 10^{11}). De même pour les réels double précision et complexes double précisions, l'exposant du facteur 10 devra valoir 14 pour que tous les chiffres de la mantisse passent devant la virgule.

■ Codage... Attention !

Pour créer un objet, il suffit de saisir dans une chaîne de caractères (délimitée par " et ") chaque champ de l'objet à créer. Tous les champs devront bien entendu être inversés, ainsi #1A2h deviendra #2A1h. La chaîne de caractères étant au premier niveau de la pile exécutez le programme ASS ou le programme GASS, couramment employés par les programmeurs en assembleur. Il ne vous reste plus ensuite qu'à sauver votre objet dans une variable globale à l'aide de [STO].

Sauvegardez le contenu de votre mémoire avant de créer manuellement vos propres objets... Les « Memory lost » sont moins rares que vous le pensez !

Instructions RPL : classement alphabétique



La liste ci-après présente les adresses d'environ 400 instructions RPL-utilisateur classées selon l'ordre alphabétique. Ces instructions ont déjà été présentées dans la deuxième partie de l'ouvrage. Les adresses des instructions sont utilisables avec SYSEVAL ou au sein d'un programme RPL-système. Cette liste est destinée à l'élaboration des listings RPL-système.

| Instructions | Adresses | Instructions | Adresses |
|--------------|----------|--------------|----------|
| == | #1E972h | ACOS | #1B72Fh |
| ! | #1BB02h | ACOSH | #1B830h |
| << | #2361Eh | →ARRY | #1D009h |
| >> | #23639h | →DATE | #1989Eh |
| % | #1C060h | →GROB | #1E5ADh |
| %CH | #1C149h | →HMS | #1BF1Eh |
| %T | #1C0D7h | →LCD | #1E58Dh |
| * | #1ADEEh | →LIST | #1C783h |
| *H | #1E150h | →NUM | #1A5E4h |
| *W | #1E170h | →Q | #1F9C4h |
| + | #1AB67h | →Qπ | #1F9E9h |
| - | #1AD09h | →STR | #1CB0Bh |
| < | #1EBBEh | →TAG | #225BEh |
| = | #1A8D8h | →TIME | #198BEh |
| > | #1EC5Dh | →UNIT | #1974Fh |
| ABS | #1AA1Fh | →V2 | #1DE66h |
| ACK | #1987Eh | →V3 | #1DEC2h |
| ACKALL | #19863h | ALOG | #1BA3Dh |

| Instructions | Adresses | Instructions | Adresses |
|---------------------|-----------------|---------------------|-----------------|
| AND | #1E783h | CLUSR | #210FCh |
| APPLY | #1F55Dh | CLΣ | #1FD2Bh |
| ARC | #1E5D2h | CNRM | #1BFBEh |
| ARCHIVE | #2125Ah | COLCT | #20A15h |
| ARG | #1B2DBh | COMB | #1C1F6h |
| ARRY→ | #1D092h | CON | #1D186h |
| ASIN | #1B6A4h | CONIC | #1E681h |
| ASINH | #1B7EBh | CONJ | #1AA6Eh |
| ASN | #224F4h | CONT | #1A8BBh |
| ASR | #1957Bh | CONVERT | #196DBh |
| ATAN | #1B79Ch | CORR | #1FDC1h |
| ATANH | #1B8A2h | COS | #1B505h |
| ATTACH | #21448h | COSH | #1B606h |
| AUTO | #1E1ABh | COV | #1FDDCh |
| AXES | #1E0BEh | CR | #1EEA4h |
| B→R | #196BBh | CRDIR | #1A105h |
| BAR | #1E741h | CROSS | #1C01Eh |
| BARPLOT | #20133h | D→R | #1BEC8h |
| BAUD | #2200Ch | DATE | #19812h |
| BEEP | #1A5C4h | DATE+ | #199D2h |
| BEGIN | #23654h | DDAYS | #199B2h |
| BESTFIT | #2025Eh | DEC | #1C574h |
| BIN | #1C559h | DECR | #209AAh |
| BINS | #2010Eh | DEFINE | #20D65h |
| BLANK | #1E416h | DEG | #1C399h |
| BOX | #1E3ECh | DELALARM | #19972h |
| BUFLEN | #22087h | DELAY | #1EF43h |
| BYTES | #1A1D9h | DELKEYS | #22548h |
| C→PX | #1E29Ah | DEPND | #1E22Bh |
| C→R | #1C98Eh | DEPTH | #1FC44h |
| CASE | #2378Dh | DET | #1BFDEh |
| CEIL | #1BC0Fh | DETACH | #2147Ch |
| CENTR | #1E0E8h | DIR | #23813h |
| CF | #1C2D5h | DISP | #1A584h |
| CHR | #1CB66h | DO | #230C3h |
| CKSM | #21FECh | DOERR | #1A339h |
| CLEAR | #1FCEBh | DOT | #1BFFEh |
| CLKADJ | #198DEh | DRAW | #1E190h |
| CLLCD | #1A858h | DRAX | #1E1C6h |
| CLOSEIO | #21ED5h | DROP | #1FBD8h |

| Instructions | Adresses | Instructions | Adresses |
|-----------------------|----------|--------------|----------|
| DROP2 | #1FBF3h | GRAD | #1C3CFh |
| DROPN | #1FC64h | GRAPH | #1E2BAh |
| DTAG | #22633h | GXOR | #1E4E4h |
| DUP | #1FB87h | HALT | #23472h |
| DUP2 | #1FBA2h | HEX | #1C58Fh |
| DUPN | #1FC7Fh | HISTOGRAM | #1E721h |
| e | #1AB23h | HISTPLOT | #20167h |
| ELSE | #22FB5h | HMS+ | #1BF5Eh |
| END | #23679h | HMS- | #1BF7Eh |
| END (structure DO) | #236B9h | HMS→ | #1BF3Eh |
| END (structure IF) | #22FD5h | HOME | #1A140h |
| END (structure WHILE) | #23694h | i | #1AB45h |
| ENG | #1C452h | IDN | #1D2DCh |
| EQ→ | #1CEE3h | IF | #22EC3h |
| ERASE | #1E25Fh | IFERR | #233DFh |
| ERR0 | #1A36Dh | IFT | #1A4CDh |
| ERRM | #1A3A3h | IFTE | #1A3FEh |
| ERRN | #1A388h | IM | #1C819h |
| EVAL | #1A3BEh | INCR | #208F4h |
| EXP | #1B905h | INDEP | #1E04Ah |
| EXPAN | #20A49h | INPUT | #224CAh |
| EXPFIT | #201FBh | INV | #1B278h |
| EXPM | #1BAC2h | IP | #1BB6Dh |
| FACT | #1BB41h | ISOL | #20A93h |
| FC? | #1C360h | KERRM | #2206Ch |
| FC?C | #1C520h | KEY | #1A873h |
| FINDALARM | #19948h | KGET | #21F24h |
| FINISH | #21FB6h | KILL | #1A303h |
| FIX | #1C3EAh | LABEL | #1E2D5h |
| FLOOR | #1BBD9h | LAST | #1A604h |
| FOR | #231A0h | LCD→ | #1E572h |
| FP | #1BBA3h | LIBS | #2142Dh |
| FREE | #213D1h | LINE | #1E398h |
| FREEZE | #1A5A4h | LINFIT | #201B1h |
| FS? | #1C313h | LIST→ | #1C95Ah |
| FS?C | #1C4A1h | LN | #1B94Fh |
| FUNCTION | #1E661h | LNP1 | #1BA8Ch |
| GET | #1D7C6h | LOG | #1B9C6h |
| GETI | #1D8C7h | LOGFIT | #201D6h |
| GOR | #1E456h | LR | #1FF20h |

| Instructions | Adresses | Instructions | Adresses |
|--------------|----------|--------------|----------|
| MANT | #1BE9Ch | PMAX | #1E09Eh |
| MATCH↓ | #1FA8Dh | PMIN | #1E07Eh |
| MATCH↑ | #1FA59h | POLAR | #1E6A1h |
| MAX | #1BC71h | POS | #1CAB4h |
| MAXR | #1AADFh | PR1 | #1EE53h |
| MAX Σ | #1FE7Eh | PREDV | #1FF7Ah |
| MEAN | #1FE99h | PREDX | #1FFBAh |
| MEM | #20FAAh | PREDY | #1FF9Ah |
| MENU | #21196h | PRLCD | #1EF63h |
| MERGE | #2137Fh | PROMPT | #23824h |
| MIN | #1BCE3h | PRST | #1EE89h |
| MINR | #1AB01h | PRSTC | #1EE6Eh |
| MIN Σ | #1FEB4h | PRVAR | #1EEBFh |
| MOD | #1BE4Dh | PURGE | #20EFEh |
| NEG | #1A995h | PUT | #1D407h |
| NEWOB | #1A2BCh | PUTI | #1D5DFh |
| NEXT | #2324Ch | PVARS | #211FCh |
| NOT | #1E88Fh | PVIEW | #1E2F0h |
| NUM | #1CB46h | PWRFIT | #20220h |
| N Σ | #1FDA6h | PX→C | #1E27Ah |
| OBJ→ | #1CF7Bh | QUAD | #20AB3h |
| OCT | #1C5AAh | QUOTE | #1F500h |
| OFF | #1A31Eh | RAD | #1C3B4h |
| OLDPRT | #1EE38h | R→B | #1969Bh |
| OPENIO | #21EB5h | R→C | #1C79Eh |
| OR | #1E809h | R→D | #1BEF4h |
| ORDER | #20FD9h | RAND | #1C1B9h |
| OVER | #1FC29h | RATIO | #1FB5Dh |
| PARAMETRIC | #1E6C1h | RCEQ | #1F133h |
| PARITY | #2202Ch | RCL | #20B40h |
| PATH | #1A125h | RCLALARM | #19928h |
| PDIM | #1E201h | RCLF | #1C619h |
| PERM | #1C236h | RCLKEYS | #22586h |
| PGDIR | #2123Ah | RCLMENU | #211E1h |
| PICK | #1FC9Ah | RCL Σ | #1FD46h |
| PICT | #1E436h | RCWS | #1C5FEh |
| PIX? | #1E36Eh | RDM | #1D0DFh |
| PIXOFF | #1E344h | RDZ | #1C1D4h |
| PIXON | #1E320h | RE | #1C7CAh |
| PKT | #220DDh | REC� | #21F62h |

| Instructions | Adresses | Instructions | Adresses |
|--------------|----------|-----------------------|----------|
| RECV | #21F96h | SRB | #1967Bh |
| REPEAT | #2305Dh | SRECV | #21E95h |
| REPL | #1C8EAh | START | #23103h |
| RES | #1E126h | STD | #1C486h |
| RESTORE | #2133Ch | STEP | #23380h |
| RL | #1959Bh | STEQ | #1F14Eh |
| RLB | #195BBh | STIME | #220A2h |
| RND | #1BD55h | STO | #20CCDh |
| RNRM | #1BF9Eh | STO* | #20753h |
| ROLL | #1FCB5h | STO+ | #2044Bh |
| ROLLD | #1FCD0h | STO- | #20358h |
| ROOT | #1F16Eh | STO/ | #2060Ch |
| ROT | #1FC0Eh | STOALARM | #198FEh |
| RR | #195DBh | STOF | #1C67Fh |
| RRB | #195FBh | STOKEYS | #22514h |
| RSD | #1C03Eh | STO Σ | #1FD0Bh |
| RULES | #20A7Dh | STR \rightarrow | #1CB26h |
| SAME | #1E761h | STWS | #1C5C5h |
| SBRK | #220C2h | SUB | #1C85Ch |
| SCALE | #1E1E1h | SWAP | #1FBBDh |
| SCATRPLOT | #2018Ch | SYSEVAL | #1A52Eh |
| SCATTER | #1E701h | TAN | #1B55Eh |
| SCI | #1C41Eh | TANH | #1B655h |
| SCL Σ | #200C4h | TAYLR | #20B20h |
| SCONJ | #203CCh | TEXT | #1E606h |
| SDEV | #1FECFh | THEN | #22EFAh |
| SEND | #21EF0h | THEN (structure CASE) | #237A8h |
| SERVER | #21FD1h | THEN (structure ERR) | #2371Fh |
| SF | #1C274h | TIME | #197F7h |
| SHOW | #20AD3h | TLINE | #1E3C2h |
| SIGN | #1B32Ah | TMENU | #2115Dh |
| SIN | #1B4ACh | TOT | #1FEEAh |
| SINH | #1B5B7h | TRANSIO | #2204Ch |
| SINV | #202CEh | TRN | #1D392h |
| SIZE | #1C9B8h | TRNC | #1BDD1h |
| SL | #1961Bh | TRUTH | #1E6E1h |
| SLB | #1963Bh | TSTR | #19992h |
| SNEG | #2034Dh | TVARS | #1A1AFh |
| SQ | #1B426h | TYPE | #1CB86h |
| SR | #1965Bh | UBASE | #19771h |

| Instructions | Adresses | Instructions | Adresses |
|---------------------|-----------------|---------------------|-----------------|
| UFACT | #197A5h | XROOT | #1B185h |
| UNTIL | #230EDh | YCOL | #1FFFAh |
| UPDIR | #1A15Bh | YRNG | #1E641h |
| UTPC | #20020h | ^ | #1B02Dh |
| UTPF | #2005Ah | - | #1FAEBh |
| UTPN | #2003Ah | ≠ | #1EA9Dh |
| UTPT | #2007Ah | ≤ | #1ECFCh |
| UVAL | #1971Bh | ≥ | #1ED9Bh |
| V→ | #1DD06h | ∑ | #1F2C9h |
| VAR | #1FF05h | ∑+ | #1FD61h |
| VARS | #1A194h | ∑- | #1FD8Bh |
| VTYPE | #1CE28h | ∑COL | #2009Ah |
| WAIT | #1A71Fh | ∑LINE | #200F3h |
| WHILE | #23033h | ∑x | #1FDF7h |
| WSLOG | #19848h | ∑x*y | #1FE63h |
| XCOL | #1FFDAh | ∑x^2 | #1FE2Dh |
| XMIT | #21E75h | ∑y | #1FE12h |
| XOR | #1E8F6h | ∑y^2 | #1FE48h |
| XPON | #1BC45h | ÷ | #1AF05h |
| XRNG | #1E621h | | |

Instructions RPL : classement par adresses



La liste ci-après présente les adresses d'environ 400 instructions RPL-utilisateur classées selon l'ordre croissant de leurs adresses. Ces instructions ont déjà été présentées dans la deuxième partie de l'ouvrage. Les adresses des instructions sont utilisables avec SYSEVAL ou au sein d'un programme RPL-système. Cette liste est destinée à la compréhension des listings RPL-système.

| Adresses | Instructions | Adresses | Instructions |
|----------|--------------|----------|--------------|
| #1957Bh | ASR | #19848h | WSLOG |
| #1959Bh | RL | #19863h | ACKALL |
| #195BBh | RLB | #1987Eh | ACK |
| #195DBh | RR | #1989Eh | →DATE |
| #195FBh | RRB | #198BEh | →TIME |
| #1961Bh | SL | #198DEh | CLKADJ |
| #1963Bh | SLB | #198FEh | STOALARM |
| #1965Bh | SR | #19928h | RCLALARM |
| #1967Bh | SRB | #19948h | FINDALARM |
| #1969Bh | R→B | #19972h | DELALARM |
| #196BBh | B→R | #19992h | TSTR |
| #196DBh | CONVERT | #199B2h | DDAYS |
| #1971Bh | UVAL | #199D2h | DATE+ |
| #1974Fh | →UNIT | #1A105h | CRDIR |
| #19771h | UBASE | #1A125h | PATH |
| #197A5h | UFACT | #1A140h | HOME |
| #197F7h | TIME | #1A15Bh | UPDIR |
| #19812h | DATE | #1A194h | VARS |

| Adresses | Instructions | Adresses | Instructions |
|-----------------|---------------------|-----------------|---------------------|
| #1A1AFh | TVARS | #1B4ACH | SIN |
| #1A1D9h | BYTES | #1B505h | COS |
| #1A2BCh | NEWOB | #1B55Eh | TAN |
| #1A303h | KILL | #1B5B7h | SINH |
| #1A31Eh | OFF | #1B606h | COSH |
| #1A339h | DOERR | #1B655h | TANH |
| #1A36Dh | ERR0 | #1B6A4h | ASIN |
| #1A388h | ERRN | #1B72Fh | ACOS |
| #1A3A3h | ERRM | #1B79Ch | ATAN |
| #1A3BEh | EVAL | #1B7EBh | ASINH |
| #1A3FEh | IFTE | #1B830h | ACOSH |
| #1A4CDh | IFT | #1B8A2h | ATANH |
| #1A52Eh | SYSEVAL | #1B905h | EXP |
| #1A584h | DISP | #1B94Fh | LN |
| #1A5A4h | FREEZE | #1B9C6h | LOG |
| #1A5C4h | BEEP | #1BA3Dh | ALOG |
| #1A5E4h | →NUM | #1BA8Ch | LNP1 |
| #1A604h | LAST | #1BAC2h | EXPM |
| #1A71Fh | WAIT | #1BB02h | ! |
| #1A858h | CLLCD | #1BB41h | FACT |
| #1A873h | KEY | #1BB6Dh | IP |
| #1A8BBh | CONT | #1BBA3h | FP |
| #1A8D8h | = | #1BBD9h | FLOOR |
| #1A995h | NEG | #1BC0Fh | CEIL |
| #1AA1Fh | ABS | #1BC45h | XPON |
| #1AA6Eh | CONJ | #1BC71h | MAX |
| #1AADFh | MAXR | #1BCE3h | MIN |
| #1AB01h | MINR | #1BD55h | RND |
| #1AB23h | e | #1BDD1h | TRNC |
| #1AB45h | i | #1BE4Dh | MOD |
| #1AB67h | + | #1BE9Ch | MANT |
| #1AD09h | - | #1BEC8h | D→R |
| #1ADEEh | * | #1BEF4h | R→D |
| #1AF05h | ÷ | #1BF1Eh | →HMS |
| #1B02Dh | ^ | #1BF3Eh | HMS→ |
| #1B185h | XROOT | #1BF5Eh | HMS+ |
| #1B278h | INV | #1BF7Eh | HMS- |
| #1B2DBh | ARG | #1BF9Eh | RNRM |
| #1B32Ah | SIGN | #1BFBEh | CNRM |
| #1B426h | SQ | #1BFDEh | DET |

| Adresses | Instructions | Adresses | Instructions |
|----------|--------------|----------|--------------|
| #1BFFEh | DOT | #1CAB4h | POS |
| #1C01Eh | CROSS | #1CB0Bh | →STR |
| #1C03Eh | RSD | #1CB26h | STR→ |
| #1C060h | % | #1CB46h | NUM |
| #1C0D7h | %T | #1CB66h | CHR |
| #1C149h | %CH | #1CB86h | TYPE |
| #1C1B9h | RAND | #1CE28h | VTYP |
| #1C1D4h | RDZ | #1CEE3h | EQ→ |
| #1C1F6h | COMB | #1CF7Bh | OBJ→ |
| #1C236h | PERM | #1D009h | →ARRY |
| #1C274h | SF | #1D092h | ARRY→ |
| #1C2D5h | CF | #1D0DFh | RDM |
| #1C313h | FS? | #1D186h | CON |
| #1C360h | FC? | #1D2DCh | IDN |
| #1C399h | DEG | #1D392h | TRN |
| #1C3B4h | RAD | #1D407h | PUT |
| #1C3CFh | GRAD | #1D5DFh | PUTI |
| #1C3EAh | FIX | #1D7C6h | GET |
| #1C41Eh | SCI | #1D8C7h | GETI |
| #1C452h | ENG | #1DD06h | V→ |
| #1C486h | STD | #1DE66h | →V2 |
| #1C4A1h | FS?C | #1DEC2h | →V3 |
| #1C520h | FC?C | #1E04Ah | INDEP |
| #1C559h | BIN | #1E07Eh | PMIN |
| #1C574h | DEC | #1E09Eh | PMAX |
| #1C58Fh | HEX | #1E0BEh | AXES |
| #1C5AAh | OCT | #1E0E8h | CENTR |
| #1C5C5h | STWS | #1E126h | RES |
| #1C5FEh | RCWS | #1E150h | *H |
| #1C619h | RCLF | #1E170h | *W |
| #1C67Fh | STOF | #1E190h | DRAW |
| #1C783h | →LIST | #1E1ABh | AUTO |
| #1C79Eh | R→C | #1E1C6h | DRAX |
| #1C7CAh | RE | #1E1E1h | SCALE |
| #1C819h | IM | #1E201h | PDIM |
| #1C85Ch | SUB | #1E22Bh | DEPND |
| #1C8EAh | REPL | #1E25Fh | ERASE |
| #1C95Ah | LIST→ | #1E27Ah | PX→C |
| #1C98Eh | C→R | #1E29Ah | C→PX |
| #1C9B8h | SIZE | #1E2BAh | GRAPH |

| Adresses | Instructions | Adresses | Instructions |
|----------|--------------|----------|--------------|
| #1E2D5h | LABEL | #1EE6Eh | PRSTC |
| #1E2F0h | PVIEW | #1EE89h | PRST |
| #1E320h | PIXON | #1EEA4h | CR |
| #1E344h | PIXOFF | #1EEBFh | PRVAR |
| #1E36Eh | PIX? | #1EF43h | DELAY |
| #1E398h | LINE | #1EF63h | PRLCD |
| #1E3C2h | TLINE | #1F133h | RCEQ |
| #1E3ECh | BOX | #1F14Eh | STEQ |
| #1E416h | BLANK | #1F16Eh | ROOT |
| #1E436h | PICT | #1F2C9h | Σ |
| #1E456h | GOR | #1F500h | QUOTE |
| #1E4E4h | GXOR | #1F55Dh | APPLY |
| #1E572h | LCD→ | #1F9C4h | →Q |
| #1E58Dh | →LCD | #1F9E9h | →Q π |
| #1E5ADh | →GROB | #1FA59h | MATCH↑ |
| #1E5D2h | ARC | #1FA8Dh | MATCH↓ |
| #1E606h | TEXT | #1FAEBh | - |
| #1E621h | XRNG | #1FB5Dh | RATIO |
| #1E641h | YRNG | #1FB87h | DUP |
| #1E661h | FUNCTION | #1FBA2h | DUP2 |
| #1E681h | CONIC | #1FBBDh | SWAP |
| #1E6A1h | POLAR | #1FBD8h | DROP |
| #1E6C1h | PARAMETRIC | #1FBF3h | DROP2 |
| #1E6E1h | TRUTH | #1FC0Eh | ROT |
| #1E701h | SCATTER | #1FC29h | OVER |
| #1E721h | HISTOGRAM | #1FC44h | DEPTH |
| #1E741h | BAR | #1FC64h | DROPN |
| #1E761h | SAME | #1FC7Fh | DUPN |
| #1E783h | AND | #1FC9Ah | PICK |
| #1E809h | OR | #1FCB5h | ROLL |
| #1E88Fh | NOT | #1FCD0h | ROLLD |
| #1E8F6h | XOR | #1FCEBh | CLEAR |
| #1E972h | = = | #1FD0Bh | STO Σ |
| #1EA9Dh | ≠ | #1FD2Bh | CL Σ |
| #1EBBEh | < | #1FD46h | RCL Σ |
| #1EC5Dh | > | #1FD61h | $\Sigma+$ |
| #1ECFCh | ≤ | #1FD8Bh | $\Sigma-$ |
| #1ED9Bh | ≥ | #1FDA6h | N Σ |
| #1EE38h | OLDPRT | #1FDC1h | CORR |
| #1EE53h | PR1 | #1FDDCh | COV |

| Adresses | Instructions | Adresses | Instructions |
|----------|---------------|----------|--------------|
| #1FDF7h | Σx | #208F4h | INCR |
| #1FE12h | Σy | #209AAh | DECR |
| #1FE2Dh | Σx^2 | #20A15h | COLCT |
| #1FE48h | Σy^2 | #20A49h | EXPAN |
| #1FE63h | $\Sigma x*y$ | #20A7Dh | RULES |
| #1FE7Eh | MAX Σ | #20A93h | ISOL |
| #1FE99h | MEAN | #20AB3h | QUAD |
| #1FEB4h | MIN Σ | #20AD3h | SHOW |
| #1FECFh | SDEV | #20B20h | TAYLR |
| #1FEEAh | TOT | #20B40h | RCL |
| #1FF05h | VAR | #20CCDh | STO |
| #1FF20h | LR | #20D65h | DEFINE |
| #1FF7Ah | PREDV | #20EFEh | PURGE |
| #1FF9Ah | PREDY | #20FAAh | MEM |
| #1FFBAh | PREDX | #20FD9h | ORDER |
| #1FFDAh | XCOL | #210FCh | CLUSR |
| #1FFFAh | YCOL | #2115Dh | TMENU |
| #20020h | UTPC | #21196h | MENU |
| #2003Ah | UTPN | #211E1h | RCLMENU |
| #2005Ah | UTPF | #211FCh | PVARS |
| #2007Ah | UTPT | #2123Ah | PGDIR |
| #2009Ah | Σ COL | #2125Ah | ARCHIVE |
| #200C4h | SCL Σ | #2133Ch | RESTORE |
| #200F3h | Σ LINE | #2137Fh | MERGE |
| #2010Eh | BINS | #213D1h | FREE |
| #20133h | BARPLOT | #2142Dh | LIBS |
| #20167h | HISTPLOT | #21448h | ATTACH |
| #2018Ch | SCATRPLOT | #2147Ch | DETACH |
| #201B1h | LINFIT | #21E75h | XMIT |
| #201D6h | LOGFIT | #21E95h | SRECV |
| #201FBh | EXPFIT | #21EB5h | OPENIO |
| #20220h | PWRFIT | #21ED5h | CLOSEIO |
| #2025Eh | BESTFIT | #21EF0h | SEND |
| #202CEh | SINV | #21F24h | KGET |
| #2034Dh | SNEG | #21F62h | RECN |
| #20358h | STO- | #21F96h | RECV |
| #203CCh | SCONJ | #21FB6h | FINISH |
| #2044Bh | STO+ | #21FD1h | SERVER |
| #2060Ch | STO/ | #21FECh | CKSM |
| #20753h | STO* | #2200Ch | BAUD |

| Adresses | Instructions | Adresses | Instructions |
|----------|--------------------|----------|-----------------------|
| #2202Ch | PARITY | #230C3h | DO |
| #2204Ch | TRANSIO | #230EDh | UNTIL |
| #2206Ch | KERRM | #23103h | START |
| #22087h | BUFLEN | #231A0h | FOR |
| #220A2h | STIME | #2324Ch | NEXT |
| #220C2h | SBRK | #23380h | STEP |
| #220DDh | PKT | #233DFh | IFERR |
| #224CAh | INPUT | #23472h | HALT |
| #224F4h | ASN | #2361Eh | << |
| #22514h | STOKEYS | #23639h | >> |
| #22548h | DELKEYS | #23654h | BEGIN |
| #22586h | RCLKEYS | #23679h | END |
| #225BEh | →TAG | #23694h | END (structure WHILE) |
| #22633h | DTAG | #236B9h | END (structure DO) |
| #22EC3h | IF | #2371Fh | THEN (structure ERR) |
| #22EFAh | THEN | #2378Dh | CASE |
| #22FB5h | ELSE | #237A8h | THEN (structure CASE) |
| #22FD5h | END (structure IF) | #23813h | DIR |
| #23033h | WHILE | #23824h | PROMPT |
| #2305Dh | REPEAT | | |

Constantes



En RPL-système, l'emploi d'un objet suppose l'introduction de sa structure complète : en-tête, taille, quartets utiles, etc. La structure des objets a déjà été exposée dans le livre *HP-48 : faites vos jeux en assembleur (DUNOD)*. Toutefois, il est possible d'introduire très facilement certains objets. En effet, certaines adresses introduisent dans la pile quelques constantes usuelles dont la liste suit.

| Adresse | Type de la constante | Constante |
|---------|----------------------|-----------|
| #2A42Eh | Constante réelle | -9 |
| #2A419h | Constante réelle | -8 |
| #2A404h | Constante réelle | -7 |
| #2A3EFh | Constante réelle | -6 |
| #2A3DAh | Constante réelle | -5 |
| #2A3C5h | Constante réelle | -4 |
| #2A3B0h | Constante réelle | -3 |
| #2A39Bh | Constante réelle | -2 |
| #2A386h | Constante réelle | -1 |
| #2A2B4h | Constante réelle | 0 |
| #494B4h | Constante réelle | .1 |
| #650BDh | Constante réelle | .5 |
| #2A2C9h | Constante réelle | 1 |
| #2A2DEh | Constante réelle | 2 |
| #2A2F3h | Constante réelle | 3 |
| #2A308h | Constante réelle | 4 |
| #2A31Dh | Constante réelle | 5 |
| #2A332h | Constante réelle | 6 |
| #2A347h | Constante réelle | 7 |

| Adresse | Type de la constante | Constante |
|---------|----------------------|-----------|
| #2A35Ch | Constante réelle | 8 |
| #2A371h | Constante réelle | 9 |
| #650E7h | Constante réelle | 10 |
| #1CC03h | Constante réelle | 11 |
| #1CC1Dh | Constante réelle | 12 |
| #1CC37h | Constante réelle | 13 |
| #1CC51h | Constante réelle | 14 |
| #1CC85h | Constante réelle | 15 |
| #1CD3Ah | Constante réelle | 16 |
| #1CD54h | Constante réelle | 17 |
| #1CC6Bh | Constante réelle | 20 |
| #1CCA4h | Constante réelle | 21 |
| #1CCC3h | Constante réelle | 22 |
| #1CCE2h | Constante réelle | 23 |
| #1CD01h | Constante réelle | 24 |
| #1CD20h | Constante réelle | 25 |
| #1CD73h | Constante réelle | 26 |
| #1CD8Dh | Constante réelle | 27 |
| #415F1h | Constante réelle | 100 |
| #650FCh | Constante réelle | 180 |
| #65126h | Constante réelle | 360 |
| #2A443h | Constante réelle | π |
| #650A8h | Constante réelle | e |
| #5196Ah | Constante complexe | (-1, 0) |
| #524AFh | Constante complexe | (0, 0) |
| #524F7h | Constante complexe | (1, 0) |
| #6542Ch | Caractère | " |
| #65433h | Caractère | # |
| #65663h | Caractère | (|
| #65678h | Caractère |) |
| #6543Ah | Caractère | * |
| #65441h | Caractère | + |
| #65448h | Caractère | , |
| #6544Fh | Caractère | - |
| #65456h | Caractère | . |
| #6545Dh | Caractère | / |
| #654AAh | Caractère | : |
| #654B1h | Caractère | " ; " |
| #654B8h | Caractère | < |
| #654BFh | Caractère | = |
| #654C6h | Caractère | > |
| #65694h | Caractère | [|

| Adresse | Type de la constante | Constante |
|---------|----------------------|-----------|
| #6569Bh | Caractère |] |
| #6568Dh | Caractère | _ |
| #656A2h | Caractère | { |
| #656A9h | Caractère | } |
| #65464h | Caractère | 0 |
| #6546Bh | Caractère | 1 |
| #65472h | Caractère | 2 |
| #65479h | Caractère | 3 |
| #65480h | Caractère | 4 |
| #65487h | Caractère | 5 |
| #6548Eh | Caractère | 6 |
| #65495h | Caractère | 7 |
| #6549Ch | Caractère | 8 |
| #654A3h | Caractère | 9 |
| #654CDh | Caractère | A |
| #65583h | Caractère | a |
| #654D4h | Caractère | B |
| #6558Ah | Caractère | b |
| #654DBh | Caractère | C |
| #65591h | Caractère | c |
| #654E2h | Caractère | D |
| #65598h | Caractère | d |
| #654E9h | Caractère | E |
| #6559Fh | Caractère | e |
| #654F0h | Caractère | F |
| #655A6h | Caractère | f |
| #654F7h | Caractère | G |
| #655ADh | Caractère | g |
| #654FEh | Caractère | H |
| #655B4h | Caractère | h |
| #65505h | Caractère | I |
| #655BBh | Caractère | i |
| #6550Ch | Caractère | J |
| #655C2h | Caractère | j |
| #65513h | Caractère | K |
| #655C9h | Caractère | k |
| #6551Ah | Caractère | L |
| #655D0h | Caractère | l |
| #65521h | Caractère | M |
| #655D7h | Caractère | m |
| #65528h | Caractère | N |
| #655DEh | Caractère | n |
| #6552Fh | Caractère | O |

| Adresse | Type de la constante | Constante |
|---------|----------------------|--|
| #655E5h | Caractère | o |
| #65536h | Caractère | P |
| #655ECh | Caractère | p |
| #6553Dh | Caractère | Q |
| #655F3h | Caractère | q |
| #65544h | Caractère | R |
| #655FAh | Caractère | r |
| #6554Bh | Caractère | S |
| #65601h | Caractère | s |
| #65552h | Caractère | T |
| #65608h | Caractère | t |
| #65559h | Caractère | U |
| #6560Fh | Caractère | u |
| #65560h | Caractère | V |
| #65616h | Caractère | v |
| #65567h | Caractère | W |
| #6561Dh | Caractère | w |
| #6556Eh | Caractère | X |
| #65624h | Caractère | x |
| #65575h | Caractère | Y |
| #6562Bh | Caractère | Y |
| #6557Ch | Caractère | Z |
| #65632h | Caractère | z |
| #13D8Ch | Objet graphique | Curseur en mode insertion (GROB 6x10) |
| #13DB4h | Objet graphique | Curseur en mode écrasement (GROB 6x10) |
| #5053Ch | Objet graphique | Réticule (GROB 5x5) |
| #5055Ah | Objet graphique | Marque (GROB 5x5) |
| #66EA5h | Objet graphique | Cadre blanc et fond noir (GROB 6x10) |
| #66ECDh | Objet graphique | Cadre noir et fond blanc (GROB 6x8) |
| #66EF1h | Objet graphique | Cadre noir et fond blanc (GROB 6x4) |

Calculs et manipulation des nombres



Les instructions de calcul classiques du RPL-utilisateur sont bien sûr accessibles en RPL-système. Cependant, l'adresse (utilisable avec SYSEVAL) de l'instruction varie en fonction du type des arguments ou de celui du résultat. Le symbole → précède les arguments alors que le symbole ← précède les résultats. 1 : et 2 : représentent les niveaux de la pile.

| Fonction | Arguments et résultats dans la pile | Adresse |
|----------|--|---------|
| % | → 2: réel 1: réel ← 2: 1: réel | #2A9C9h |
| * | → 2: entier sys. 1: entier sys. ← 2: 1: entier sys. | #03EC2h |
| * | → 2: entier bin. 1: entier bin. ← 2: 1: entier bin. | #53ED3h |
| * | → 2: réel 1: entier bin. ← 2: 1: entier bin. | #542D1h |
| * | → 2: entier bin. 1: réel ← 2: 1: entier bin. | #542EAh |

| Fonction | Arguments et résultats dans la pile | Adresse |
|-----------------|---|----------------|
| * | → 2: réel 1: réel ← 2: 1: réel | #2A9BCh |
| * | → 2: complexe 1: complexe ← 2: 1: complexe | #51D88h |
| * | → 2: réel long 1: réel long ← 2: 1: réel long | #2A99Ah |
| * | → 2: objet unité 1: objet unité ← 2: 1: objet unité | #0F792h |
| + | → 2: entier sys. 1: entier sys. ← 2: 1: entier sys. | #03DBCh |
| + | → 2: entier bin. 1: entier bin. ← 2: 1: entier bin. | #53EA0h |
| + | → 2: réel 1: entier bin. ← 2: 1: entier bin. | #54330h |
| + | → 2: entier bin. 1: réel ← 2: 1: entier bin. | #54349h |
| + | → 2: réel 1: réel ← 2: 1: réel | #2A974h |
| + | → 2: complexe 1: complexe ← 2: 1: complexe | #51C16h |
| + | → 2: réel long 1: réel long ← 2: 1: réel long | #2A943h |
| - | → 2: entier sys. 1: entier sys. ← 2: 1: entier sys. | #03DE0h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|------------|---|---------|
| - | → 2: entier bin. 1: entier bin. ← 2: 1: entier bin. | #53EB0h |
| - | → 2: réel 1: entier bin. ← 2: 1: entier bin. | #542FEh |
| - | → 2: entier bin. 1: réel ← 2: 1: entier bin. | #5431Ch |
| - | → 2: réel 1: réel ← 2: 1: réel | #2A981h |
| - | → 2: complexe 1: complexe ← 2: 1: complexe | #51CFCh |
| - | → 2: réel long 1: réel long ← 2: 1: réel long | #2A94Fh |
| - | → 2: réel 1: réel ← 2: 1: réel long | #2A95Bh |
| - | → 2: objet unité 1: objet unité ← 2: 1: objet unité | #0F774h |
| ABS | → 1: réel ← 1: réel | #2A900h |
| ABS | → 1: complexe ← 1: complexe | #52062h |
| ABS | → 1: réel long ← 1: réel long | #2A8F0h |
| ABS | → 1: objet unité ← 1: objet unité | #0F5FCh |

| Fonction | Arguments et résultats dans la pile | Adresse |
|-----------------|--|----------------|
| ACOS | → 1: réel ← 1: réel | #2ACF1h |
| ACOS | → 1: complexe ← 1: complexe | #52863h |
| ACOS | → 1: réel long ← 1: réel long | #2AD08h |
| ACOSH | → 1: réel ← 1: réel | #2AE13h |
| ACOSH | → 1: complexe ← 1: complexe | #52836h |
| ALOG | → 1: réel ← 1: réel | #2ABBAh |
| ALOG | → 1: complexe ← 1: complexe | #52305h |
| AND | → 2: entier bin. 1: entier bin. ← 2: 1: entier bin. | #53D04h |
| AND | → 2: entier sys. 1: entier sys. ← 2: 1: entier sys. | #03EB1h |
| ARG | → 1: complexe ← 1: complexe | #52099h |
| ASIN | → 1: réel ← 1: réel | #2ACC1h |
| ASIN | → 1: complexe ← 1: complexe | #52804h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|--------------|---|---------|
| ASIN | → 1: réel long ← 1: réel long | #2ACD8h |
| ASINH | → 1: réel ← 1: réel | #2AE00h |
| ASINH | → 1: complexe ← 1: complexe | #5281Dh |
| ASR | → 1: entier bin. ← 1: entier bin. | #53E65h |
| ATAN | → 1: réel ← 1: réel | #2AD21h |
| ATAN | → 1: complexe ← 1: complexe | #52675h |
| ATANH | → 1: réel ← 1: réel | #2AE26h |
| ATANH | → 1: complexe ← 1: complexe | #527EBh |
| CEIL | → 1: réel ← 1: réel | #2AF73h |
| COMB | → 2: réel 1: réel ← 2: 1: réel | #2AE62h |
| CONJ | → 1: complexe ← 1: complexe | #51BB2h |
| COS | → 1: réel ← 1: réel | #2AC40h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|-----------------|--|----------------|
| COS | → 1: complexe ← 1: complexe | #52571h |
| COS | → 1: réel long ← 1: réel long | #2AC78h |
| COS | → 1: objet unité ← 1: objet unité | #0F660h |
| COSH | → 1: réel ← 1: réel | #2ADDAh |
| COSH | → 1: complexe ← 1: complexe | #52648h |
| COSH | → 1: réel long ← 1: réel long | #2ADC7h |
| D→R | → 1: réel ← 1: réel | #2A622h |
| EXP | → 1: réel ← 1: réel | #2AB2Fh |
| EXP | → 1: complexe ← 1: complexe | #52193h |
| EXP | → 1: réel long ← 1: réel long | #2AB1Ch |
| EXPM | → 1: réel ← 1: réel | #2AB42h |
| FACT | → 1: réel ← 1: réel | #2B0C4h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|--------------|--------------------------------------|----------------|
| FACT | → 1: réel ← 1: réel | #2AE4Ch |
| FLOOR | → 1: réel ← 1: réel | #2AF86h |
| FP | → 1: réel ← 1: réel | #2AF4Dh |
| FP | → 1: objet unité ← 1: objet unité | #0FD0Eh |
| IM | → 1: complexe ← 1: complexe | #519B7h |
| INV | → 1: réel ← 1: réel | #2AAAFh |
| INV | → 1: complexe ← 1: complexe | #51EFAh |
| INV | → 1: réel long ← 1: réel long | #2AA92h |
| IP | → 1: réel ← 1: réel | #2AF60h |
| LN | → 1: réel ← 1: réel | #2AB6Eh |
| LN | → 1: complexe ← 1: complexe | #521E3h |
| LN | → 1: réel long ← 1: réel long | #2AB5Bh |

| Fonction | Arguments et résultats dans la pile | Adresse |
|-----------------|---|----------------|
| LNP1 | → 1: réel ← 1: réel | #2ABA7h |
| LNP1 | → 1: réel long ← 1: réel long | #2AB94h |
| LOG | → 1: réel ← 1: réel | #2AB81h |
| LOG | → 1: complexe ← 1: complexe | #522BFh |
| MANT | → 1: réel ← 1: réel | #2A930h |
| MAX | → 2: entier sys. 1: entier sys. ← 2: 1: entier sys. | #624C6h |
| MAX | → 2: réel 1: réel ← 2: 1: réel | #2A6F5h |
| MAX | → 2: réel long 1: réel long ← 2: 1: réel long | #2A6DCh |
| MAX | → 2: objet unité 1: objet unité ← 2: 1: objet unité | #0FB6Fh |
| MIN | → 2: entier sys. 1: entier sys. ← 2: 1: entier sys. | #624BAh |
| MIN | → 2: réel 1: réel ← 2: 1: réel | #2A70Eh |
| NEG | → 1: réel ← 1: réel | #2A920h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|-------------|--|----------------|
| NEG | → 1: complexe ← 1: complexe | #51B70h |
| NEG | → 1: réel long ← 1: réel long | #2A910h |
| NOT | → 1: entier bin. ← 1: entier bin. | #53D4Eh |
| OR | → 2: entier bin. 1: entier bin. ← 2: 1: entier bin. | #53D15h |
| P→R | → 2: réel 1: réel ← 2: réel 1: réel | #2B4BBh |
| PERM | → 2: réel 1: réel ← 2: 1: réel | #2AE75h |
| R→D | → 1: réel ← 1: réel | #2A655h |
| RAND | | #2AFC2h |
| RDZ | → 1: réel ← 1: réel | #2B044h |
| RE | → 1: complexe ← 1: complexe | #519A3h |
| RLB | → 1: entier bin. ← 1: entier bin. | #53E3Bh |
| RR | → 1: entier bin. ← 1: entier bin. | #53DA4h |
| RRB | → 1: entier bin. ← 1: entier bin. | #53DE1h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|-----------------|--|----------------|
| SIGN | → 1: réel ← 1: réel | #2A8D7h |
| SIGN | → 1: complexe ← 1: complexe | #520CBh |
| SIN | → 1: réel ← 1: réel | #2ABEFh |
| SIN | → 1: complexe ← 1: complexe | #52530h |
| SIN | → 1: réel long ← 1: réel long | #2AC27h |
| SIN | → 1: objet unité ← 1: objet unité | #0F62Eh |
| SINH | → 1: réel ← 1: réel | #2ADAEh |
| SINH | → 1: complexe ← 1: complexe | #5262Fh |
| SINH | → 1: réel long ← 1: réel long | #2AD95h |
| SL | → 1: entier bin. ← 1: entier bin. | #53D5Eh |
| SLB | → 1: entier bin. ← 1: entier bin. | #53D6Eh |
| SR | → 1: entier bin. ← 1: entier bin. | #53D81h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|------------------|--|----------------|
| SRB | → 1: entier bin. ← 1: entier bin. | #53D91h |
| TAN | → 1: réel ← 1: réel | #2AC91h |
| TAN | → 1: complexe ← 1: complexe | #525B7h |
| TAN | → 1: réel long ← 1: réel long | #2ACA8h |
| TAN | → 1: objet unité ← 1: objet unité | #0F674h |
| TANH | → 1: réel ← 1: réel | #2ADEDh |
| TANH | → 1: complexe ← 1: complexe | #5265Ch |
| Test = | → 2: réel 1: réel ← 2: 1: booléen | #2A8C1h |
| Test ≠ | → 2: réel 1: réel ← 2: 1: booléen | #2A8CCh |
| Test ≤ | → 2: réel 1: réel ← 2: 1: booléen | #2A8B6h |
| Test ≤ | → 2: réel long 1: réel long ← 2: 1: booléen | #2A8ABh |
| Test < | → 2: réel 1: réel ← 2: 1: booléen | #2A871h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|-------------------|--|----------------|
| Test < | → 2: réel long 1: réel long ← 2: 1: booléen | #2A81Fh |
| Test ≠1 | → 1: entier sys. ← 1: booléen | #622B6h |
| Test ≠0 | → 1: entier sys. ← 1: booléen | #03CC7h |
| Test =0 | → 1: entier sys. ← 1: booléen | #03CA6h |
| Test =1 | → 1: entier sys. ← 1: booléen | #622A7h |
| Test ≥ | → 2: réel 1: réel ← 2: 1: booléen | #2A8A0h |
| Test ≥ | → 2: réel long 1: réel long ← 2: 1: booléen | #2A895h |
| Test > | → 2: réel 1: réel ← 2: 1: booléen | #2A88Ah |
| Test > | → 2: réel long 1: réel long ← 2: 1: booléen | #2A87Fh |
| Test ≠0 | → 1: réel ← 1: booléen | #2A7CFh |
| Test =0 | → 1: réel ← 1: booléen | #2A76Bh |
| Test <0 | → 1: réel ← 1: booléen | #2A738h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|-------------------|---|---------|
| Test >0 | → 1: réel ← 1: booléen | #2A799h |
| Test ≥0 | → 1: réel ← 1: booléen | #2A7F7h |
| Test ≤0 | → 1: réel long ← 1: booléen | #2A80Bh |
| Test ≠0 | → 1: réel long ← 1: booléen | #2A7BBh |
| Test =0 | → 1: réel long ← 1: booléen | #2A75Ah |
| Test <0 | → 1: réel long ← 1: booléen | #2A727h |
| Test >0 | → 1: réel long ← 1: booléen | #2A788h |
| Test ≥0 | → 1: réel long ← 1: booléen | #2A7E3h |
| Test = | → 2: entier sys. 1: entier sys. ← 2: 1: booléen | #03D19h |
| Test ≠ | → 2: entier sys. 1: entier sys. ← 2: 1: booléen | #03D4Eh |
| Test ≠ | → 2: entier bin. 1: entier bin. ← 2: 1: réel renvoie 0 ou 1 | #544ECh |
| Test < | → 2: entier sys. 1: entier sys. ← 2: 1: booléen | #03CE4h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|------------------|---|----------------|
| Test ≤ | → 2: entier bin. 1: entier bin. ← 2: 1: réel renvoie 0 ou 1 | #5453Fh |
| Test < | → 2: entier bin. 1: entier bin. ← 2: 1: réel renvoie 0 ou 1 | #54552h |
| Test > | → 2: entier sys. 1: entier sys. ← 2: 1: booléen | #03D83h |
| Test ≥ | → 2: entier bin. 1: entier bin. ← 2: 1: réel renvoie 0 ou 1 | #5452Ch |
| Test > | → 2: entier bin. 1: entier bin. ← 2: 1: réel renvoie 0 ou 1 | #54500h |
| XOR | → 2: entier bin. 1: entier bin. ← 2: 1: entier bin. | #53D26h |
| XPON | → 1: réel ← 1: réel | #2AE39h |
| XROOT | → 2: réel 1: réel ← 2: 1: réel | #2AA81h |
| ^ | → 2: réel 1: réel ← 2: 1: réel | #2AA70h |
| ^ | → 2: complexe 1: complexe ← 2: 1: complexe | #52374h |
| ^ | → 2: complexe 1: réel ← 2: 1: complexe | #52360h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|----------|---|---------|
| ^ | → 2: réel long 1: réel long ← 2: 1: réel long | #2AA5Fh |
| √ | → 1: réel ← 1: réel | #2AB09h |
| √ | → 1: complexe ← 1: complexe | #52107h |
| √ | → 1: réel long ← 1: réel long | #2AAEAh |
| √ | → 1: réel ← 1: réel long | #2AAF6h |
| √ | → 1: objet unité ← 1: objet unité | #0F92Ch |
| ÷ | → 2: entier bin. 1: entier bin. ← 2: 1: entier bin. | #53F05h |
| ÷ | → 2: réel 1: entier bin. ← 2: 1: entier bin. | #5429Fh |
| ÷ | → 2: entier bin. 1: réel ← 2: 1: entier bin. | #542BDh |
| ÷ | → 2: réel 1: réel ← 2: 1: réel | #2A9FEh |
| ÷ | → 2: complexe 1: complexe ← 2: 1: complexe | #51EC8h |
| ÷ | → 2: réel long 1: réel long ← 2: 1: réel long | #2A9E8h |

Pile et ligne de commande



Les adresses SYSEVAL qui suivent correspondent à des opérations relatives à la pile ou à la ligne de commande. Quand une adresse correspond à une instruction RPL ou à une séquence d'instructions RPL, celle-ci est précisée (par exemple, #630C9h correspond à 4 **PICK OVER** en RPL-utilisateur). Dans les autres cas, une courte explication est fournie.

| Fonction | Adresse |
|---------------|----------|
| 3 DROPN | #60F4Bh |
| 3 PICK | #611FEh |
| 3 PICK 3 PICK | #63C68h |
| 3 PICK OVER | #630B5h |
| 3 PICK SWAP | #62EDFh |
| 3 ROLLD | #60FAC h |
| 3 ROLLD DROP | #6284Bh |
| 3 ROLLD DUP | #62CF5h |
| 3 ROLLD OVER | #6308Dh |
| 4 DROPN | #60F7Eh |
| 4 PICK | #6121Ch |
| 4 PICK 4 PICK | #63FBAh |
| 4 PICK OVER | #630C9h |
| 4 PICK SWAP | #62EF3h |
| 4 ROLL | #60FBBh |

| Fonction | Adresse |
|--|----------------|
| 4 ROLL 4 ROLL | #62001h |
| 4 ROLL OVER | #630A1h |
| 4 ROLL ROT | #63001h |
| 4 ROLL SWAP | #62ECBh |
| 4 ROLLD | #6109Eh |
| 4 ROLLD DUP | #62D09h |
| 4 ROLLD ROT | #63015h |
| 5 DROPN | #60F72h |
| 5 PICK | #6123Ah |
| 5 ROLL | #60FD8h |
| 5 ROLLD | #610C4h |
| 6 DROPN | #60F66h |
| 6 PICK | #6125Eh |
| 6 ROLL | #61002h |
| 6 ROLLD | #610FAh |
| 7 DROPN | #60F54h |
| 7 PICK | #61282h |
| 7 ROLL | #6106Bh |
| 7 ROLLD | #62BC4h |
| 8 PICK | #612A9h |
| 8 ROLL | #6103Ch |
| 8 ROLLD | #63119h |
| Activation du mode PRG ou ALG (selon le type de l'objet placé au niveau 1) | #42EC7h |
| Annulation du mode alphabétique | #40D39h |
| DROP | #03244h |
| DROP DUP | #627A7h |
| DROP le numéro du niveau concerné est précisé par un entier système | #62F89h |
| DROP OVER | #63029h |
| DROP ROT | #62FC5h |
| DROP SWAP | #6270Ch |
| DROP2 | #03258h |

| Fonction | Arguments et résultats dans la pile | Adresse |
|--------------------|--|---------|
| DROPN | le numéro du niveau concerné est précisé par un entier système | #0326Eh |
| DUP | | #03188h |
| DUP 3 PICK | | #611F9h |
| DUP 4 ROLLD | | #61099h |
| DUP DUP | | #62CB9h |
| DUP ROT | | #62FB1h |
| DUP2 | | #031ACh |
| | Duplication des niveaux 1 et 2 puis 5 ROLL | #63C40h |
| DUPN | le numéro du niveau concerné est précisé par un entier système | #031D9h |
| | Mode ALG pour la ligne de commande | #53976h |
| | Mode PRG pour la ligne de commande | #11533h |
| OVER | | #032C2h |
| OVER 5 PICK | | #63C90h |
| OVER DUP | | #62CCDh |
| OVER SWAP | | #62D31h |
| | Permutation des niveaux 1 et 3 | #60F33h |
| | Permutation des niveaux 2 et 3 | #60EE7h |
| PICK | le numéro du niveau concerné est précisé par un entier système | #032E2h |
| | Rappel de la pile sauvegardée | #61F8Fh |
| ROLL | le numéro du niveau concerné est précisé par un entier système | #03325h |
| ROLLD | le numéro du niveau concerné est précisé par un entier système | #0339Eh |
| ROT | | #03295h |
| ROT DUP | | #62775h |
| ROT OVER | | #62CA5h |
| ROT | puis duplication des niveaux 1 et 2 | #62C7Dh |
| | Sauvegarde de la pile | #61D41h |
| | Suppression des niveaux 1 et 3 | #62726h |
| | Suppression des niveaux 2 à 4 | #6113Ch |

| Fonction | Adresse |
|---|----------------|
| Suppression des niveaux 2 et 3 | #6112Ah |
| Suppression du niveau 2 | #60F9Bh |
| Suppression du niveau 2 puis DUP | #62830h |
| Suppression du niveau 3 | #60F21h |
| Suppression du niveau 3 puis SWAP | #60F0Eh |
| Suppression du niveau 4 | #62864h |
| Suppression du niveau 5 | #62880h |
| SWAP | #03223h |
| SWAP 3 PICK | #63C54h |
| SWAP 4 PICK | #63C7Ch |
| SWAP 4 ROLL | #63C2Ch |
| SWAP DUP | #62747h |
| SWAP OVER | #61380h |
| SWAP puis duplication des niveaux 1 et 2 | #6386Ch |
| Verrouillage du clavier en mode alphabétique | #40D25h |

Listes et chaînes de caractères



Les adresses SYSEVAL qui suivent correspondent à des opérations relatives aux listes ou aux chaînes de caractères. Quand une adresse correspond à une instruction RPL ou à une séquence d'instructions RPL, celle-ci est précisée. Une courte explication précise l'opération effectuée ainsi que la nature des arguments et le type des résultats.

| Fonction | Adresse |
|---|---------|
| Ajout d'un espace à la fin de la chaîne donnée en argument | #62BB0h |
| Ajout d'un objet (ou d'un caractère) à la fin d'une liste (ou d'une chaîne) | #052FAh |
| Ajout du caractère LF (line feed, passage à la ligne suivante) à la fin de la chaîne donnée en argument | #63191h |
| Ajout du caractère placé au niveau 1 à la chaîne placée au niveau 2 | #052EEh |
| AND bit à bit sur deux chaînes de mêmes longueurs | #18873h |
| CHR prend un réel en argument | #140F1h |
| Concaténation de deux listes ou de deux chaînes de caractères | #0521Fh |
| DUP puis longueur d'une chaîne ou d'une liste exprimée sous la forme d'un entier système | #63231h |
| DUP puis longueur du chaîne exprimée en octets et renvoyée sous la forme d'un entier système | #627BBh |

| Fonction | Adresse |
|--|----------------|
| Longueur d'une chaîne ou d'une liste exprimée sous la forme d'un entier système | #0567Bh |
| Longueur du chaîne exprimée en octets et renvoyée sous la forme d'un entier système | #05636h |
| Longueur du chaîne exprimée en quartets et renvoyée sous la forme d'un entier système | #05616h |
| OR bit à bit sur deux chaînes de mêmes longueurs | #18887h |
| OVER puis concaténation des chaînes placées aux niveaux 1 à 3 | #6910Ch |
| OVER puis longueur du chaîne exprimée en octets et renvoyée sous la forme d'un entier système | #05622h |
| SWAP puis concaténation de deux chaînes de caractères | #622EFh |
| XOR bit à bit sur deux chaînes de mêmes longueurs | #1889Bh |

Gestion des types



Les adresses qui suivent sont utilisables avec SYSEVAL. Le symbole → précède les arguments alors que le symbole ← précède les résultats. 1: et 2: représentent les niveaux de la pile. Ces adresses permettent la conversion des types ou des tests relatifs aux types. Les tests renvoient un booléen (TRUE ou FALSE). Les instructions équivalentes du RPL-utilisateur sont communiquées quand elles existent, par exemple, **TYPE**.

| Fonction | Adresse |
|--|---------|
| → STR prend un entier binaire en argument et ôte le suffixe précisant la base → 1: entier binaire ← 1: chaîne | #540BBh |
| → STR prend un réel en argument → 1: réel ← 1: chaîne | #162B8h |
| → STR sans vérification de la présence d'un argument | #14088h |
| Conversion d'un caractère en chaîne → 1: caractère ← 1: chaîne | #6475Ch |
| Conversion d'un caractère en entier système → 1: caractère ← 1: entier système | #05A51h |
| Conversion d'un complexe long en complexe → 1: complexe long ← 1: complexe | #519F8h |

| Fonction | Adresse |
|---|----------------|
| Conversion d'un entier binaire en entier système → 1: entier binaire ← 1: entier système | #05A03h |
| Conversion d'un entier binaire en réel → 1: entier binaire ← 1: réel | #5435Dh |
| Conversion d'un entier système en caractère → 1: entier système ← 1: caractère | #05A75h |
| Conversion d'un entier système en entier binaire → 1: entier système ← 1: entier binaire | #059CCh |
| Conversion d'un entier système en réel → 1: entier système ← 1: réel | #18DBFh |
| Conversion d'un entier système en réel long → 1: entier système ← 1: réel long | #63B96h |
| Conversion d'un nom en chaîne → 1: nom ← 1: chaîne | #05BE9h |
| Conversion d'un nom en nom local → 1: nom ← 1: nom local | #05AEDh |
| Conversion d'un nom en objet unité (le nom devient l'unité) → 1: nom ← 1: objet unité | #28AC9h |
| Conversion d'un nom quelconque en nom global → 1: nom ← 1: nom global | #05B01h |
| Conversion d'un objet-unité en chaîne → 1: objet-unité ← 1: chaîne | #0F218h |
| Conversion d'un réel (valeur absolue arrondie à l'entier le plus proche) en un entier système → 1: réel ← 1: entier système | #18CD7h |

| Fonction | Adresse |
|---|---------|
| Conversion d'un réel en complexe | #51A37h |
| → 1: réel | |
| ← 1: complexe | |
| Conversion d'un réel en entier binaire | #543F9h |
| → 1: réel | |
| ← 1: entier binaire | |
| Conversion d'un réel en réel long | #2A5C1h |
| → 1: réel | |
| ← 1: réel long | |
| Conversion d'un réel long en réel | #2A5B0h |
| → 1: réel long | |
| ← 1: réel | |
| Conversion d'une chaîne en nom global | #05B15h |
| → 1: chaîne | |
| ← 1: nom global | |
| Conversion de la partie réelle d'un réel en entier système | #2EC11h |
| → 1: réel | |
| ← 1: entier système | |
| Conversion des parties réelle et imaginaire d'un complexe en deux réels longs | #519CBh |
| → 2: 1: complexe | |
| ← 2: réel long 1: réel long | |
| Conversion des réels des niveaux 1 et 2 en deux entiers système | #194F7h |
| → 2: réel 1: réel | |
| ← 2: entier sys. 1: entier sys. | |
| Conversion du premier caractère d'une chaîne en caractère | #050EDh |
| → 1: chaîne | |
| ← 1: caractère | |
| Création d'un complexe à partir de deux réels longs | #51A07h |
| → 2: réel long 1: réel long | |
| ← 2: 1: complexe | |
| Création d'une chaîne contenant la valeur en décimal d'un entier système | #167E4h |
| → 1: entier système | |
| ← 1: chaîne | |

| Fonction | Adresse |
|---|----------------|
| DUP puis test : un objet est-il du type caractère ? → 2: 1: objet ← 2: objet 1: booléen | #62020h |
| DUP puis test : un objet est-il du type chaîne de caractères ? → 2: 1: objet ← 2: objet 1: booléen | #62154h |
| DUP puis test : un objet est-il du type entier binaire ? → 2: 1: objet ← 2: objet 1: booléen | #6213Fh |
| DUP puis test : un objet est-il du type entier système ? → 2: 1: objet ← 2: objet 1: booléen | #6212Ah |
| DUP puis test : un objet est-il du type expression symbolique ? → 2: 1: objet ← 2: objet 1: booléen | #621D2h |
| DUP puis test : un objet est-il du type liste ? → 2: 1: objet ← 2: objet 1: booléen | #62211h |
| DUP puis test : un objet est-il du type nom global ? → 2: 1: objet ← 2: objet 1: booléen | #62035h |
| DUP puis test : un objet est-il du type nom local ? → 2: 1: objet ← 2: objet 1: booléen | #62115h |
| DUP puis test : un objet est-il du type nom XLIB ? → 2: 1: objet ← 2: objet 1: booléen | #621A8h |
| DUP puis test : un objet est-il du type nombre complexe ? → 2: 1: objet ← 2: objet 1: booléen | #6217Eh |
| DUP puis test : un objet est-il du type nombre réel ? → 2: 1: objet ← 2: objet 1: booléen | #62169h |
| DUP puis test : un objet est-il du type objet graphique ? → 2: 1: objet ← 2: objet 1: booléen | #621FCh |

| Fonction | Adresse |
|---|----------------|
| Un objet est-il du type nom local ? → 1: objet ← 1: booléen | #6211Ah |
| Un objet est-il du type nom XLIB ? → 1: objet ← 1: booléen | #621ADh |
| Un objet est-il du type nombre complexe ? → 1: objet ← 1: booléen | #62183h |
| Un objet est-il du type nombre réel ? → 1: objet ← 1: booléen | #6216Eh |
| Un objet est-il du type objet graphique ? → 1: objet ← 1: booléen | #62201h |
| Un objet est-il du type objet taggué ? → 1: objet ← 1: booléen | #6222Bh |
| Un objet est-il du type objet-unité ? → 1: objet ← 1: booléen | #6204Fh |
| Un objet est-il du type programme RPL ? → 1: objet ← 1: booléen | #621ECh |
| Un objet est-il du type répertoire ? → 1: objet ← 1: booléen | #621C2h |
| Un objet est-il du type tableau ? → 1: objet ← 1: booléen | #62198h |
| Un objet est-il du type tableau complexe → 1: objet ← 1: booléen | #62256h |

Manipulation des booléens



Les adresses qui suivent sont utilisables avec SYSEVAL. Le symbole → précède les arguments alors que le symbole ← précède les résultats. 1 : et 2 : représentent les niveaux de la pile. Ces adresses placent des booléens (TRUE ou FALSE) dans la pile ou permettent des opérations logiques entre booléens. Les instructions équivalentes du RPL-utilisateur sont communiquées quand elles existent, par exemple, **DROP**.

| Fonction | Adresse |
|--|---------|
| AND | #03B46h |
| → 2 : booléen 1 : booléen | |
| ← 2 : 1 : booléen | |
| DROP puis introduction de FALSE | #6210Ch |
| DROP puis introduction de TRUE | #62103h |
| Introduction de FALSE aux niveaux 1 et 2 | #2F934h |
| Introduction de FALSE dans la pile | #03AC0h |
| Introduction de FALSE puis de TRUE | #6350Bh |
| Introduction de TRUE aux niveaux 1 et 2 | #0BBEDh |
| Introduction de TRUE dans la pile | #03A81h |
| Introduction de TRUE puis de FALSE | #634F7h |
| NOR | #635B0h |
| → 2 : booléen 1 : booléen | |
| ← 2 : 1 : booléen | |

| Fonction | Adresse |
|--|----------------|
| NOT → 1:booléen ← 1:booléen | #03AF2h |
| OR → 2: booléen ← 2: 1: booléen 1: booléen | #03B75h |
| SWAP puis introduction de TRUE | #4F1D8h |
| XOR → 2: booléen ← 2: 1: booléen 1: booléen | #03ADAh |

Variables, répertoires et mémoire



Les adresses qui suivent sont utilisables avec SYSEVAL. Elles correspondent à des instructions manipulant les variables (locales et/ou globales), les répertoires, les bibliothèques et la mémoire. Les instructions équivalentes du RPL-utilisateur sont communiquées quand elles existent, par exemple, **BYTES**.

| Fonction | Adresse |
|--|---------|
| Affichage de l'écran "Try to recover memory ?" | #0CBA2h |
| ARCHIVE prend en argument un objet-taggué | #21273h |
| ATTACH prend un entier système en argument | #21C6Fh |
| ATTACH prend un entier système en argument et attache une bibliothèque à HOME | #07709h |
| BYTES considère l'objet contenu dans une variable quand le nom de ladite variable est donné en argument | #1A265h |
| BYTES ne permet pas d'utiliser le nom d'une variable à la place du contenu de ladite variable | #1A1FCh |
| CLEAR (suppression du contenu de la pile) puis récupération de l'espace occupé en mémoire par des objets devenus inutiles (<i>garbage collection</i>) | #18355h |
| CRDIR le nom donné en argument ne doit pas correspondre à un répertoire existant non vide | #184E1h |

| Fonction | Adresse |
|--|---------|
| Création d'un nombre de variables locales précisé par un entier au niveau 2 et portant toutes le nom local précisé au niveau 1 | #61CE9h |
| Création d'une variable locale ayant pour nom le nom vide et dont le contenu est celui du niveau 1 | #634CFh |
| DUP puis stockage de l'objet du niveau 1 dans la variable locale n°04 | #61610h |
| Duplication du dernier lot de variables locales (conservation de ces valeurs malgré la destruction des originaux par la HP-48) | #61745h |
| Effacement de la pile mémorisée et initialisation des variables locales existantes | #17D46h |
| Espace libre en mémoire renvoyé sous la forme d'un entier-système (pas de <i>garbage collection</i>) | #05F61h |
| Espace occupé (en quartets) par l'objet placé au niveau 1 de la pile | #05902h |
| EVAL prend en argument un nom XLIB | #02FEFh |
| EVAL prend en argument un objet-taggué | #217C7h |
| EVAL prend en argument une expression, une liste ou un programme RPL | #06F9Fh |
| EVAL s'applique à un objet qui n'est ni une liste, ni un objet-taggué, ni une expression | #06F8Eh |
| <i>Garbage collection</i> (récupération de l'espace occupé par les objets devenus inutiles) | #05F42h |
| HOME | #08D92h |
| LASTARG indépendant du flag -55 | #1A631h |
| Le contenu du répertoire courant est placé dans la pile | #08D5Ah |
| Le répertoire caché devient le répertoire courant | #640BEh |
| Le répertoire donné en argument devient le répertoire courant | #08D08h |
| NEWOB fonctionne aussi avec les objets de la ROM | #06657h |
| NEWOB SWAP | #62C69h |
| Nom de la variable du répertoire courant dont le contenu est donné en argument | #082E3h |

| Fonction | Adresse |
|--|---------|
| ORDER | #20FF2h |
| PATH | #1848Ch |
| PGDIR | #18595h |
| Place le contenu du niveau 2 dans la variable locale dont le numéro est précisé par l'entier système placé au niveau 1 | #075E9h |
| Place le contenu du niveau 2 dans la variable locale existante et dont le nom est placé au niveau 1 | #07D1Bh |
| PURGE appliqué au dernier lot de variables globales | #07497h |
| PURGE fonctionne aussi avec les répertoires | #08C27h |
| PURGE prend un nom de variable globale en argument | #1854Fh |
| PURGE prend un nom global en argument et ne s'applique qu'à une variable du répertoire caché | #6408Ch |
| PVARS prend en argument un entier système désignant le port choisi | #21839h |
| PVARS prend en argument un réel désignant le port choisi | #2120Bh |
| Rappel de la variable locale n°01 | #613B6h |
| Rappel de la variable locale n°02 | #613E7h |
| Rappel de la variable locale n°03 | #6140Eh |
| Rappel de la variable locale n°04 | #61438h |
| Rappel de la variable locale n°05 | #6145Ch |
| Rappel de la variable locale n°06 | #6146Ch |
| Rappel de la variable locale n°07 | #6147Ch |
| Rappel de la variable locale n°08 | #6148Ch |
| Rappel de la variable locale n°09 | #6149Ch |
| Rappel de la variable locale n°10 | #614ACh |
| Rappel de la variable locale n°11 | #614BCh |
| Rappel de la variable locale n°12 | #614CCh |
| Rappel de la variable locale n°13 | #614DCh |
| Rappel de la variable locale n°14 | #614ECh |
| Rappel de la variable locale n°15 | #614FCh |

| Fonction | Adresse |
|--|---------|
| Rappel de la variable locale n°16 | #6150Ch |
| Rappel de la variable locale n°17 | #6151Ch |
| Rappel de la variable locale n°18 | #6152Ch |
| Rappel de la variable locale n°19 | #6153Ch |
| Rappel de la variable locale n°20 | #6154Ch |
| Rappel de la variable locale n°21 | #6155Ch |
| Rappel de la variable locale n°22 | #6156Ch |
| Rappel du contenu d'une variable du répertoire caché suivi de TRUE (renvoie FALSE si la variable n'existe pas) | #64023h |
| RCL s'applique à une liste { chemin_d'accès 'nom' } | #20B9Ah |
| RCL applicable à un nom de variable globale ou locale | #20B81h |
| Remplacement de la pile courante par la pile sauvegardée | #61F8Fh |
| RESTORE prend en argument un objet-taggué | #2134Bh |
| Sauvegarde de la pile courante | #61D41h |
| STO avec remplacement d'une variable locale (si le nom est local) ou création (ou remplacement) d'une variable globale (susceptible de remplacer un répertoire) | #07D27h |
| STO avec stockage des étiquettes et absence de protection des répertoires | #07D27h |
| STO avec stockage des étiquettes et protection des répertoires | #18513h |
| Stockage dans le répertoire caché de l'objet du niveau 2 sous le nom donné au niveau 1 | #64078h |
| Stockage de l'objet du niveau 1 dans la variable locale n°01 | #615E0h |
| Stockage de l'objet du niveau 1 dans la variable locale n°02 | #615F0h |
| Stockage de l'objet du niveau 1 dans la variable locale n°03 | #61600h |
| Stockage de l'objet du niveau 1 dans la variable locale n°04 | #61615h |
| Stockage de l'objet du niveau 1 dans la variable locale n°05 | #61625h |
| Stockage de l'objet du niveau 1 dans la variable locale n°06 | #61635h |
| Stockage de l'objet du niveau 1 dans la variable locale n°07 | #61645h |
| Stockage de l'objet du niveau 1 dans la variable locale n°08 | #61655h |

| Fonction | Adresse |
|--|---------|
| Stockage de l'objet du niveau 1 dans la variable locale n°09 | #61665h |
| Stockage de l'objet du niveau 1 dans la variable locale n°10 | #61675h |
| Stockage de l'objet du niveau 1 dans la variable locale n°11 | #61685h |
| Stockage de l'objet du niveau 1 dans la variable locale n°12 | #61695h |
| Stockage de l'objet du niveau 1 dans la variable locale n°13 | #616A5h |
| Stockage de l'objet du niveau 1 dans la variable locale n°14 | #616B5h |
| Stockage de l'objet du niveau 1 dans la variable locale n°15 | #616C5h |
| Stockage de l'objet du niveau 1 dans la variable locale n°16 | #616D5h |
| Stockage de l'objet du niveau 1 dans la variable locale n°17 | #616E5h |
| Stockage de l'objet du niveau 1 dans la variable locale n°18 | #616F5h |
| Stockage de l'objet du niveau 1 dans la variable locale n°19 | #61705h |
| Stockage de l'objet du niveau 1 dans la variable locale n°20 | #61715h |
| Stockage de l'objet du niveau 1 dans la variable locale n°21 | #61725h |
| Stockage de l'objet du niveau 1 dans la variable locale n°22 | #61735h |
| Suppression des derniers arguments même si le flag -55 est désarmé | #112ECh |
| Suppression des derniers arguments si le flag -55 est armé | #53842h |
| SYSEVAL prend en argument un entier binaire | #1A547h |
| SYSEVAL prend en argument un entier système | #1A556h |
| TVARS prend un réel en argument | #186E8h |
| UPDIR | #1A16Fh |
| VARS | #18779h |

Gestion du clavier



Les adresses qui suivent sont utilisables avec SYSEVAL. Ces adresses correspondent pour la plupart à des instructions équivalentes à leurs homologues en RPL-utilisateur, certaines réserves devant toutefois souvent être émises quant à leur champ d'application. Les instructions équivalentes du RPL-utilisateur sont communiquées quand elles existent, par exemple, **ASN**.

| Fonction | Adresse |
|--|---------|
| ASN prend un réel en argument | #41B28h |
| DELKEYS ne peut prendre que 'S' comme argument | #41BB9h |
| DELKEYS prend une liste en argument | #41B3Ch |
| Initialisation du tampon de clavier sans traitement de son contenu | #00D71h |
| STOKEYS ne peut prendre que 'S' comme argument | #41BA5h |
| STOKEYS prend une liste en argument | #41AA1h |
| Test : la touche [ON] a-t-elle été pressée ? Renvoie un booléen | #42262h |
| Test : le tampon de clavier est-il non vide ? Renvoie un booléen | #42402h |
| Test : le tampon de clavier est-il non vide ? Si oui, renvoie TRUE, sinon, renvoie le code de la touche la plus ancienne suivi de TRUE. Cette touche est ôtée du tampon. | #04714h |

| Fonction | Adresse |
|---|----------------|
| Test : le tampon de clavier est-il non vide ? Si oui, renvoie TRUE, sinon, renvoie le code de la touche la plus ancienne suivi de TRUE. Pas de modification du contenu du tampon. | #04708h |

Graphisme



Les adresses qui suivent sont utilisables avec SYSEVAL. Ces adresses correspondent pour la plupart à des instructions (ou à des séquences d'instructions) équivalentes à leurs homonymes en RPL-utilisateur, certaines réserves devant toutefois souvent être émises quant à leur champ d'application. Les instructions équivalentes du RPL-utilisateur sont communiquées quand elles existent, par exemple, **FREEZE**.

| Fonction | Adresse |
|---|---------|
| *H prend un réel en argument | #4B553h |
| *W prend un réel en argument | #4B5ADh |
| 1 DISP prend une chaîne de caractères en argument | #1245Bh |
| 1 FREEZE | #3902Ch |
| 1 FREEZE 2 FREEZE | #3921Bh |
| 1 FREEZE 2 FREEZE 4 FREEZE | #3922Fh |
| 2 DISP prend une chaîne de caractères en argument | #1246Bh |
| 3 DISP prend une chaîne de caractères en argument | #1247Bh |
| 4 DISP prend une chaîne de caractères en argument | #1248Bh |
| 5 DISP prend une chaîne de caractères en argument | #1249Bh |
| 6 DISP prend une chaîne de caractères en argument | #124ABh |

| Fonction | Adresse |
|--|----------------|
| 7 DISP prend une chaîne de caractères en argument | #124BBh |
| Affichage dans le grob courant de l'expression algébrique placée au niveau 1. L'expression apparaît comme si elle était éditée par Equation Writer. Si besoin est, les dimensions du GROB sont augmentées. | #659DEh |
| Affiche la chaîne placée au niveau 1 en caractère de taille intermédiaire sur le grob courant | #12725h |
| Augmentation de la hauteur d'un GROB d'un nombre de lignes blanches précisé par un entier système | #12DD1h |
| BOX prend deux complexes en arguments | #4F688h |
| BOX prend quatre entiers système en arguments | #4E582h |
| CLLCD | #5046Ah |
| Création d'un GROB blanc dont les dimensions sont données par deux entiers système | #1158Fh |
| ERASE | #4B60Ch |
| Hauteur puis largeur (entiers système) du GROB se trouvant au niveau 1 de la pile | #50578h |
| Initialisation de l'écran (zone d'état, pile et menu) | #01FA7h |
| Initialisation des 55 premières lignes du grob courant (large de 131 colonnes) | #01F6Dh |
| Initialisation des lignes 0 à 15 de l'écran | #0E06Fh |
| Initialisation des lignes 0 à 7 de l'écran | #0E083h |
| Initialisation des lignes 8 à 15 de l'écran | #0E097h |
| Initialisation du GROB courant avec conservation de ses dimensions | #134AEh |
| Inversion du GROB donné en argument (sans NEWOB) | #122FFh |
| Largeur (entier système) du GROB se trouvant au niveau 1 | #63C04h |
| Largeur puis hauteur (entiers binaires) du GROB se trouvant au niveau 1 de la pile | #1CA62h |

| Fonction | Adresse |
|---|---------|
| LCD → | #503D4h |
| LINE prend deux complexes en arguments | #4F584h |
| LINE prend quatre entiers système en arguments | #50758h |
| PDIM prend en argument deux entiers système déterminant les dimensions de PICT | #4B323h |
| PIXOFF prend deux entiers système en arguments | #1380Fh |
| PIXOFF prend un complexe en argument | #4F471h |
| PIXON prend deux entiers système en arguments | #13825h |
| PIXON prend un complexe en argument | #4F3EFh |
| RCEQ | #1572Bh |
| Scrolling d'une colonne vers la droite | #1357Fh |
| Scrolling d'une colonne vers la droite avec répétition si la touche [→] est enfoncée | #4D18Ch |
| Scrolling d'une colonne vers la gauche | #134E4h |
| Scrolling d'une colonne vers la gauche avec répétition si la touche [←] est enfoncée | #4D150h |
| Scrolling d'une ligne vers le bas | #13220h |
| Scrolling d'une ligne vers le bas avec répétition si la touche [↓] est enfoncée | #4D16Eh |
| Scrolling d'une ligne vers le haut | #131C8h |
| Scrolling d'une ligne vers le haut avec répétition si la touche [↑] est enfoncée | #4D132h |
| Scrolling vers la droite (jusqu'à l'extrémité du GROB) | #51703h |
| Scrolling vers la gauche (jusqu'à l'extrémité du GROB) | #516E5h |
| Scrolling vers le bas (jusqu'à l'extrémité du GROB) | #516AEh |
| Scrolling vers le haut (jusqu'à l'extrémité du GROB) | #51690h |
| STEQ | #15717h |

| Fonction | Adresse |
|--|----------------|
| TEXT | #503C5h |
| TLINE prend deux complexes en arguments | #4F598h |
| TLINE prend quatre entiers système en arguments | #5072Bh |

Modes et flags



Les adresses qui suivent sont utilisables avec SYSEVAL. Certaines adresses correspondent à leurs homonymes en RPL-utilisateur, quelques réserves devant toutefois souvent être émises quant à leur champ d'application (types des arguments ou des résultats). Les instructions équivalentes du RPL-utilisateur sont communiquées quand elles existent, par exemple, **BIN**.

| Fonction | Adresse |
|---|---------|
| Allumage du témoin "shift-droit" dans la zone d'état | #11347h |
| Allumage du témoin "shift-gauche" dans la zone d'état | #11361h |
| Allumage du témoin alpha dans la zone d'état | #1132Dh |
| Armement du flag -3 (résultats numériques) | #53B9Ch |
| Armement du flag -62 (activation du mode USER) | #40D61h |
| Armement du flag -n (n est un entier système) | #53731h |
| Armement du flag n (n est un entier système) | #53725h |
| BIN | #53C43h |
| CF prend un réel comme argument | #1C2EEh |
| DEC | #53C5Bh |
| DEG | #2A5D2h |

| Fonction | Adresse |
|--|---------|
| Désarmement du flag -3 (résultats numériques) | #53B88h |
| Désarmement du flag -62 (désactivation du mode USER) | #40D7Ah |
| Désarmement du flag -n (n est un entier système) | #53761h |
| Désarmement du flag n (n est un entier système) | #53755h |
| ENG prend un entier système comme argument | #166FBh |
| ENG prend un réel comme argument | #1C46Bh |
| Extinction des témoins alpha, "shift-droit" et "shift-gauche" dans la zone d'état | #3FCAFh |
| Extinction du témoin "shift-droit" dans la zone d'état | #11354h |
| Extinction du témoin "shift-gauche" dans la zone d'état | #1136Eh |
| Extinction du témoin alpha dans la zone d'état | #1133Ah |
| Extinction du témoin d'entrée/sortie dans la zone d'état | #11387h |
| FIX prend un entier système comme argument | #166E3h |
| FIX prend un réel comme argument | #1C403h |
| GRAD | #2A604h |
| HEX | #53C37h |
| Inversion de l'état du flag -62 | #40D4Dh |
| Inversion de l'état du flag -n (n est un entier système) | #3EDA2h |
| L'état du flag dont le numéro est précisé par un réel est renvoyé sous la forme 0 (armé) ou 1 (désarmé) | #1C379h |
| L'état du flag dont le numéro est précisé par un réel est renvoyé sous la forme 0 (désarmé) ou 1 (armé) | #1C32Ch |
| La base de numération utilisée pour les entiers binaires est renvoyée sous la forme d'un entier système | #54050h |
| La base de numération utilisée pour les entiers binaires est renvoyée sous la forme du caractère utilisé comme suffixe pour cette base | #5407Ah |

| Fonction | Adresse |
|--|---------|
| Les flags système sont modifiés en fonction de l'entier binaire placé au niveau 1 | #1C6E3h |
| Les flags utilisateur sont modifiés en fonction de l'entier binaire placé au niveau 1 | #1C6F7h |
| OCT | #53C4Fh |
| RAD | #2A5F0h |
| Renvoi dans la pile de l'entier binaire (64 bits) résumant l'état des 64 flags système | #1C637h |
| Renvoi dans la pile de l'entier binaire (64 bits) résumant l'état des 64 flags utilisateur | #1C64Eh |
| SCI prend un entier système comme argument | #166EFh |
| SCI prend un réel comme argument | #1C437h |
| SF prend un réel comme argument | #1C28Dh |
| STD | #16707h |
| STOF l'état des flags système est représenté par l'entier binaire du niveau 2 alors que les flags utilisateur sont représentés par l'entier binaire placé au niveau 1 | #1C6CFh |
| STOF prend une liste en argument | #1C6A2h |
| STWS prend un entier système en argument | #53CAAh |
| Taille courante des entiers binaires renvoyée sous la forme d'un entier système | #54039h |
| Test : le flag -51 (point décimal) est-il désarmé ? Renvoie un booléen | #167BFh |

Communication série



Les adresses qui suivent sont utilisables avec SYSEVAL. Elles correspondent pour la plupart à des instructions équivalentes à leurs homonymes en RPL-utilisateur, certaines réserves devant toutefois souvent être émises quant à leur champ d'application (type des arguments). Les instructions équivalentes du RPL-utilisateur sont communiquées quand elles existent, par exemple, **BUFLN**.

| Fonction | Adresse |
|--|---------|
| BAUD prend un réel en argument | #2EC84h |
| BUFLN | #2EDE1h |
| CLOSEIO avec initialisation du tampon série | #315C6h |
| Création de IOPAR dans HOME si la variable n'existe pas ou vérification de sa validité si la variable existe | #2EA4Fh |
| DELAY prend un réel en argument | #31FFDh |
| Retour chariot effectué par l'imprimante | #31854h |
| Fin du mode serveur Kermit | #2D9B2h |
| Impression d'une chaîne de caractères donnée en argument | #32161h |
| Initialisation du port de communication et vérification de IOPAR | #2EB37h |
| KERRM | #2EDA6h |
| KGET prend un nom global ou une chaîne de caractère en argument | #2E7EFh |

| Fonction | Adresse |
|--|----------------|
| PARITY prend un réel en argument | #2ECCAh |
| PKT prend deux chaînes en argument | #2E8D1h |
| PR1 prend un objet graphique en argument | #32B74h |
| PRLCD | #31EE2h |
| SBRK | #2EE18h |
| SEND prend une liste en argument | #2E6EBh |
| SRCEV prend un réel en argument | #2EE97h |
| Stockage dans IOPAR de l'objet placé au niveau 1 sans test de validité | #2E9CBh |
| TRANSIO prend un réel en argument | #2ED10h |

Adresses en vrac



Ces adresses correspondent pour la plupart à des séquences d'instructions équivalentes à leurs homonymes en RPL-utilisateur. Les premières adresses de cette liste placent des constantes (de tous types) au niveau 1 de la pile. Le symbole → précède les arguments alors que le symbole ← précède les résultats. Ceux-ci ne sont d'ailleurs précisés que lorsqu'ils ne sont pas évidents, et ce, afin de ne pas allonger démesurément cette liste.

| Fonction | Adresse | |
|--------------------------|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| 180 | #650FCh | #650FCh |
| 200 | #65111h | #65111h |
| 360 | #65126h | #65126h |
| 400 | #6513Bh | #6513Bh |
| 8192 | #0EFEEh | #0EFEEh |
| 8192 | #1A7CEh | #1A7CEh |
| 491520 | #0F003h | #0F003h |
| 29491200 | #0F018h | #0F018h |
| 707788800 | #0F02Dh | #0F02Dh |
| 4954521600 | #0F042h | #0F042h |
| ! | #2B0C4h | #2B0C4h |
| →1: réel | | |
| " " chaîne de caractères | #65254h | #65254h |

| Fonction | Adresse | |
|-----------------------------|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| "#" chaîne de caractères | #6518Eh | #6518Eh |
| "\$" chaîne de caractères | #651A6h | #651A6h |
| "&" chaîne de caractères | #651B2h | #651B2h |
| "\" chaîne de caractères | #65284h | #65284h |
| "\'" chaîne de caractères | #6571Fh | #6571Fh |
| "(" chaîne de caractères | #652B4h | #652B4h |
| "()" chaîne de caractères | #6573Bh | #6573Bh |
| ")" chaîne de caractères | #652C0h | #652C0h |
| "*" chaîne de caractères | #652D8h | #652D8h |
| "+" chaîne de caractères | #652F0h | #652F0h |
| "," chaîne de caractères | #65290h | #65290h |
| "-" chaîne de caractères | #652FCh | #652FCh |
| "." chaîne de caractères | #6529Ch | #6529Ch |
| "0" chaîne de caractères | #6534Ch | #6534Ch |
| "1" chaîne de caractères | #65358h | #65358h |
| "2" chaîne de caractères | #65364h | #65364h |
| "3" chaîne de caractères | #65370h | #65370h |
| "4" chaîne de caractères | #6537Ch | #6537Ch |
| "5" chaîne de caractères | #65388h | #65388h |
| "6" chaîne de caractères | #65394h | #65394h |
| "7" chaîne de caractères | #653A0h | #653A0h |
| "8" chaîne de caractères | #653ACh | #653ACh |
| "9" chaîne de caractères | #653B8h | #653B8h |
| ": " chaîne de caractères | #15442h | #15442h |
| ": : " chaîne de caractères | #6572Dh | #6572Dh |
| ";" chaîne de caractères | #652A8h | #652A8h |
| "<" chaîne de caractères | #651EEh | #651EEh |
| "=" chaîne de caractères | #65308h | #65308h |
| "EXIT" chaîne de caractères | #65769h | #65769h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| "GRAD" chaîne de caractères | #657A7h | #657A7h |
| "GROB" chaîne de caractères | #6532Ch | #6532Ch |
| "R∕∕" chaîne de caractères | #656C5h | #656C5h |
| "R∕Z" chaîne de caractères | #656D5h | #656D5h |
| "RAD" chaîne de caractères | #65797h | #65797h |
| "Undefined" chaîne de caractères | #6577Bh | #6577Bh |
| "UNKNOWN" chaîne de caractères | #65260h | #65260h |
| "[" chaîne de caractères | #6515Ch | #6515Ch |
| "[" chaîne de caractères | #6516Ah | #6516Ah |
| "[]" chaîne de caractères | #65711h | #65711h |
| "]" chaîne de caractères | #65150h | #65150h |
| "^" chaîne de caractères | #652CCh | #652CCh |
| "_" chaîne de caractères | #6519Ah | #6519Ah |
| "{" chaîne de caractères | #65176h | #65176h |
| "{" chaîne de caractères | #65703h | #65703h |
| " " chaîne de caractères | #65206h | #65206h |
| "}" chaîne de caractères | #65182h | #65182h |
| "Σ" chaîne de caractères | #651FAh | #651FAh |
| "√" chaîne de caractères | #65314h | #65314h |
| % (XLIB 2 124) | #1C060h | #1C060h |
| % interne → 2: expr. symb. 1: expr. symb. | #55C6Fh | #55C6Fh |
| % interne → 2: expr. symb. 1: réel/réel_unité | #55C3Dh | #55C3Dh |
| % interne → 2: réel_unité 1: réel | #0FBABh | #0FBABh |
| % interne → 2: réel 1: réel | #2A9C9h | #2A9C9h |
| % interne → 2: réel/réel_unité 1: expr. symb. | #55C56h | #55C56h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| %CH (XLIB 2 126) | #1C149h | #1C149h |
| %CH interne →2: expr. symb. 1: expr. symb. | #55CBAh | #55CBAh |
| %CH interne →2: expr. symb. 1: réel/réel_unité | #55C88h | #55C88h |
| %CH interne →2: réel 1: réel | #2AA30h | #2AA30h |
| %CH interne →2: réel/réel_unité 1: expr. symb. | #55CA1h | #55CA1h |
| %CH interne →2: réel/réel_unité 1: réel/réel_unité | #0FC3Ch | #0FC3Ch |
| %T (XLIB 2 125) | #1C0D7h | #1C0D7h |
| %T interne →2: expr. symb. 1: expr. symb. | #55D05h | #55D05h |
| %T interne →2: expr. symb. 1: réel/réel_unité | #55CD3h | #55CD3h |
| %T interne →2: réel 1: réel | #2AA0Bh | #2AA0Bh |
| %T interne →2: réel/réel_unité 1: expr. symb. | #55CECh | #55CECh |
| %T interne →2: réel/réel_unité 1: réel/réel_unité | #0FCCDh | #0FCCDh |
| * →2: entier sys. 1: entier sys. | #03EC2h | #03EC2h |
| * (XLIB 2 71) | #1ADEEh | #1ADEEh |
| * interne →2/1: réel/complexe 1/2: matrice | #362DCh | #362DCh |
| * interne →2: entier bin. 1: entier bin. | #53ED3h | #53ED3h |
| * interne →2: entier bin. 1: réel | #542EAh | #542EAh |
| * interne →2: expr. symb. 1: réel/Cplx/réel_unité | #55FC1h | #55FC1h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| * <i>interne</i> →2: expr. symb. 1: expr. symb. | #55FF3h | #55FF3h |
| * <i>interne</i> →2: réel long 1: réel long | #2A99Ah | #2A99Ah |
| * <i>interne</i> →2: complexe 1: complexe | #51D88h | #51D88h |
| * <i>interne</i> →2: complexe 1: réel | #51D4Ch | #51D4Ch |
| * <i>interne</i> →2: réel 1: entier bin. | #542D1h | #542D1h |
| * <i>interne</i> →2: réel 1: complexe | #51D60h | #51D60h |
| * <i>interne</i> →2: réel 1: réel | #2A9BCh | #2A9BCh |
| * <i>interne</i> →2:réel/cplx/réel_unité 1: expr. symb. | #55FDAh | #55FDAh |
| * <i>interne</i> →2: réel/réel_unité 1: réel/réel_unité | #0F792h | #0F792h |
| *10 <i>interne</i> →1: entier sys. | #6264Eh | #6264Eh |
| *10 <i>interne</i> →1: réel | #62BF1h | #62BF1h |
| *1000 →1: réel | #1415Fh | #1415Fh |
| *6 <i>interne</i> →1: entier sys. | #62691h | #62691h |
| *8 <i>interne</i> →1: entier sys. | #62674h | #62674h |
| *H <i>interne</i> →1: reel | #4B553h | #4B553h |
| *W <i>interne</i> →1: réel | #4B5ADh | #4B5ADh |
| + | #1ACD7h | #1ACD7h |
| + (XLIB 2 68) | #1AB67h | #1AB67h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| + (XLIB 2 69) | #1ACD7h | #1ACD7h |
| + et DUP interne →2: entier sys. 1: entier sys. | #627D5h | #627D5h |
| + et OVER interne →1: entier sys. | #63051h | #63051h |
| + et ROLL interne →n..3: quelconque 2: ent. sys.1: ent. sys. | #612DEh | #612DEh |
| + et ROLLD interne →n..3: quelconque 2: ent. sys.1: ent. sys. | #6133Eh | #6133Eh |
| + et SWAP interne →1: entier sys. | #62DFEh | #62DFEh |
| + et SWAP interne →2: chaîne 1: chaîne | #63F6Ah | #63F6Ah |
| + interne →2: chaîne 1: caractère | #052EEh | #052EEh |
| + interne →2: chaîne 1: chaîne | #0518Ah | #0518Ah |
| + interne →2: chaîne 1: quelconque | #1ACA7h | #1ACA7h |
| + interne →2: entier bin. 1: entier bin. | #53EA0h | #53EA0h |
| + interne →2: entier bin. 1: réel | #54349h | #54349h |
| + interne →2: entier sys. 1: entier sys. | #03DBCh | #03DBCh |
| + interne →2: expr. symb. 1: expr. symb. | #55F5Dh | #55F5Dh |
| + interne →2: expr. symb. 1: réel/cplx/réel_unité | #55F2Bh | #55F2Bh |
| + interne →2: Graphique 1: Graphique | #4F8D1h | #4F8D1h |
| + interne →2: liste 1: quelconque | #052FAh | #052FAh |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| + <i>interne</i> → 2: réel long 1: réel long | #2A943h | #2A943h |
| + <i>interne</i> → 2: matrice 1: matrice | #36115h | #36115h |
| + <i>interne</i> → 2: complexe 1: complexe | #51C16h | #51C16h |
| + <i>interne</i> → 2: complexe 1: réel | #51BD0h | #51BD0h |
| + <i>interne</i> → 2: quelconque 1: chaîne | #1ACBBh | #1ACBBh |
| + <i>interne</i> → 2: quelconque 1: liste | #1AC93h | #1AC93h |
| + <i>interne</i> → 2: réel 1: entier bin. | #54330h | #54330h |
| + <i>interne</i> → 2: réel 1: complexe | #51BF8h | #51BF8h |
| + <i>interne</i> → 2: réel 1: réel | #2A974h | #2A974h |
| + <i>interne</i> → 2:réel/cplx/réel_unité 1:expr. symb. | #55F44h | #55F44h |
| + <i>interne</i> → 2:réel/réel_unité 1:réel/réel_unité | #0F6A2h | #0F6A2h |
| +1 <i>et DUP interne</i> → 1: entier sys. | #62809h | #62809h |
| +1 <i>et PICK interne</i> → n..2: quelconque 1: entier sys. | #611A3h | #611A3h |
| +1 <i>et ROLL interne</i> → n..2: quelconque 1: entier sys. | #612F3h | #612F3h |
| +1 <i>et ROLLD interne</i> → n..2: quelconque 1: entier sys. | #61353h | #61353h |
| +1 <i>interne</i> → 1: entier sys. | #03DEFh | #03DEFh |
| +1 <i>interne</i> → 1: réel | #50262h | #50262h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| +10 interne →1: entier sys. | #625DAh | #625DAh |
| +12 interne →1: entier sys. | #625EAh | #625EAh |
| +2 et PICK →1: entier sys. | #611BEh | #611BEh |
| +2 et ROLL interne →n..2: quelconque 1: entier sys. | #61318h | #61318h |
| +2 et ROLLD interne →n..2: quelconque 1: entier sys. | #61365h | #61365h |
| +2 interne →1: entier sys. | #03E2Dh | #03E2Dh |
| +3 et PICK →1: entier sys. | #611D2h | #611D2h |
| +3 interne →1: entier sys. | #6256Ah | #6256Ah |
| +4 interne →1: entier sys. | #6257Ah | #6257Ah |
| +5 interne →1: entier sys. | #6258Ah | #6258Ah |
| +6 interne →1: entier sys. | #6259Ah | #6259Ah |
| +7 interne →1: entier sys. | #625AAh | #625AAh |
| +8 interne →1: entier sys. | #625BAh | #625BAh |
| +9 interne →1: entier sys. | #625CAh | #625CAh |
| - (XLIB 2 70) | #1AD09h | #1AD09h |
| - et ROLL interne →n..3:quelconque 2:ent. sys. 1:ent. sys. | #612CCh | #612CCh |
| - et ROLLD interne →n..3:quelconque 2:ent. sys. 1:ent. sys. | #6132Ch | #6132Ch |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| - <i>interne</i> → 2: entier bin. 1: entier bin. | #53EB0h | #53EB0h |
| - <i>interne</i> → 2: entier bin. 1: réel | #5431Ch | #5431Ch |
| - <i>interne</i> → 2: entier sys. 1: entier sys. | #03DE0h | #03DE0h |
| - <i>interne</i> → 2: expr. symb. 1: expr. symb. | #55FA8h | #55FA8h |
| - <i>interne</i> → 2: expr. symb. 1: réel/cplx/réel_unité | #55F76h | #55F76h |
| - <i>interne</i> → 2: réel long 1: réel long | #2A94Fh | #2A94Fh |
| - <i>interne</i> → 2: matrice 1: matrice | #36278h | #36278h |
| - <i>interne</i> → 2: complexe 1: réel | #51CE8h | #51CE8h |
| - <i>interne</i> → 2: complexe 1: réel | #51CFCh | #51CFCh |
| - <i>interne</i> → 2: réel 1: entier bin. | #542FEh | #542FEh |
| - <i>interne</i> → 2: réel 1: complexe | #51CD4h | #51CD4h |
| - <i>interne</i> → 2: réel 1: réel | #2A981h | #2A981h |
| - <i>interne</i> → 2: réel/cplx/réel_unité 1: expr. symb. | #55F8Fh | #55F8Fh |
| - <i>interne</i> → 2: réel/réel_unité 1: réel/réel_unité | #0F774h | #0F774h |
| - .5 | #650D2h | #650D2h |
| -1 <1h> et SWAP interne → 1: entier sys. | #62E4Eh | #62E4Eh |
| -1 et → LIST interne → n..2: quelconque 1: entier sys. | #631A5h | #631A5h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| -1 et DUP interne →1: entier sys. | #6281Ah | #6281Ah |
| -1 et ROT interne →1: entier sys. | #62FD9h | #62FD9h |
| -1 interne →1: entier sys. | #03E0Eh | #03E0Eh |
| -1.E-499 | #2A4B1h | #2A4B1h |
| -2 interne →1: entier sys. | #03E4Eh | #03E4Eh |
| -3 interne →1: entier sys. | #625FAh | #625FAh |
| -4 interne →1: entier sys. | #6260Ah | #6260Ah |
| -5 interne →1: entier sys. | #6261Ah | #6261Ah |
| -6 interne →1: entier sys. | #6262Ah | #6262Ah |
| -9.999999999999E499 | #2A487h | #2A487h |
| .1 (réel Long) | #2A562h | #2A562h |
| .5 | #650BDh | #650BDh |
| .5 (réel Long) | #2A57Ch | #2A57Ch |
| / (XLIB 2 72) | #1AF05h | #1AF05h |
| / interne →2: entier sys. 1: entier sys. ← 2: reste 1: quotient | #03EF7h | #03EF7h |
| / interne →2: entier bin. 1: entier bin. | #53F05h | #53F05h |
| / interne →2: entier bin. 1: réel | #542BDh | #542BDh |
| / interne →2: expr. symb. 1: expr. symb. | #5603Eh | #5603Eh |
| / interne →2: expr. symb. 1: réel/cplx/réel_unité | #5600Ch | #5600Ch |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| / interne →2: réel long 1: réel long | #2A9E8h | #2A9E8h |
| / interne →2: matrice 1: matrice | #36B60h | #36AC3h |
| / interne →2: matrice 1: réel/complexe | #363CCh | #363DBh |
| / interne →2: complexe 1: complexe | #51EC8h | #51EC8h |
| / interne →2: réel 1: entier bin. | #5429Fh | #5429Fh |
| / interne →2: réel 1: réel | #2A9FEh | #2A9FEh |
| / interne →2:réel/cplx/réel_unité 1: expr. symb. | #56025h | #56025h |
| / interne →2:réel/réel_unité 1: réel/réel_unité | #0F823h | #0F823h |
| /2 interne →1: entier sys. | #03E8Eh | #03E8Eh |
| 0 (réel Long) | #2A4C6h | #2A4C6h |
| 1 (réel Long) | #2A4E0h | #2A4E0h |
| 1 →GROB interne →1: chaîne | #1200Ch | #1200Ch |
| 1.E-499 | #2A49Ch | #2A49Ch |
| 10 (réel Long) | #2A596h | #2A596h |
| 10 ROLLD interne | #6312Dh | #6312Dh |
| 2 (réel Long) | #2A4FAh | #2A4FAh |
| 2 →GROB interne →1: chaîne | #11D00h | #11D00h |
| 2 →LIST interne | #631B9h | #631B9h |
| 2.71828182846 | #650A8h | #650A8h |
| 3 (réel Long) | #2A514h | #2A514h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| 3 →GROB interne →i: chaîne | #11CF3h | #11CF3h |
| 3 →LIST interne | #631CDh | #631CDh |
| 3 PICK interne | #611FEh | #611FEh |
| 3 ROLL interne | #60FACH | #60FACH |
| 3.14159265358979 (réel Long) | #2A458h | #2A458h |
| 3.14159265359 | #2A443h | #2A443h |
| 4 (réel Long) | #2A52Eh | #2A52Eh |
| 4 PICK + et SWAP interne →i: entier sys. | #62DE5h | #62DE5h |
| 4 PICK et 4 PICK interne | #63FBAh | #63FBAh |
| 4 PICK et SWAP interne | #62EF3h | #62EF3h |
| 4 PICK interne | #6121Ch | #6121Ch |
| 4 ROLL et SWAP interne | #62ECBh | #62ECBh |
| 4 ROLL interne | #60FBBh | #60FBBh |
| 4 ROLLD et DUP interne | #62D09h | #62D09h |
| 4 ROLLD interne | #6109Eh | #6109Eh |
| 5 (réel Long) | #2A548h | #2A548h |
| 5 PICK interne | #6123Ah | #6123Ah |
| 5 ROLL interne | #60FD8h | #60FD8h |
| 5 ROLLD interne | #610C4h | #610C4h |
| 6 PICK interne | #6125Eh | #6125Eh |
| 6 ROLL interne | #61002h | #61002h |
| 6 ROLLD interne | #610FAh | #610FAh |
| 7 PICK interne | #61282h | #61282h |
| 7 ROLL interne | #6106Bh | #6106Bh |
| 7 ROLLD interne | #62BC4h | #62BC4h |
| 8 PICK interne | #612A9h | #612A9h |
| 8 ROLL interne | #6103Ch | #6103Ch |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| 8 ROLLD interne | #63119h | #63119h |
| 9.999999999999E499 | #2A472h | #2A472h |
| < (XLIB 2 235) | #1EBBEh | #1EBBEh |
| < interne | #142A6h | #142A6h |
| →2: chaîne 1: chaîne | | |
| < interne | #54552h | #54552h |
| →2: entier bin. 1: entier bin. | | |
| < interne | #55B43h | #55B43h |
| →2: expr. symb. 1: expr. symb. | | |
| < interne | #55B11h | #55B11h |
| →2: expr. symb. 1: réel/réel_unité | | |
| < interne | #1EC40h | #1EC40h |
| →2: réel 1: réel | | |
| < interne | #2A81Fh | #2A81Fh |
| →2: réel 1: réel | | |
| ← 1: booléen | | |
| < interne | #2A871h | #2A871h |
| →2: réel 1: réel | | |
| ← 1: booléen | | |
| < interne | #55B2Ah | #55B2Ah |
| →2: réel/réel_unité 1: expr. symb. | | |
| < interne | #0F5ACh | #0F5ACh |
| →2: réel/réel_unité 1: réel/réel_unité | | |
| = (XLIB 2 59) | #1A8D8h | #1A8D8h |
| = interne | #55927h | #55927h |
| →2: quelconque 1: quelconque | | |
| == (XLIB 2 233) | #1E972h | #1E972h |
| == interne | #55A94h | #55A94h |
| →2: complexe 1: réel/cplx/réel_unité | | |
| == interne | #544D9h | #544D9h |
| →2: entier bin. 1: entier bin. | | |
| == interne | #55A7Bh | #55A7Bh |
| →2: expr. symb. 1: réel/cplx/réel_unité | | |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| == interne →2: expr. symb. 1: expr. symb. | #55AADh | #55AADh |
| == interne →2: complexe 1: réel | #1EA76h | #1EA76h |
| == interne →2: quelconque 1: quelconque | #1EA30h | #1EA30h |
| == interne →2: réel 1: complexe | #1EA6Ch | #1EA6Ch |
| == interne →2: réel 1: réel ← 1: booléen | #2A8C1h | #2A8C1h |
| == interne →2: réel/réel_unité 1: réel/réel_unité | #0F584h | #0F584h |
| == interne →2: Signé/quelconque 1: Signé/quelconque | #1EA44h | #1EA44h |
| > (XLIB 2 236) | #1EC5Dh | #1EC5Dh |
| > interne →2: chaîne 1: chaîne | #1420Ah | #1420Ah |
| > interne →2: entier bin. 1: entier bin. | #54500h | #54500h |
| > interne →2: expr. symb. 1: expr. symb. | #55B8Eh | #55B8Eh |
| > interne →2: expr. symb. 1: réel/réel_unité | #55B5Ch | #55B5Ch |
| > interne →2: réel 1: réel | #1ECDfh | #1ECDfh |
| > interne →2: réel 1: réel ← 1: booléen | #2A87Fh | #2A87Fh |
| > interne →2: réel 1: réel ← 1: booléen | #2A88Ah | #2A88Ah |
| > interne →2: réel/réel_unité 1: expr. symb. | #55B75h | #55B75h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| > <i>interne</i> →2: réel/réel_unité 1: réel/réel_unité | #0F5C0h | #0F5C0h |
| ABS (XLIB 2 61) | #1AA1Fh | #1AA1Fh |
| ABS interne →1: expr. symb. | #54F04h | #54F04h |
| ABS interne →1: réel long | #2A8F0h | #2A8F0h |
| ABS interne →1: complexe | #52062h | #52062h |
| ABS interne →1: réel_unité | #0F5FCh | #0F5FCh |
| ABS interne →1: réel | #2A900h | #2A900h |
| ABS interne →1: réel → 1: entier sys. | #18CD7h | #18CD7h |
| ABS(IP(réel)) →1: réel → 1: entier sys. | #2EC11h | #2EC11h |
| ACK (XLIB 2 21) | #1987Eh | #1987Eh |
| ACK interne | #0DDC1h | #0DDC1h |
| ACKALL (XLIB 2 20) | #19863h | #19863h |
| ACKALL interne | #0DDA8h | #0DDA8h |
| ACOS (XLIB 2 88) | #1B72Fh | #1B72Fh |
| ACOS interne →1: expr. symb. | #55062h | #55062h |
| ACOS interne →1: complexe | #52863h | #52863h |
| ACOS interne →1: réel ≤ 1 | #2ACF1h | #2ACF1h |
| ACOS interne →1: réel | #1B775h | #1B775h |
| ACOSH (XLIB 2 91) | #1B830h | #1B830h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| ACOSH interne → 1: expr. symb. | #550ADh | #550ADh |
| ACOSH interne → 1: complexe | #52836h | #52836h |
| ACOSH interne → 1: réel | #1B86Ch | #1B86Ch |
| → (XLIB 1792 16) | #234C1h | #234C1h |
| → (XLIB 1792 4) | #22FEBh | #22FEBh |
| → algébrique interne → n..1: entier sys. | #0546Dh | #0546Dh |
| → ARRAY (XLIB 2 170) | #1D009h | #1D009h |
| → ARRAY interne → 1: liste | #1D040h | #1D040h |
| → ARRAY interne → 1: réel | #1D02Ch | #1D02Ch |
| → DATE (XLIB 2 22) | #1989Eh | #1989Eh |
| → DATE interne | #0CD2Bh | #0CD2Bh |
| → entier sys. → 1: réel | #18CEAh | #18CEAh |
| → GROB | #1E5ADh | #1E5ADh |
| → GROB (XLIB 2 215) | #1E5ADh | #1E5ADh |
| → GROB interne → 2: quelconque 1: réel | #5048Dh | #5048Dh |
| → HMS (XLIB 2 114) | #1BF1Eh | #1BF1Eh |
| → HMS interne → 1: réel | #2A673h | #2A673h |
| → LCD | #1E58Dh | #1E58Dh |
| → LCD (XLIB 2 214) | #1E58Dh | #1E58Dh |
| → LCD interne → 1: Graphique | #50438h | #50438h |
| → LIST (XLIB 2 152) | #1C783h | #1C783h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| → LIST interne →n..2: quelconque 1: entier sys. | #05459h | #05459h |
| → NUM (XLIB 2 53) | #1A5E4h | #1A5E4h |
| → Programme et EVAL interne →n..2: quelconque 1: entier sys. | #632D1h | #632D1h |
| → programme interne | #63FE7h | #63FE7h |
| → programme interne →n..2: quelconque 1: entier sys. | #5445h | #5445h |
| → Q (XLIB 2 263) | #1F9C4h | #1F9C4h |
| → Qπ (XLIB 2 264) | #1F9E9h | #1F9E9h |
| → réel →1: entier sys. | #18DBFh | #18DBFh |
| → STR | #1CB0Bh | #1CB0Bh |
| → STR (XLIB 2 162) | #1CB0Bh | #1CB0Bh |
| → STR interne →1: quelconque | #14088h | #14088h |
| → STR interne →1: quelconque | #15978h | #15978h |
| → STR interne →1: quelconque | #15B13h | #15B13h |
| → STR interne →1: quelconque | #15B31h | #15B31h |
| → STR interne →1: réel | #15B3Dh | #15B3Dh |
| → STR interne →1: réel | #162ACh | #162ACh |
| → STR interne →1: réel | #162B8h | #162B8h |
| → TAG interne →2: quelconque 1: chaîne | #225F5h | #225F5h |
| → TAG interne →2: quelconque 1: réel | #22618h | #22618h |
| → TIME (XLIB 2 23) | #198BEh | #198BEh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| → TIME interne | #0CD53h | #0CD53h |
| → UNIT interne →n..2: quelconque 1: entier sys. | #05481h | #05481h |
| → V2 (XLIB 2 181) | #1DE66h | #1DE66h |
| → V2 interne →2: réel 1: réel | #1DE7Fh | #1DE7Fh |
| → V3 (XLIB 2 182) | #1DEC2h | #1DEC2h |
| → V3 interne →3: réel 2: réel 1: réel | #1DEDBh | #1DEDBh |
| <i>Affichage alarme suivante</i> | #19DE2h/ | - |
| <i>Allocation taille en quartets</i> →2: Objet 1: entier sys. | #61C1Ch | #61C1Ch |
| ALOG (XLIB 2 96) | #1BA3Dh | #1BA3Dh |
| ALOG interne →1: expr. symb. | #5512Ah | #5512Ah |
| ALOG interne →1: complexe | #52305h | #52305h |
| ALOG interne →1: réel | #2ABBAh | #2ABBAh |
| AND (XLIB 2 229) | #1E783h | #1E783h |
| AND interne →2: chaîne 1: chaîne | #18873h | #18873h |
| AND interne →2: entier bin. 1: entier bin. | #53D04h | #53D04h |
| AND interne →2: entier sys. 1: entier sys. | #03EB1h | #03EB1h |
| AND interne →2: expr. symb. 1: expr. symb. | #559CCh | #559CCh |
| AND interne →2: expr. symb. 1: réel | #5599Ah | #5599Ah |
| AND interne →2: réel 1: expr. symb. | #559B3h | #559B3h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| AND interne →2: réel 1: réel | #1E7DDh | #1E7DDh |
| APPLY (sur la pile)(XLIB 2 258) | #1F55Dh | #1F55Dh |
| APPLY (usage algébrique)(XLIB 2 259) | #1F5C5h | #1F5C5h |
| APPLY interne (sur la pile) →2: liste 1: nom global/nom local | #1F585h | #1F585h |
| ARCHIVE (XLIB 2 352) | #2125Ah | #2125Ah |
| ARCHIVE interne →1: Signé | #21273h | #21273h |
| ARG (XLIB 2 77) | #1B2DBh | #1B2DBh |
| ARG interne →1: expr. symb. | #54F4Fh | #54F4Fh |
| ARG interne →1: complexe | #52099h | #52099h |
| ARG interne →1: réel | #1B30Dh | #1B30Dh |
| ARRY → (XLIB 2 171) | #1D092h | #1D092h |
| ARRY → interne →1: matrice | #1D0ABh | #1D0ABh |
| ASIN (XLIB 2 87) | #1B6A4h | #1B6A4h |
| ASIN interne →1: expr. symb. | #55049h | #55049h |
| ASIN interne →1: complexe | #52804h | #52804h |
| ASIN interne →1: réel ≤ 1 | #2ACC1h | #2ACC1h |
| ASIN interne →1: réel | #1B6EAh | #1B6EAh |
| ASINH (XLIB 2 90) | #1B7EBh | #1B7EBh |
| ASINH interne →1: expr. symb. | #55094h | #55094h |
| ASINH interne →1: complexe | #5281Dh | #5281Dh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| ASINH interne → 1: réel | #2AE00h | #2AE00h |
| ASN (XLIB 2 379) | #224F4h | #224F4h |
| ASN interne → 2: quelconque 1: réel | #41B28h | #41B28h |
| ASR (XLIB 2 0) | #1957Bh | #1957Bh |
| ATAN (XLIB 2 89) | #1B79Ch | #1B79Ch |
| ATAN interne → 1: expr. symb. | #5507Bh | #5507Bh |
| ATAN interne → 1: complexe | #52675h | #52675h |
| ATAN interne → 1: réel | #2AD21h | #2AD21h |
| ATANH (XLIB 2 92) | #1B8A2h | #1B8A2h |
| ATANH interne → 1: expr. symb. | #550C6h | #550C6h |
| ATANH interne → 1: complexe | #527EBh | #527EBh |
| ATANH interne → 1: réel | #1B8DEh | #1B8DEh |
| ATTACH → 1: réel | #21461h | #21461h |
| ATTACH (pas sur HOME) interne → 2: répertoire 1: entier sys. | #21CBAh | #21CBAh |
| ATTACH (XLIB 2 357) | #21448h | #21448h |
| ATTACH sur répertoire HOME → 1: entier sys. | #07709h | #07709h |
| ATTACH interne → 1: entier sys. | #21C6Fh | #21C6Fh |
| AUTO (XLIB 2 192) | #1E1ABh | #1E1ABh |
| AUTO interne | #491D5h | #491D5h |
| AXES (XLIB 2 186) | #1E0BEh | #1E0BEh |

| Fonction | Adresse | |
|--|-----------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| AXES interne →1: complexe | #4B03Ah/#4B03Ah | |
| BAR | #1E741h/#1E741h | |
| BAR (XLIB 2 227) | #1E741h/#1E741h | |
| BARPLOT | #20133h/#20133h | |
| BARPLOT (XLIB 2 316) | #20133h/#20133h | |
| BAUD (XLIB 2 370) | #2200Ch/#2200Ch | |
| BAUD interne →1: réel | #2EC84h/#2EC84h | |
| BEEP (XLIB 2 52) | #1A5C4h/#1A5C4h | |
| BEEP interne →2: entier sys. 1: entier sys. | #141B2h/#141B2h | |
| BEEP interne →2: réel 1: réel | #1415Ah/#1415Ah | |
| BESTFIT (XLIB 2 323) | #2025Eh/#2025Eh | |
| BIN (XLIB 2 144) | #1C559h/#1C559h | |
| BIN interne | #53C43h/#53C43h | |
| Binaire→entier sys. →1: entier bin. | #05A03h/#05A03h | |
| BINS (XLIB 3 315) | #2010Eh/#2010Eh | |
| BINS interne →3: réel 2: réel 1: réel | #4C8F4h/#4C944h | |
| BLANK (XLIB 2 209) | #1E416h/#1E416h | |
| BLANK interne →2: entier bin. 1: entier bin. | #4F6A1h/#4F6A1h | |
| BLANK interne →2: entier sys. 1: entier sys. | #1158Fh/#1158Fh | |
| BOX (XLIB 2 208) | #1E3ECh/#1E3ECh | |
| BOX interne →2: liste 1: liste | #4F665h/#4F665h | |
| BOX interne →2: complexe 1: complexe | #4F688h/#4F688h | |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| BUFLEN (XLIB 2 374) | #22087h | #22087h |
| BUFLEN interne | #2EDE1h | #2EDE1h |
| BYTES (XLIB 2 38) | #1A1D9h | #1A1D9h |
| BYTES interne → 1: nom global | #1A265h | #1A265h |
| BYTES interne → 1: quelconque sauf nom global | #1A1FCh | #1A1FCh |
| BYTES interne ← 2: entier sys. 1: entier | #05944h | #05944h |
| C→PX (XLIB 2 199) | #1E29Ah | #1E29Ah |
| C→PX interne → 1: complexe | #4F179h | #4F179h |
| C→R (XLIB 2 159) | #1C98Eh | #1C98Eh |
| C→R interne → 1: matrice | #360B6h | #360B6h |
| C→R interne → 1: complexe | #05D2Ch | #05D2Ch |
| <i>caractère (espace)</i> | #65686h | #65686h |
| <i>caractère #</i> | #65433h | #65433h |
| <i>caractère (</i> | #65663h | #65663h |
| <i>caractère *</i> | #6543Ah | #6543Ah |
| <i>caractère +</i> | #65441h | #65441h |
| <i>caractère ,</i> | #65448h | #65448h |
| <i>caractère -</i> | #6544Fh | #6544Fh |
| <i>caractère .</i> | #65456h | #65456h |
| <i>caractère /</i> | #6545Dh | #6545Dh |
| <i>caractère 0</i> | #65464h | #65464h |
| <i>caractère 1</i> | #6546Bh | #6546Bh |
| <i>caractère 2</i> | #65472h | #65472h |
| <i>caractère 3</i> | #65479h | #65479h |
| <i>caractère 4</i> | #65480h | #65480h |

| Fonction | Adresse | |
|-------------------------------------|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| caractère 5 | #65487h | #65487h |
| caractère 6 | #6548Eh | #6548Eh |
| caractère 7 | #65495h | #65495h |
| caractère 8 | #6549Ch | #6549Ch |
| caractère 9 | #654A3h | #654A3h |
| caractère : | #654AAh | #654AAh |
| caractère ;" | #654B1h | #654B1h |
| caractère < | #654B8h | #654B8h |
| caractère = | #654BFh | #654BFh |
| caractère > | #654C6h | #654C6h |
| caractère A | #654CDh | #654CDh |
| caractère a | #65583h | #65583h |
| caractère B | #654D4h | #654D4h |
| caractère b | #6558Ah | #6558Ah |
| caractère C | #654DBh | #654DBh |
| caractère c | #65591h | #65591h |
| caractère ASCII n°0 | #6541Eh | #6541Eh |
| caractère ASCII n°10 (Line Feed) | #6566Ah | #6566Ah |
| caractère ASCII n°132 } | #6565Ch | #6565Ch |
| caractère ASCII n°133 Σ | #6567Fh | #6567Fh |
| caractère ASCII n°135 π | #65671h | #65671h |
| caractère ASCII n°136 der | #65655h | #65655h |
| caractère ASCII n°137 \leq | #656B0h | #656B0h |
| caractère ASCII n°138 \geq | #656B7h | #656B7h |
| caractère ASCII n°139 \neq | #656BEh | #656BEh |
| caractère ASCII n°141 \rightarrow | #65639h | #65639h |
| caractère ASCII n°171 " | #65640h | #65640h |
| caractère ASCII n°31 ... | #65425h | #65425h |
| caractère D | #65400h | #65400h |

| Fonction | Adresse | |
|--------------------|-------------------|-------------------|
| | HP-48 S/SX | HP-48 C/GX |
| <i>caractère d</i> | #65598h | #65598h |
| <i>caractère e</i> | #6559Fh | #6559Fh |
| <i>caractère F</i> | #654F0h | #654F0h |
| <i>caractère f</i> | #655A6h | #655A6h |
| <i>caractère G</i> | #654F7h | #654F7h |
| <i>caractère g</i> | #655ADh | #655ADh |
| <i>caractère H</i> | #654FEh | #654FEh |
| <i>caractère h</i> | #655B4h | #655B4h |
| <i>caractère I</i> | #65505h | #65505h |
| <i>caractère i</i> | #655BBh | #655BBh |
| <i>caractère J</i> | #6550Ch | #6550Ch |
| <i>caractère j</i> | #655C2h | #655C2h |
| <i>caractère K</i> | #65513h | #65513h |
| <i>caractère k</i> | #655C9h | #655C9h |
| <i>caractère L</i> | #6551Ah | #6551Ah |
| <i>caractère l</i> | #655D0h | #655D0h |
| <i>caractère M</i> | #65521h | #65521h |
| <i>caractère m</i> | #655D7h | #655D7h |
| <i>caractère N</i> | #65528h | #65528h |
| <i>caractère n</i> | #655DEh | #655DEh |
| <i>caractère O</i> | #6552Fh | #6552Fh |
| <i>caractère P</i> | #65536h | #65536h |
| <i>caractère p</i> | #655ECh | #655ECh |
| <i>caractère Q</i> | #6553Dh | #6553Dh |
| <i>caractère q</i> | #655F3h | #655F3h |
| <i>caractère R</i> | #65544h | #65544h |
| <i>caractère r</i> | #655FAh | #655FAh |
| <i>caractère S</i> | #6554Bh | #6554Bh |
| <i>caractère s</i> | #65601h | #65601h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| <i>caractère</i> T | #65552h | #65552h |
| <i>caractère</i> t | #65608h | #65608h |
| <i>caractère</i> U | #65559h | #65559h |
| <i>caractère</i> u | #6560Fh | #6560Fh |
| <i>caractère</i> V | #65560h | #65560h |
| <i>caractère</i> v | #65616h | #65616h |
| <i>caractère</i> W | #65567h | #65567h |
| <i>caractère</i> w | #6561Dh | #6561Dh |
| <i>caractère</i> X | #6556Eh | #6556Eh |
| <i>caractère</i> x | #65624h | #65624h |
| <i>caractère</i> Y | #65575h | #65575h |
| <i>caractère</i> y | #6562Bh | #6562Bh |
| <i>caractère</i> Z | #6557Ch | #6557Ch |
| <i>caractère</i> z | #65632h | #65632h |
| <i>caractère</i> [| #65694h | #65694h |
| <i>caractère</i>] | #6569Bh | #6569Bh |
| <i>caractère</i> _ | #6568Dh | #6568Dh |
| <i>caractère</i> { | #656A2h | #656A2h |
| <i>caractère</i> } | #656A9h | #656A9h |
| <i>caractère</i>) | #65678h | #65678h |
| CASE (XLIB 1792 25) | #2378Dh | #2378Dh |
| CEIL (XLIB 2 104) | #1BC0Fh | #1BC0Fh |
| CEIL interne → 1: expr. symb. | #551D9h | #551D9h |
| CEIL interne → 1: réel_unité | #0FD36h | #0FD36h |
| CEIL interne → 1: réel | #2AF73h | #2AF73h |
| CENTR | #1E0E8h | #1E0E8h |
| CENTR (XLIB 2 187) | #1E0E8h | #1E0E8h |

| Fonction | Adresse | |
|---|-----------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| CENTR interne →1: complexe | #4AC61h/#4AC61h | |
| CF (XLIB 2 132) | #1C2D5h/#1C2D5h | |
| CF interne →1: réel | #1C2EEh/#1C2EEh | |
| CF interne (sur flag système) →1: entier sys. | #53761h/#53761h | |
| CF interne (sur flag utilisateur) →1: entier sys. | #53755h/#53755h | |
| chaîne → nom global →1: chaîne | #05B15h/#05B15h | |
| Changement de l'en-tête →2: quelconque 1: entier sys.) | #05AB3h/#05AB3h | |
| Changement de l'en-tête →2: quelconque 1: entier sys.) | #05ACCh/#05ACCh | |
| CHR (XLIB 2 165) | #1CB66h/#1CB66h | |
| CHR interne →1: réel | #140F1h/#140F1h | |
| chr (10) (Line Feed) | #65238h/#65238h | |
| chr (27) | #651BEh/#651BEh | |
| CKSM (XLIB 2 369) | #21FECh/#21FECh | |
| CKSM interne →1: réel | #2ED4Ch/#2ED4Ch | |
| CLEAR (XLIB 2 282) | #1FCEBh/#1FCEBh | |
| CLKADJ (XLIB 2 24) | #198DEh/#198DEh | |
| CLKADJ interne | #0CD3Fh/#0CD3Fh | |
| CLLCD (XLIB 2 56) | #1A858h/#1A858h | |
| CLLCD interne | #5046Ah/#5046Ah | |
| CLOSEIO (XLIB 2 362) | #21ED5h/#21ED5h | |
| CLOSEIO interne | #315C6h/#315C6h | |
| CLVAR (XLIB 2 346) | #210FCh/#210FCh | |
| CLΣ (XLIB 2 284) | #1FD2Bh/#1FD2Bh | |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| CLΣ interne | #2C22Fh | #2C22Fh |
| CNRM (XLIB 2 119) | #1BFBEh | #1BFBEh |
| Code de configuration pour la bibliothèque 2 | #189FCh | #189FCh |
| COLCT (XLIB 2 333) | #20A15h | #20A15h |
| COLCT interne →1: réel/complexe/expr. symb. | #57D90h | #57D90h |
| COLΣ (XLIB 2 312) | #2009Ah | #2009Ah |
| COLΣ interne →2: réel/matrice 1: réel | #2C684h | #2C675h |
| COMB (XLIB 2 129) | #1C1F6h | #1C1F6h |
| COMB interne →2: expr. symb. 1: expr. symb. | #55D50h | #55D50h |
| COMB interne →2: expr. symb. 1: réel | #55D1Eh | #55D1Eh |
| COMB interne →2: réel 1: expr. symb. | #55D37h | #55D37h |
| COMB interne →2: réel 1: réel | #2AE62h | #2AE62h |
| CON (XLIB 2 173) | #1D186h | #1D186h |
| CON interne →2: liste 1: complexe | #1D221h | #1D221h |
| CON interne →2: liste 1: réel | #35CAEh | #35CAEh |
| CON interne →2: liste 1: réel/complexe | #1D1EAh | #1D1EAh |
| CON interne →2: nom global 1: complexe | #1D262h | #1D262h |
| CON interne →2: nom global 1: réel | #1D23Fh | #1D23Fh |
| CON interne →2: Nom Local 1: complexe | #1D2ADh | #1D2ADh |
| CON interne →2: Nom Local 1: réel | #1D28Ah | #1D28Ah |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| CONIC (XLIB 2 221) | #1E681h | #1E681h |
| CONJ (XLIB 2 62) | #1AA6Eh | #1AA6Eh |
| CONJ interne →1: expr. symb. | #54F1Dh | #54F1Dh |
| CONJ interne →1: matrice | #35F30h | #35F30h |
| CONJ interne →1: complexe | #51BB2h | #51BB2h |
| CONT (XLIB 2 58) | #1A8BBh | #1A8BBh |
| CONT interne | #11076h | #11076h |
| CONVERT (XLIB 2 11) | #196DBh | #196DBh |
| CONVERT interne | #0F371h | #0F371h |
| <i>Copie des variables du port dans la pile</i> →1: entier sys. | #21922h | #21922h |
| CORR (XLIB 2 289) | #1FDC1h | #1FDC1h |
| CORR interne | #2C84Bh | #2C83Ch |
| COS (XLIB 2 82) | #1B505h | #1B505h |
| COS interne →1: expr. symb. | #54FCCh | #54FCCh |
| COS interne →1: réel long | #2AC57h | #2AC57h |
| COS interne →1: complexe | #52571h | #52571h |
| COS interne →1: réel_unité | #0F660h | #0F660h |
| COS interne →1: réel | #2AC40h | #2AC40h |
| COSH (XLIB 2 85) | #1B606h | #1B606h |
| COSH interne →1: expr. symb. | #55017h | #55017h |
| COSH interne →1: complexe | #52648h | #52648h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/CX |
| COSH interne →1: réel | #2ADDAh | #2ADDAh |
| <i>Coupe la chaîne au saut de ligne(LF)</i> →1: chaîne | #127A7h | #127A7h |
| COV (XLIB 2 290) | #1FDDCh | #1FDDCh |
| COV interne | #2C8F5h | #2C8E6h |
| CR (XLIB 2 243) | #1EEA4h | #1EEA4h |
| CR interne | #31868h | #31868h |
| CRDIR (XLIB 2 32) | #1A105h | #1A105h |
| CROSS (XLIB 2 122) | #1C01Eh | #1C01Eh |
| CROSS interne →2: matrice 1: matrice | #36782h | #36791h |
| D→R (XLIB 2 112) | #1BEC8h | #1BEC8h |
| D→R interne →1: expr. symb. | #55224h | #55224h |
| D→R interne →1: réel | #2A622h | #2A622h |
| DATE (XLIB 2 17) | #19812h | #19812h |
| DATE interne | #0CC0Eh | #0CC0Eh |
| DATE+ (XLIB 2 31) | #199D2h | #199D2h |
| DATE+ interne | #0CC5Bh | #0CC5Bh |
| DDAYS (XLIB 2 30) | #199B2h | #199B2h |
| DDAYS interne | #0CC39h | #0CC39h |
| DEC (XLIB 2 145) | #1C574h | #1C574h |
| DEC interne | #53C5Bh | #53C5Bh |
| <i>Découpage</i> →1: entier sys. <abcde> ← 2: <cde>1: <ab> | #04DD7h | #04DD7h |
| DECR (XLIB 2 332) | #209AAh | #209AAh |
| DECR interne →1: nom global | #209CDh | #209CDh |

| Fonction | Adresse | |
|---|-----------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| DECR interne → 1: Nom Local | #209EBh/#209EBh | |
| DEFINE (XLIB 2 342) | #20D65h/#20D65h | |
| DEFINE interne → 1: Expression Algébrique | #20D7Eh/#20D7Eh | |
| DEFINE interne → 1: nom global/Nom Local | #20DBFh/#20DBFh | |
| DEG (XLIB 2 135) | #1C399h/#1C399h | |
| DEG interne | #2A5D2h/#2A5D2h | |
| DELALARM (XLIB 2 28) | #19972h/#19972h | |
| DELALARM interne | #0E724h/#0E724h | |
| DELAY (XLIB 2 245) | #1EF43h/#1EF43h | |
| DELAY interne → 1: réel | #31FFDh/#31FFDh | |
| DELKEYS (XLIB 2 381) | #22548h/#22548h | |
| DELKEYS interne → 1: liste | #41B3Ch/#41B3Ch | |
| DELKEYS interne → 1: nom global/Nom Local | #41BB9h/#41BB9h | |
| DELKEYS interne → 1: réel | #41B69h/#41B69h | |
| DEPND (XLIB 2 196) | #1E22Bh/#1E22Bh | |
| DEPND interne → 1: liste | #4AFC7h/#4AFC7h | |
| DEPND interne → 1: nom global | #4AFB3h/#4AFB3h | |
| DEPND interne → 1: réel | #47A8Dh/#47A8Dh | |
| DEPTH (XLIB 2 276) | #1FC44h/#1FC44h | |
| DEPTH interne → 1: entier sys. | #0314Ch/#0314Ch | |
| Dérivation Complète (XLIB 2 247) | #1EF7Eh/#1EF7Eh | |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| <i>Dérivation complète interne</i> →2: expr. symb. 1: expr. symb. | #54954h | #54954h |
| <i>Dérivation pas à pas (non évaluée)</i> | #63B6Eh | #63B6Eh |
| <i>Dérivation pas à pas (XLIB 2 248)</i> | #1EFD2h | #1EFD2h |
| <i>Dérivation pas à pas interne</i> →2: expression algébrique 1: expr. symb. | #1F0F5h | #1F0F5h |
| <i>Dernier message d'erreur</i> | #04ED1h | #04ED1h |
| DET (XLIB 2 120) | #1BFDEh | #1BFDEh |
| DET interne →1: matrice | #36A2Ah | #36A48h |
| DETACH (XLIB 2 358) | #2147Ch | #2147Ch |
| DETACH (du répertoire HOME) interne →1: entier sys. | #076AEh | #076AEh |
| DETACH interne →1: entier sys. | #21CE5h | #21CE5h |
| DETACH interne →1: réel | #21495h | #21495h |
| DETACH interne (pas sur HOME) →2: répertoire 1: entier sys. | #21D2Bh | #21D2Bh |
| <i>Destruction des variables locales</i> | #07497h | #07497h |
| <i>Deux entier sys. → Nombres réels</i> | #1950Bh | #1950Bh |
| <i>Deux Nombres réels → Entiers sys.</i> | #194F7h | #194F7h |
| DIR (XLIB 1792 27) | #23813h | #23813h |
| DISP | #1A584h | #1A584h |
| DISP (XLIB 2 50) | #1A584h | #1A584h |
| DISP interne taille 10 →2: chaîne 1: entier sys. | #123C8h | #123C8h |
| DISP interne taille 8 →2: chaîne 1: entier sys. | #12429h | #12429h |
| DISP interne →2: quelconque 1: réel | #140ABh | #140ABh |
| DO (XLIB 1792 7) | #230C3h | #230C3h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| DOERR (XLIB 2 42) | #1A339h | #1A339h |
| DOERR interne → 1: chaîne | #15048h | #15048h |
| DOERR interne → 1: entier bin. | #1501Bh | #1501Bh |
| DOERR interne → 1: entier sys. | #1502Fh | #1502Fh |
| DOERR interne → 1: réel | #15007h | #15007h |
| DOT (XLIB 2 121) | #1BFFEh | #1BFFEh |
| DOT interne → 2: matrice 1: matrice | #366F6h | #36705h |
| DRAX (XLIB 2 193) | #1E1C6h | #1E1C6h |
| DROP (XLIB 2 272) | #1FBD8h | #1FBD8h |
| DROP et vide une chaîne → 2: chaîne 1: quelconque → 1: chaîne "" | #04D57h | #04D57h |
| DROP et DROPN interne | #63FA6h | #63FA6h |
| DROP et DUP interne | #627A7h | #627A7h |
| DROP et NEXT (boucle interne) interne | #63466h | #63466h |
| DROP et OVER interne | #63029h | #63029h |
| DROP et ROT interne | #62FC5h | #62FC5h |
| DROP et SWAP interne | #6270Ch | #6270Ch |
| DROP interne | #03244h | #03244h |
| DROP niveau 3 interne | #60F21h | #60F21h |
| DROP niveau 4 interne | #62864h | #62864h |
| DROP niveau 5 interne | #62880h | #62880h |
| DROP SWAP et DROP interne | #62726h | #62726h |
| DROP2 (XLIB 2 273) | #1FBF3h | #1FBF3h |
| DROP2 (XLIB 2 273) | #1FBF3h | #1FBF3h |
| DROP2 et 0 interne | #1F047h | #1F047h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| DROP2 et FAUX interne | #62B0Bh | #62B0Bh |
| DROP2 interne | #03258h | #03258h |
| DROP3 interne | #60F4Bh | #60F4Bh |
| DROP4 interne | #60F7Eh | #60F7Eh |
| DROP4 interne | #60F83h | #60F83h |
| DROP5 interne | #60F72h | #60F72h |
| DROP6 interne | #60F66h | #60F66h |
| DROP7 interne | #60F54h | #60F54h |
| DROPN (XLIB 2 277) | #1FC64h | #1FC64h |
| DROPN et DROP interne →1: entier sys. | #62F75h | #62F75h |
| DROPN interne →n..2: quelconque 1: entier sys. | #0326Eh | #0326Eh |
| DTAG (XLIB 2 384) | #22633h | #22633h |
| DTAG interne →1: quelconque | #64775h | #64775h |
| DTAG niveau 2 interne | #647A2h | #647A2h |
| DUP (XLIB 2 269) | #1FB87h | #1FB87h |
| DUP +2 et ROLL interne →n..2: quelconque 1: entier sys. | #61305h | #61305h |
| DUP DUP interne | #62CB9h | #62CB9h |
| DUP et +2 interne →1: entier sys. | #626F7h | #626F7h |
| DUP et <0h> interne | #63A88h | #63A88h |
| DUP et <1h> interne | #63A9Ch | #63A9Ch |
| DUP et → compteur de boucle entier sys. | #63411h | #63411h |
| DUP et PICK interne →n..2: quelconque 1: entier sys. | #630DDh | #630DDh |
| DUP et ROLL interne →n..2: quelconque 1: entier sys. | #630F1h | #630F1h |
| DUP et ROT interne | #62FB1h | #62FB1h |

| Fonction | Adresse | |
|---|-----------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| DUP et SIZE interne → 1: chaîne ← 2: chaîne 1: entier sys. | #627BBh/#627BBh | |
| DUP interne | #03188h/#03188h | |
| DUP2 (XLIB 2 270) | #1FBA2h/#1FBA2h | |
| DUP2 et 5 ROLL interne | #63C40h/#63C40h | |
| DUP2 interne | #031ACh/#031ACh | |
| DUPN (XLIB 2 278) | #1FC7Fh/#1FC7Fh | |
| DUPN interne → n..2: quelconque 1: entier sys. | #031D9h/#031D9h | |
| e (2.71828182846) interne | #54D7Bh/#54D7Bh | |
| e (XLIB 2 66) | #1AB23h/#1AB23h | |
| Efface les derniers arguments | #112ECh/#112ECh | |
| ELSE (XLIB 1792 2) | #22FB5h/#22FB5h | |
| ELSE interne | #633B2h/#633B2h | |
| END (XLIB 1792 22) | #23694h/#23694h | |
| END (XLIB 1792 23) | #236B9h/#236B9h | |
| END (XLIB 1792 3) | #22FD5h/#22FD5h | |
| ENG (XLIB 2 140) | #1C452h/#1C452h | |
| ENG interne → 1: entier sys. | #166FBh/#166FBh | |
| ENG interne → 1: réel | #1C46Bh/#1C46Bh | |
| entier sys. → caractère → 1: entier sys. | #05A75h/#05A75h | |
| entier sys. → Binaire → 1: entier sys. | #059CCh/#059CCh | |
| EQ → (XLIB 2 168) | #1CEE3h/#1CEE3h | |
| EQ → interne → 1: Expression Algébrique | #1CF2Eh/#1CF2Eh | |
| ERASE interne | #4B60Ch/#4B60Ch | |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| ERRO (XLIB 2 43) | #1A36Dh/ | #1A36Dh |
| ERRO interne | #1400Eh/ | #1400Eh |
| <i>Erreur "Bad Argument Type"</i> | #18CB2h/ | #18CB2h |
| <i>Erreur "Bad Argument Value"</i> | #16CA7h/ | #16CA7h |
| <i>Erreur "Bad Argument Value"</i> | #18CA2h/ | #18CA2h |
| <i>Erreur "Can't edit Null Char"</i> | #10F54h/ | #10F54h |
| <i>Erreur "Circular Reference"</i> | #11016h/ | #11016h |
| <i>Erreur "Directory not Allowed"</i> | #11026h/ | #11026h |
| <i>Erreur "Directory Recursion"</i> | #04FC2h/ | #04FC2h |
| <i>Erreur "Halt not allowed"</i> | #10FC6h/ | #10FC6h |
| <i>Erreur "Insufficient Memory"</i> | #04FB6h/ | #04FB6h |
| <i>Erreur "Invalid Data Card"</i> | #04FDAh/ | #04FDAh |
| <i>Erreur "Invalid Definition"</i> | #11046h/ | #11046h |
| <i>Erreur "Invalid PPAR"</i> | #10F96h/ | #10F96h |
| <i>Erreur "Invalid Syntax"</i> | #10F86h/ | #10F86h |
| <i>Erreur "Invalide Date"</i> | #0CBB7h/ | #0CBB7h |
| <i>Erreur "Invalide Time"</i> | #0CBC4h/ | #0CBC4h |
| <i>Erreur "Invalide user Function"</i> | #10F64h/ | #10F64h |
| <i>Erreur "Last Command Disabled"</i> | #10FF6h/ | #10FF6h |
| <i>Erreur "Last Stack Disabled"</i> | #10FE6h/ | #10FE6h |
| <i>Erreur "Missing Library"</i> | #11056h/ | #11056h |
| <i>Erreur "Name Conflict"</i> | #11066h/ | #11066h |
| <i>Erreur "No current Equation"</i> | #10F74h/ | #10F74h |
| <i>Erreur "No room in port"</i> | #04FFEh/ | #04FFEh |
| <i>Erreur "Non Empty Directory"</i> | #11036h/ | #11036h |
| <i>Erreur "Non existent ΣDAT"</i> | #2D2E6h/ | - |
| <i>Erreur "Non Real Result"</i> | #10FA6h/ | #10FA6h |
| <i>Erreur "Nonexistent Alarm"</i> | #0CBAEh/ | #0CBAEh |
| <i>Erreur "Object in Use"</i> | #04FE6h/ | #04FE6h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| Erreur "Object not in port" | #0500Ah | #0500Ah |
| Erreur "Port not Available" | #04FF2h | #04FF2h |
| Erreur "Power lost" | #04FAAh | #04FAAh |
| Erreur "Too Few Arguments" | #18CC2h | #18CC2h |
| Erreur "Unable to Isolate" | #10FB6h | #10FB6h |
| Erreur "Undefined Local Name" | #04FCEh | #04FCEh |
| Erreur "Undefined Name" | #18C92h | #18C92h |
| Erreur "Undefined Xlib Name" | #05016h | #05016h |
| Erreur "Wrong Argument Count" | #11006h | #11006h |
| ERRM (XLIB 2 45) | #1A3A3h | #1A3A3h |
| ERRM interne | #14065h | #14065h |
| ERRN (XLIB 2 44) | #1A388h | #1A388h |
| ERRN interne | #1404Ch | #1404Ch |
| EVAL (XLIB 2 46) | #1A3BEh | #1A3BEh |
| EVAL interne → 1: Algèbrique/liste/Programme | #18EBAh | #18EBAh |
| EVAL interne → 1: Programme | #06F9Fh | #06F9Fh |
| EVAL interne → 1:quelconque sauf Expr. symb./liste/Signé | #06F8Eh | #06F8Eh |
| EVAL interne → 1: Signé | #217C7h | #217C7h |
| EXP (XLIB 2 93) | #1B905h | #1B905h |
| EXP interne → 1: expr. symb. | #550DFh | #550DFh |
| EXP interne → 1: réel long | #2AB1Ch | #2AB1Ch |
| EXP interne → 1: complexe | #52193h | #52193h |
| EXP interne → 1: réel | #2AB2Fh | #2AB2Fh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| EXPAN (XLIB 2 334) | #20A49h | #20A49h |
| EXPAN interne → 1: réel/complexe/expr. symb. | #57A0Ch | #57A0Ch |
| EXPFIT (XLIB 2 321) | #201FBh | #201FBh |
| EXPM (XLIB 2 98) | #1BAC2h | #1BAC2h |
| EXPM interne → 1: expr. symb. | #5515Ch | #5515Ch |
| EXPM interne → 1: réel | #2AB42h | #2AB42h |
| <i>Exprime le résultat en réel_unités MKSA</i> | #0F561h | #0F561h |
| FACT (XLIB 2 100) | #1BB41h | #1BB41h |
| FAUX (false) | #03AC0h | #03AC0h |
| FAUX et VRAI interne | #6350Bh | #6350Bh |
| FC? (XLIB 2 134) | #1C360h | #1C360h |
| FC? interne → 1: réel | #1C379h | #1C379h |
| FC?C (XLIB 2 143) | #1C520h | #1C520h |
| FC?C interne → 1: réel | #1C539h | #1C539h |
| FINDALARM (XLIB 2 27) | #19948h | #19948h |
| FINDALARM interne → 1: List | #0EB31h | #0EB31h |
| FINDALARM interne → 1: réel | #0EAD7h | #0EAD7h |
| FINISH (XLIB 2 367) | #21FB6h | #21FB6h |
| FINISH interne | #2E876h | #2E876h |
| FIX (XLIB 2 138) | #1C3EAh | #1C3EAh |
| FIX interne → 1: réel | #1C403h | #1C403h |
| FLOOR (XLIB 2 103) | #1BBD9h | #1BBD9h |
| FLOOR interne → 1: expr. symb. | #551C0h | #551C0h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| FLOOR interne →1: réel long | #2AF99h | #2AF99h |
| FLOOR interne →1: réel_unité | #0FD22h | #0FD22h |
| FLOOR interne →1: réel | #2AF86h | #2AF86h |
| FOR (XLIB 1792 10) | #231A0h | #231A0h |
| For 0 to premier élément de la pile -1 →1: entier sys. | #073C3h | #073C3h |
| For 1 to premier élément de la pile -1 →1: entier sys. | #073CEh | #073CEh |
| For 1 to premier élément de la pile →1: entier sys. | #073DBh | #073DBh |
| FOR interne →2: expr. symb. 1: réel | #2322Ch | #2322Ch |
| FOR interne →2: réel/expr. symb. 1: expr. symb. | #23213h | #23213h |
| FP (XLIB 2 102) | #1BBA3h | #1BBA3h |
| FP interne →1: réel_unité | #0FD0Eh | #0FD0Eh |
| FP interne →1: réel | #2AF4Dh | #2AF4Dh |
| FP interne →1: Symbolic | #551A7h | #551A7h |
| FREE (XLIB 2 355) | #213D1h | #213D1h |
| FREE interne →2: liste 1: réel | #21B74h | #21B74h |
| FREE interne →2: réel/nom global/Nom Local 1: réel | #21408h | #21408h |
| FREEZE | #1A5A4h | #1A5A4h |
| FREEZE (zone d'état de l'écran) | #393D3h | #393D3h |
| FREEZE (barre de menu) | #39451h | #39451h |
| FREEZE (pile) | #393FDh | #393FDh |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| FREEZE (tout l'écran) | #391EEh | #391EEh |
| FREEZE (XLIB 2 51) | #1A5A4h | #1A5A4h |
| FREEZE interne → 1: réel | #142FBh | #142FBh |
| FS? (XLIB 2 133) | #1C313h | #1C313h |
| FS? interne → 1: réel | #1C32Ch | #1C32Ch |
| FS? interne → 1: réel → 1: booléen | #1C331h | #1C331h |
| FS? interne (sur flag système) → 1: entier sys. → 1: booléen | #53784h | #53784h |
| FS? interne (sur flag utilisateur) → 1: entier sys. → 1: booléen | #53778h | #53778h |
| FS?C (XLIB 2 142) | #1C4A1h | #1C4A1h |
| FS?C interne → 1: réel | #1C4BAh | #1C4BAh |
| FS?C interne → 1: réel → 1: booléen | #1C4BFh | #1C4BFh |
| FUNCTION (XLIB 2 220) | #1E661h | #1E661h |
| GET (XLIB 2 178) | #1D7C6h | #1D7C6h |
| GET interne → 2: liste 1: réel/liste | #1D898h | #1D898h |
| GET interne → 2: matrice 1: réel/liste | #1D86Bh | #1D86Bh |
| GET interne → 2: nom global/Nom Local 1: réel/liste | #1D825h | #1D825h |
| GET interne sur un objet quelconque | #056B6h | #056B6h |
| GETI (XLIB 2 179) | #1D8C7h | #1D8C7h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 C/GX |
| GETI interne →2: liste 1: réel/liste | #1D9BCh | #1D9BCh |
| GETI interne →2: matrice 1: réel/liste | #1D96Ch | #1D96Ch |
| GETI interne →2: nom global/Nom Local 1: réel/liste | #1D926h | #1D926h |
| GOR | #1E456h | #1E456h |
| GOR (XLIB 2 211) | #1E456h | #1E456h |
| GOR interne →3: Graphique 2: liste 1: Graphique | #1E46Ah | #1E46Ah |
| GOR interne →3: Graphique 2: complexe 1: Graphique | #1E488h | #1E488h |
| GOR interne →3: PICT 2: complexe 1: Graphique | #1E4C4h | #1E4C4h |
| GOR/GXOR interne →4:booléen 3:Graphique 2:liste 1:Graphique | #4F6BAh | #4F6BAh |
| GOR/GXOR interne →4:booléen 3:Graphique 2:Cplx 1:Graphique | #4F6F6h | #4F6F6h |
| GOR/GXOR interne →4:booléen 3:PICT 2:Cplx/liste 1:Graphique | #4F741h | #4F741h |
| GRAD (XLIB 2 137) | #1C3CFh | #1C3CFh |
| GRAD interne | #2A604h | #2A604h |
| GRAPH | #1E2BAh | #1E2BAh |
| GRAPH (XLIB 2 200) | #1E2BAh | #1E2BAh |
| GRAPH interne | #4D1AAh | #4D1AAh |
| Graphic 0x0 | #505B2h | #505B2h |
| GXOR (XLIB 2 212) | #1E4E4h | #1E4E4h |
| HALT (XLIB 1792 14) | #23472h | #23472h |
| HALT interne | #14378h | #14378h |
| HEX (XLIB 2 146) | #1C58Fh | #1C58Fh |
| HEX interne | #53C37h | #53C37h |
| HISTOGRAM (XLIB 2 226) | #1E721h | #1E721h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| HISTPLOT (XLIB 2 317) | #20167h | #20167h |
| HMS+ (XLIB 2 116) | #1BF5Eh | #1BF5Eh |
| HMS+ interne →2: réel 1: réel | #2A6A0h | #2A6A0h |
| HMS- (XLIB 2 117) | #1BF7Eh | #1BF7Eh |
| HMS- interne →2: réel 1: réel | #2A6C8h | #2A6C8h |
| HMS→ (XLIB 2 115) | #1BF3Eh | #1BF3Eh |
| HMS→ interne →1: réel | #2A68Ch | #2A68Ch |
| HOME (XLIB 2 34) | #1A140h | #1A140h |
| HOME interne | #08D92h | #08D92h |
| <i>Horloge affichée</i> | #53AACh | #53AACh |
| <i>Horloge non affichée</i> | #53ABAh | #53ABAh |
| i (0,1) Interne | #54D58h | #54D58h |
| i (XLIB 2 67) | #1AB45h | #1AB45h |
| IDN (XLIB 2 174) | #1D2DCh | #1D2DCh |
| IDN interne →1: matrice | #35D35h | #35D35h |
| IDN interne →1: nom global | #1D34Ah | #1D34Ah |
| IDN interne →1: Nom Local | #1D36Dh | #1D36Dh |
| IDN interne →1: réel | #1D313h | #1D313h |
| IF (XLIB 1792 0) | #22EC3h | #22EC3h |
| IFERR (XLIB 1792 13) | #233DFh | #233DFh |
| IFT (XLIB 2 48) | #1A4CDh | #1A4CDh |
| IFT interne →2: expr. symb. 1: quelconque | #1A513h | #1A513h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| IFT interne →2: réel 1: quelconque | #1A4F0h | #1A4F0h |
| IFTE (XLIB 2 47) | #1A3FEh | #1A3FEh |
| IFTE interne →3: expr.symb. 2/1: réel/ complexe/ expr.symb. | #54565h | #54565h |
| IFTE interne →3: réel 2: quelconque 1: quelconque | #1A4A3h | #1A4A3h |
| IM (XLIB 2 155) | #1C819h | #1C819h |
| IM interne →1: expr. symb. | #54EB9h | #54EB9h |
| IM interne →1: matrice | #35FEEh | #35FEEh |
| IM interne →1: complexe | #519B7h | #519B7h |
| IM interne →1: réel | #5198Fh | #5198Fh |
| INCR (XLIB 2 331) | #208F4h | #208F4h |
| INCR interne →1: nom global | #20917h | #20917h |
| INCR interne →1: nom local | #20980h | #20980h |
| INDEP (XLIB 2 183) | #1E04Ah | #1E04Ah |
| INDEP interne →1: liste | #4AF8Bh | #4AF8Bh |
| INDEP interne →1: nom global | #4AF77h | #4AF77h |
| INDEP interne →1: réel | #47A6Ah | #47A6Ah |
| INPUT (XLIB 2 378) | #224CAh | #224CAh |
| INPUT interne →2: chaîne 1: chaîne | #43395h | #43395h |
| INPUT interne →2: chaîne 1: liste | #433CCh | #433CCh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| INV (XLIB 2 76) | #1B278h | #1B278h |
| INV interne →1: expr. symb. | #54F36h | #54F36h |
| INV interne →1: réel long | #2AA92h | #2AA92h |
| INV interne →1: matrice | #36B0Bh | #36A99h |
| INV interne →1: complexe | #51EFAh | #51EFAh |
| INV interne →1: réel_unité | #0F841h | #0F841h |
| INV interne →1: réel | #2AAAFh | #2AAAFh |
| IP (XLIB 2 101) | #1BB6Dh | #1BB6Dh |
| IP interne →1: expr. symb. | #5518Eh | #5518Eh |
| IP interne →1: réel_unité | #0FCFAh | #0FCFAh |
| IP interne →1: réel | #2AF60h | #2AF60h |
| ISOL (XLIB 2 336) | #20A93h | #20A93h |
| ISOL interne →2: expr. symb. 1: nom global | #572A2h | #57293h |
| <i>Itération de la boucle</i> | #06FB7h | #06FB7h |
| KERRM (XLIB 2 373) | #2206Ch | #2206Ch |
| KERRM interne | #2EDA6h | #2EDA6h |
| KEY (XLIB 2 57) | #1A873h | #1A873h |
| KGET (XLIB 2 364) | #21F24h | #21F24h |
| KGET interne →1: chaîne/nom global/Nom Local | #2E7EFh | #2E7EFh |
| KGET interne →1: List | #2E835h | #2E835h |
| KILL (XLIB 2 40) | #1A303h | #1A303h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| KILL interne | #10FD6h | #10FD6h |
| LABEL (XLIB 2 201) | #1E2D5h | #1E2D5h |
| LABEL interne | #4E875h | #4E889h |
| LASTARG (XLIB 2 54) | #1A604h | #1A604h |
| LCD → (XLIB 2 213) | #1E572h | #1E572h |
| LCD → interne | #503D4h | #503D4h |
| LIBS (XLIB 2 356) | #2142Dh | #2142Dh |
| LIBS interne | #21D54h | #21D54h |
| LINE (XLIB 2 206) | #1E398h | #1E398h |
| LINE interne →2: liste 1: liste | #4F525h | #4F525h |
| LINE interne →2: complexe 1: complexe | #4F584h | #4F584h |
| LINFIT (XLIB 2 319) | #201B1h | #201B1h |
| LIST → (XLIB 2 158) | #1C95Ah | #1C95Ah |
| LIST → interne →1: List → n..2: quelconque 1: entier sys. | #62C41h | #62C41h |
| LIST → interne →1: liste → n..1: contenu | #62B88h | #62B88h |
| LN (XLIB 2 94) | #1B94Fh | #1B94Fh |
| LN interne →1: expr. symb. | #550F8h | #550F8h |
| LN interne →1: réel long | #2AB5Bh | #2AB5Bh |
| LN interne →1: réel >0 | #2AB6Eh | #2AB6Eh |
| LN interne →1: réel | #1B995h | #1B995h |
| LNP1 (XLIB 2 97) | #1BA8Ch | #1BA8Ch |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| LNP1 interne → 1: expr. symb. | #55143h | #55143h |
| LNP1 interne → 1: réel long | #2AB94h | #2AB94h |
| LNP1 interne → 1: réel | #2ABA7h | #2ABA7h |
| LOG (XLIB 2 95) | #1B9C6h | #1B9C6h |
| LOG interne → 1: expr. symb. | #55111h | #55111h |
| LOG interne → 1: complexe | #522BFh | #522BFh |
| LOG interne → 1: réel >0 | #2AB81h | #2AB81h |
| LOG interne → 1: réel | #1BA0Ch | #1BA0Ch |
| LOGFIT (XLIB 2 320) | #201D6h | #201D6h |
| <i>Long complexe</i> → 2 Longs réels → 1: Long complexe | #05DBCh | #05DBCh |
| <i>réel long</i> → <i>réel (interne)</i> → 1: réel long | #2A5B0h | #2A5B0h |
| <i>Longs réels</i> → <i>Long complexe</i> → 2,1: Nombres réels longs | #05C8Ah | #05C72h |
| LR (XLIB 2 302) | #1FF20h | #1FF20h |
| LR interne | #2CA30h | #647BBh |
| MANT (XLIB 2 111) | #1BE9Ch | #1BE9Ch |
| MANT interne → 1: expr. symb. | #5520Bh | #5520Bh |
| MANT interne → 1: réel | #2A930h | #2A930h |
| MAX (XLIB 2 106) | #1BC71h | #1BC71h |
| MAX interne → 2: expr. symb. 1: expr. symb. | #55E7Ch | #55E7Ch |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| MAX interne →2: expr. symb. 1: réel/réel_unité | #55E4Ah | #55E4Ah |
| MAX interne →2: réel 1: réel | #2A6F5h | #2A6F5h |
| MAX interne →2:réel/réel_unité 1: réel/réel_unité | #0FB6Fh | #0FB6Fh |
| MAXR | #5251Ch | #5251Ch |
| MAXR (XLIB 2 64) | #1AADFh | #1AADFh |
| MAX Σ (XLIB 2 296) | #1FE7Eh | #1FE7Eh |
| MAX Σ interne | #2C558h | #2C558h |
| MEAN (XLIB 2 297) | #1FE99h | #1FE99h |
| MEAN interne | #2C571h | #2C571h |
| MEM (XLIB 2 344) | #20FAAh | #20FAAh |
| MENU (XLIB 2 348) | #21196h | #21196h |
| MERGE (XLIB 3 354) | #2137Fh | #2137Fh |
| MERGE interne →1: réel | #21398h | #21398h |
| MIN (XLIB 2 107) | #1BCE3h | #1BCE3h |
| MIN interne →2: expr. symb. 1: expr. symb. | #55EC7h | #55EC7h |
| MIN interne →2: réel 1: réel | #2A70Eh | #2A70Eh |
| MIN interne →2:réel/réel_unité 1: expr. symb. | #55EAEh | #55EAEh |
| MIN interne →2:réel/réel_unité 1:réel/réel_unité | #0FB8Dh | #0FB8Dh |
| MINR (XLIB 2 65) | #1AB01h | #1AB01h |
| MINR interne (1.E-499) | #54CDBh | #54CDBh |
| MIN Σ (XLIB 2 298) | #1FEB4h | #1FEB4h |
| MIN Σ interne | #2C58Ah | #2C58Ah |
| MOD (XLIB 2 110) | #1BE4Dh | #1BE4Dh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| MOD <i>interne</i> → 2: expr. symb. 1: expr. symb. | #56089h | #56089h |
| MOD <i>interne</i> → 2: expr. symb. 1: réel | #56057h | #56057h |
| MOD <i>interne</i> → 2: réel 1: expr. symb. | #56070h | #56070h |
| MOD <i>interne</i> → 2: réel 1: réel | #2ABDCh | #2ABDCh |
| NEG (XLIB 2 60) | #1A995h | #1A995h |
| NEG <i>interne</i> → 1: entier bin. | #53EC3h | #53EC3h |
| NEG <i>interne</i> → 1: expr. symb. | #54EEBh | #54EEBh |
| NEG <i>interne</i> → 1: Graphique | #4FC28h | #4FC28h |
| NEG <i>interne</i> → 1: matrice | #35DEBh | #35DEBh |
| NEG <i>interne</i> → 1: complexe | #51B70h | #51B70h |
| NEG <i>interne</i> → 1: réel_unité | #0F615h | #0F615h |
| NEG <i>interne</i> → 1: PICT | #4FC3Ch | #4FC3Ch |
| NEG <i>interne</i> → 1: réel | #2A920h | #2A920h |
| NEWOB (XLIB 2 39) | #1A2BCh | #1A2BCh |
| NEWOB <i>et SWAP</i> <i>interne</i> | #62C69h | #62C69h |
| NEWOB <i>interne</i> | #06657h | #06657h |
| Next (<i>boucle interne</i>) | #07334h | #07334h |
| NEXT (XLIB 1792 11) | #2324Ch | #2324Ch |
| NEXT <i>interne</i> | #2326Ah | #2326Ah |
| <i>Nom Local ou Global</i> → chaîne | #05BE9h | #05BE9h |

| Fonction | Adresse | |
|--|-------------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| Nombre d'éléments d'une matrice → 1: matrice → 1: entier sys. | #03562h / #03562h | |
| NOT interne → 1: booléen | #03AF2h / #03AF2h | |
| NOT interne → 1: chaîne | #188D2h / #188D2h | |
| NOT interne → 1: chaîne | #18961h / #18961h | |
| NOT interne → 1: entier bin. | #53D4Eh / #53D4Eh | |
| NOT interne → 1: expr. symb. | #54ED2h / #54ED2h | |
| NOT interne → 1: réel | #1E8D9h / #1E8D9h | |
| NUM (XLIB 2 164) | #1CB46h / #1CB46h | |
| NUM interne → 1: chaîne | #1410Fh / #1410Fh | |
| $N\sum$ (XLIB 2 288) | #1FDA6h / #1FDA6h | |
| $N\sum$ interne | #2C535h / #2C535h | |
| OBJ→ (XLIB 2 169) | #1CF7Bh / #1CF7Bh | |
| OBJ→ interne → 1: Signé | #05EC7h / #05EC9h | |
| OBJ→ interne → 1: Expression Algébrique | #1CFD0h / #1CFD0h | |
| OBJ→ interne → 1: réel_unité | #0F34Eh / #0F34Eh | |
| OBJ→ interne → 1: Signé | #05EC7h / #05EC9h | |
| OCT (XLIB 2 147) | #1C5AAh / #1C5AAh | |
| OCT interne | #53C4Fh / #53C4Fh | |
| OFF (XLIB 2 41) | #1A31Eh / #1A31Eh | |
| OFF interne | #041A7h / #041A7h | |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| OFF interne | #041EDh | #041EDh |
| OFF interne | #3A9CEh | #3A9CEh |
| OFF interne | #041A7h | #041A7h |
| OFF interne | #041EDh | #041EDh |
| OLDPRT (XLIB 2 239) | #1EE38h | #1EE38h |
| OLDPRT interne | #31DABh | #31DABh |
| OPENIO (XLIB 2 361) | #21EB5h | #21EB5h |
| OR (XLIB 2 230) | #1E809h | #1E809h |
| OR interne →2: chaîne 1: chaîne | #18887h | #18887h |
| OR interne →2: chaîne 1: chaîne | #188F5h | #188F5h |
| OR interne →2: entier bin. 1: entier bin. | #53D15h | #53D15h |
| OR interne →2: expr. symb. 1: expr. symb. | #55A17h | #55A17h |
| OR interne →2: réel 1: expr. symb. | #559FEh | #559FEh |
| OR interne →2: réel 1: réel | #1E863h | #1E863h |
| ORDER →1: liste | #20FF2h | #20FF2h |
| ORDER (XLIB 2 345) | #20FD9h | #20FD9h |
| OVER (XLIB 2 275) | #1FC29h | #1FC29h |
| OVER et 5 PICK interne | #63C90h | #63C90h |
| OVER et DUP interne | #62CCDh | #62CCDh |
| OVER et OVER interne | #63C68h | #63C68h |
| OVER et SIZE interne →2: chaîne 1:quelconque → 1: entier sys. | #05622h | #05622h |
| OVER et SWAP interne | #62D31h | #62D31h |
| OVER et SWAP interne | #62EDFh | #62EDFh |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| OVER interne | #032C2h | #032C2h |
| PARAMETRIC (XLIB 2 223) | #1E6C1h | #1E6C1h |
| PARITY (XLIB 2 371) | #2202Ch | #2202Ch |
| PARITY interne →1: réel | #2ECCAh | #2ECCAh |
| PATH (XLIB 2 33) | #1A125h | #1A125h |
| PATH interne | #1848Ch | #1848Ch |
| PDIM interne →2: entier bin. 1: entier bin. | #4B300h | #4B300h |
| PDIM interne →2: entier sys. 1: entier sys. | #4B323h | #4B323h |
| PDIM interne →2: complexe 1: Complexe | #4B206h | #4B206h |
| PERM (XLIB 2 130) | #1C236h | #1C236h |
| PERM interne →2: expr. symb. 1: expr. symb. | #55D9Bh | #55D9Bh |
| PERM interne →2: expr. symb. 1: réel | #55D69h | #55D69h |
| PERM interne →2: réel 1: expr. symb. | #55D82h | #55D82h |
| PERM interne →2: réel 1: réel | #2AE75h | #2AE75h |
| PGDIR (XLIB 2 351) | #2123Ah | #2123Ah |
| PGDIR interne →1: nom global | #18595h | #18595h |
| PICK (XLIB 2 279) | #1FC9Ah | #1FC9Ah |
| PICT (XLIB 2 210) | #1E436h | #1E436h |
| PIX? (XLIB 2 205) | #1E36Eh | #1E36Eh |
| PIX? interne →1: liste | #4F4BCh | #4F4BCh |
| PIX? interne →1: complexe | #4F4A3h | #4F4A3h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| PIXOFF (XLIB 2 204) | #1E344h | #1E344h |
| PIXOFF interne →1: liste | #4F48Ah | #4F48Ah |
| PIXOFF interne →1: complexe | #4F471h | #4F471h |
| PIXON (XLIB 2 203) | #1E31Ah | #1E31Ah |
| PIXON interne →1: liste | #4F458h | #4F458h |
| PIXON interne →1: complexe | #4F3EFh | #4F3EFh |
| PKT (XLIB 2 377) | #220DDh | #220DDh |
| PKT interne →2: chaîne 1: chaîne | #2E8D1h | #2E8D1h |
| PMAX (XLIB 2 185) | #1E09Eh | #1E09Eh |
| PMAX interne →1: complexe | #4B0C6h | #4B0C6h |
| PMIN (XLIB 2 184) | #1E07Eh | #1E07Eh |
| PMIN interne →1: complexe | #4B09Eh | #4B09Eh |
| POLAR (XLIB 2 222) | #1E6A1h | #1E6A1h |
| POS →2: chaîne 1: chaîne | #1CAD7h | #1CAD7h |
| POS (XLIB 2 161) | #1CAB4h | #1CAB4h |
| POS interne →2: List 1: quelconque | #1CAF0h | #1CAF0h |
| PR1 (XLIB 2 240) | #1EE53h | #1EE53h |
| PR1 interne | #318FEh | #318FEh |
| PREDV (XLIB 2 303) | #1FF7Ah | #1FF7Ah |
| PREDX (XLIB 2 305) | #1FFBAh | #1FFBAh |
| PREDX interne →1: réel | #2CB75h | #2CB52h |
| PREDY (XLIB 2 304) | #1FF9Ah | #1FF9Ah |

| Fonction | Adresse | |
|---|-----------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| PREDY interne →1: réel | #2CB02h/#2CADFh | |
| <i>Premier caractère d'une chaîne</i> →1: chaîne ← 1: caractère | #050EDh/#050EDh | |
| PRLCD (XLIB 2 246) | #1EF63h/#1EF63h | |
| PRLCD interne | #31EE2h/#31EE2h | |
| <i>Programme vide</i> | #40788h/#40788h | |
| PROMPT (XLIB 1792 28) | #23824h/#23824h | |
| PRST (XLIB 2 242) | #1EE89h/#1EE89h | |
| PRST interne | #31A25h/#31A25h | |
| PRSTC (XLIB 2 241) | #1EE6Eh/#1EE6Eh | |
| PRSTC interne | #318A4h/#318A4h | |
| PRVAR (XLIB 2 244) | #1EEBFh/#1EEBFh | |
| PRVAR interne →1: liste | #1EF1Eh/#1EF1Eh | |
| PRVAR interne →1: nom global/Nom Local | #31D56h/#31D56h | |
| PRVAR interne →1: Signé | #1EECh/#1EECh | |
| PURGE (XLIB 2 343) | #20EFEh/#20EFEh | |
| PURGE interne →1: liste | #20F35h/#20F35h | |
| PURGE interne →1: nom global | #1854Fh/#1854Fh | |
| PURGE interne →1: PICT | #20F8Ah/#20F8Ah | |
| PURGE interne →1: Signé | #217F1h/#217F1h | |
| PUT (XLIB 2 176) | #1D407h/#1D407h | |
| PUT interne →3:liste 2:réel/liste 1:quelconque | #1D524h/#1D524h | |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| PUT interne → 3:matrice 2:réel/liste 1:Nombre | #1D4DEh | #1D4DEh |
| PUT interne → 3:nom global 2:réel/liste 1:quelconque | #1D484h | #1D484h |
| PUT interne → 3:Nom Local 2:réel/liste 1:quelconque | #1D565h | #1D565h |
| PUTI (XLIB 2 177) | #1D5DFh | #1D5DFh |
| PUTI interne → 3:List 2:réel/liste 1:quelconque | #1D701h | #1D701h |
| PUTI interne → 3:matrice 2:réel/liste 1:Nombre | #1D6B6h | #1D6B6h |
| PUTI interne → 3:nom global 2:réel/liste 1:quelconque | #1D65Ch | #1D65Ch |
| PUTI interne → 3:Nom Local 2:réel/liste 1:quelconque | #1D747h | #1D747h |
| PVARS (XLIB 2 350) | #211FCh | #211FCh |
| PVIEW (XLIB 2 202) | #1E2F0h | #1E2F0h |
| PVIEW interne → 1: liste | #4F02Fh | #4F027h |
| PVIEW interne → 1: complexe | #4F011h | #4F009h |
| PWRFIT (XLIB 2 322) | #20220h | #20220h |
| PX→C (XLIB 2 198) | #1E27Ah | #1E27Ah |
| PX→C interne → 1: liste | #4F0ACh | #4F0ACh |
| QUAD (XLIB 2 337) | #20AB3h | #20AB3h |
| QUAD interne → 2: expr. symb. 1: nom global | #591ADh | #591ADh |
| QUOTE (XLIB 2 257) | #1F500h | #1F500h |
| QUOTE interne → 1: expression algébrique | #1F542h | #1F542h |
| RAD (XLIB 2 136) | #1C3B4h | #1C3B4h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| RAD interne | #2A5F0h | #2A5F0h |
| R→B (XLIB 2 9) | #1969Bh | #1969Bh |
| R→B interne | #543F9h | #543F9h |
| R→C (XLIB 2 153) | #1C79Eh | #1C79Eh |
| R→C interne | #36039h | #36039h |
| →2: matrice 1: matrice | | |
| R→C interne | #05C27h | #05C27h |
| →2: réel 1: réel | | |
| R→D (XLIB 2 113) | #1BEF4h | #1BEF4h |
| R→D interne | #5523Dh | #5523Dh |
| →1: expr. symb. | | |
| R→D interne | #2A655h | #2A655h |
| →1: réel | | |
| R→SB et DUP interne | #62CE1h | #62CE1h |
| RAND (XLIB 2 127) | #1C1B9h | #1C1B9h |
| RAND interne | #2AFC2h | #2AFC2h |
| <i>Rappel du répertoire en cours</i> | #08D5Ah | #08D5Ah |
| <i>Rappel du répertoire HOME</i> | #08D82h | #08D82h |
| <i>Rappel de la n-ième Alarme</i> | #0E402h | #0E402h |
| →1: entier sys. | | |
| <i>Rappel des 5 quartets à l'adresse #70673h</i> | #04CE6h | #04CE6h |
| <i>Rappel du GROB menu (131x8)</i> | #12645h | #12645h |
| <i>Rappel des drapeaux système</i> | #1C637h | #1C637h |
| <i>Rappel des drapeaux utilisateur</i> | #1C64Eh | #1C64Eh |
| <i>Rappel de PICT</i> | #12665h | #12665h |
| RATIO (XLIB 2 268) | #1FB5Dh | #1FB5Dh |
| RCEQ (XLIB 2 249) | #1F133h | #1F133h |
| RCEQ interne | #1572Bh | #1572Bh |
| RCEQ interne | #15744h | #15744h |
| ← Contenu + booléen | | |
| RCL (XLIB 2 340) | #20B40h | #20B40h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| RCL interne →1: liste | #20B9Ah | #20B9Ah |
| RCL interne →1: nom global/Nom Local | #20B81h | #20B81h |
| RCL interne →1: nom global/Nom Local ← Contenu, booléen | #62A34h | #62A34h |
| RCL interne →1: PICT | #20CADh | #20CADh |
| RCL interne →1: Signé | #21761h | #21761h |
| RCLALARM (XLIB 2 26) | #19928h | #19928h |
| RCLALARM interne | #0E3DFh | #0E3DFh |
| RCLF (XLIB 2 150) | #1C619h | #1C619h |
| RCLKEYS (XLIB 2 382) | #22586h | #22586h |
| RCLMENU interne | #415C9h | #415C9h |
| RCLΣ (XLIB 2 285) | #1FD46h | #1FD46h |
| RCLΣ interne | #2C2ACh | #2C2ACh |
| RCLΣ interne ← Contenu et booléen | #2C293h | #2C293h |
| RCWS (XLIB 2 149) | #1C5FEh | #1C5FEh |
| RCWS interne ← entier sys. | #54039h | #54039h |
| RCWS interne | #53CF0h | #53CF0h |
| RDM (XLIB 2 172) | #1D0DFh | #1D0DFh |
| RDM interne →2: matrice 1: liste | #1D10Ch | #1D10Ch |
| RDM interne →2: nom global 1: liste | #1D125h | #1D125h |
| RDM interne →2: Nom Local 1: liste | #1D152h | #1D152h |
| RDZ (XLIB 2 128) | #1C1D4h | #1C1D4h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| RDZ interne →1: réel | #2B044h | #2B044h |
| RE (XLIB 2 154) | #1C7CAh | #1C7CAh |
| RE interne →1: expr. symb. | #54EA0h | #54EA0h |
| RE interne →1: matrice | #35F8Fh | #35F8Fh |
| RE interne →1: complexe | #519A3h | #519A3h |
| RECN (XLIB 2 365) | #21F62h | #21F62h |
| RECN interne →1: chaîne/nom global/Nom Local | #2D816h | #2D816h |
| RECV (XLIB 2 366) | #21F96h | #21F96h |
| <i>réel</i> → entier bin. | #54422h | #54422h |
| <i>réel</i> → réel long (interne) | #2A5C1h | #2A5C1h |
| <i>réel</i> → complexe | #51A37h | #51A37h |
| <i>réel</i> → entier sys. et SWAP interne | #62E7Bh | #62E7Bh |
| Remplace premier élément de la pile par FAUX | #620A0h | #620A0h |
| Remplace premier élément de la pile par VRAI | #62080h | #62080h |
| REPEAT (XLIB 1792 6) | #2305Dh | #2305Dh |
| REPEAT interne →1: expr. symb. | #230A3h | #230A3h |
| REPEAT interne →1: réel | #23085h | #23085h |
| REPL (XLIB 2 157) | #1C8EAh | #1C8EAh |
| REPL interne →3:chaîne 2:réel 1:chaîne | #4FAF7h | #4FAF7h |
| REPL interne →3:Graphique 2:liste 1:Graphique | #4F999h | #4F999h |
| REPL interne →3:Graphique 2:complexe 1:Graphique | #4F9F3h | #4F9F3h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| REPL interne → 3:liste 2:réel 1:liste | #4FA7Ah | #4FA7Ah |
| REPL interne → 3:PICT 2:liste/complexe 1:Graphique | #4FA2Fh | #4FA2Fh |
| RES interne → 1: réel | #4AFEfH | #4AFEfH |
| RES interne → 1: réel>0/entier bin. | #4B012h | #4B012h |
| RESTORE (XLIB 2 353) | #2133Ch | #2133Ch |
| RESTORE interne → 1: Sauvegarde | #21B2Fh | #21B2Fh |
| RL (XLIB 2 1) | #1959Bh | #1959Bh |
| RL interne | #53E0Ch | #53E0Ch |
| RLB (XLIB 2 2) | #195BBh | #195BBh |
| RLB interne | #53E3Bh | #53E3Bh |
| RND (XLIB 2 108) | #1BD55h | #1BD55h |
| RND interne → 2: expr. symb. 1: expr. symb. | #55DE6h | #55DE6h |
| RND interne → 2: expr. symb. 1: réel | #55DB4h | #55DB4h |
| RND interne → 2: matrice 1: réel | #35E2Ch | #35E2Ch |
| RND interne → 2: complexe 1: réel | #35EC2h | #35EC2h |
| RND interne → 2: réel_unité 1: réel | #0FD68h | #0FD68h |
| RND interne → 2: réel 1: réel | #2B529h | #2B529h |
| RND interne → 2:réel/cplx/matr./réel_unité 1:exp.symb | #55DCDh | #55DCDh |
| RNRM (XLIB 2 118) | #1BF9Eh | #1BF9Eh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| RNRM interne →1: matrice | #3690Dh | #3690Dh |
| ROLL (XLIB 2 280) | #1FCB5h | #1FCB5h |
| ROLL et DROP interne →n..2:quelconque 1: entier sys. | #62F89h | #62F89h |
| ROLL et SWAP interne →n..2:quelconque 1: entier sys. | #62D45h | #62D45h |
| ROLL interne →n..2: quelconque 1: entier sys. | #03325h | #03325h |
| ROLLD (XLIB 2 281) | #1FCD0h | #1FCD0h |
| ROLLD interne →n..2: quelconque 1: entier sys. | #0339Eh | #0339Eh |
| ROOT (XLIB 2 251) | #1F16Eh | #1F16Eh |
| ROOT interne (3 arguments) | #32F77h | #32F77h |
| ROT (XLIB 2 274) | #1FC0Eh | #1FC0Eh |
| ROT et +1 (entier sys.) interne | #637B8h | #637B8h |
| ROT et DUP interne | #62775h | #62775h |
| ROT et DUP2 interne →3:quelconque 2:quelconque 1:quelconque | #62C7Dh | #62C7Dh |
| ROT et OVER interne | #62CA5h | #62CA5h |
| ROT interne | #03295h | #03295h |
| RR (XLIB 2 3) | #195DBh | #195DBh |
| RR interne | #53DA4h | #53DA4h |
| RRB (XLIB 2 4) | #195FBh | #195FBh |
| RRB interne | #53DE1h | #53DE1h |
| RSD (XLIB 2 123) | #1C03Eh | #1C03Eh |
| RSD interne →3:matrice 2:matrice 1:matrice | #365ACh | #365BBh |
| RULES (XLIB 2 335) | #20A7Dh | #20A7Dh |
| SAME (XLIB 2 228) | #1E761h | #1E761h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| SAME interne →2: quelconque 1: quelconque ← booléen | #03B97h | #03B97h |
| Saute l'objet suivant | #0714Dh | #0714Dh |
| Saute les deux objets suivants | #0715Ch | #0715Ch |
| Sauve le dernier RPL et vérifie $DEPTH \geq 1$ | #18AA5h | #18AA5h |
| Sauve le dernier RPL et vérifie $DEPTH \geq 2$ | #18A80h | #18A80h |
| Sauve le dernier RPL et vérifie $DEPTH \geq 3$ | #18A5Bh | #18A5Bh |
| Sauve le dernier RPL et vérifie $DEPTH \geq 4$ | #18B92h | #18B92h |
| Sauve le dernier RPL et vérifie $DEPTH \geq 5$ | #18B6Dh | #18B6Dh |
| Sauve le dernier RPL, vérifie $DEPTH \geq 1$ et contrôle les arguments | #18ECEh | #18ECEh |
| Sauve le dernier RPL, vérifie $DEPTH \geq 2$ et contrôle les arguments | #18EDFh | #18EDFh |
| Sauve le dernier RPL, vérifie $DEPTH \geq 3$ et contrôle les arguments | #18EF0h | #18EF0h |
| Sauve le dernier RPL, vérifie $DEPTH \geq 4$ et contrôle les arguments | #18F01h | #18F01h |
| Sauve le dernier RPL, vérifie $DEPTH \geq 5$ et contrôle les arguments | #18F12h | #18F12h |
| SBRK (XLIB 2 376) | #220C2h | #220C2h |
| SBRK interne | #2EE18h | #2EE18h |
| SCALE (XLIB 2 194) | #1E1E1h | #1E1E1h |
| SCALE interne →2: réel 1: réel | #4AE3Ch | #4AE3Ch |
| SCATRLOT (XLIB 2 318) | #2018Ch | #2018Ch |
| SCATTER (XLIB 2 225) | #1E701h | #1E701h |
| SCI (XLIB 2 139) | #1C41Eh | #1C41Eh |
| SCI interne →1: entier sys. | #166EFh | #166EFh |
| SCI interne →1: réel | #1C437h | #1C437h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| SCL Σ (XLIB 2 313) | #200C4h | #200C4h |
| SCONJ (XLIB 2 326) | #203CCh | #203CCh |
| SCONJ interne →1: nom global | #203EFh | #203EFh |
| SCONJ interne →1: nom Local | #20412h | #20412h |
| SDEV (XLIB 2 299) | #1FECFh | #1FECFh |
| SDEV interne | #2C5A3h | #2C5A3h |
| SEND (XLIB 2 363) | #21EF0h | #21EF0h |
| SEND interne →1: liste | #2E6EBh | #2E6EBh |
| SEND interne →1: nom global/nom local | #2E5ABh | #2E5ABh |
| SERVER (XLIB 2 368) | #21FD1h | #21FD1h |
| SERVER interne | #2D9F5h | #2D9F5h |
| SF (XLIB 2 131) | #1C274h | #1C274h |
| SF Interne →1: réel | #1C28Dh | #1C28Dh |
| SF interne (flags système) →1: entier sys. | #53731h | #53731h |
| SF interne (flags utilisateur) →1: entier sys. | #53725h | #53725h |
| SHOW (XLIB 2 338) | #20AD3h | #20AD3h |
| SHOW interne →2: exp. symb. 1: nom global/Nom Local | #58D75h | #58D75h |
| SHOW interne →2: expr. symb. 1: liste | #20B00h | #20B00h |
| SIGN (XLIB 2 78) | #1B32Ah | #1B32Ah |
| SIGN interne →1: expr. symb. | #54F68h | #54F68h |
| SIGN interne →1: complexe | #520CBh | #520CBh |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| SIGN interne → 1: réel_unité | #0FCE6h | #0FCE6h |
| SIGN interne → 1: réel | #2A8D7h | #2A8D7h |
| SIN (XLIB 2 81) | #1B4ACh | #1B4ACh |
| SIN interne → 1: expr. symb. | #54FB3h | #54FB3h |
| SIN interne → 1: réel long | #2AC06h | #2AC06h |
| SIN interne → 1: complexe | #52530h | #52530h |
| SIN interne → 1: réel_unité | #0F62Eh | #0F62Eh |
| SIN interne → 1: réel | #2ABEFh | #2ABEFh |
| SINH (XLIB 2 84) | #1B5B7h | #1B5B7h |
| SINH interne → 1: expr. symb. | #54FFEh | #54FFEh |
| SINH interne → 1: complexe | #5262Fh | #5262Fh |
| SINH interne → 1: réel | #2ADAEh | #2ADAEh |
| SINV (XLIB 2 324) | #202CEh | #202CEh |
| SINV interne → 1: nom global | #202F1h | #202F1h |
| SINV interne → 1: Nom Local | #20314h | #20314h |
| SIZE | #1C9B8h | #1C9B8h |
| SIZE (XLIB 2 160) | #1C9B8h | #1C9B8h |
| SIZE interne → 1: chaîne | #1CA26h | #1CA26h |
| SIZE interne → 1: chaîne ← entier sys. | #05636h | #05636h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| SIZE interne →1: entier bin. | #1CA85h | #1CA85h |
| SIZE interne →1: expr. symb. | #59F91h | #59F91h |
| SIZE interne →1: Graphique | #1CA62h | #1CA62h |
| SIZE interne →1: matrice | #1CA4Eh | #1CA4Eh |
| SIZE interne →1: matrice ← liste de deux entiers sys. | #035A9h | #035A9h |
| SIZE interne →1: réel_unité | #1CA3Ah | #1CA3Ah |
| SIZE interne →1: quelconque ← entier sys. | #05902h | #05902h |
| SIZE interne →1: chaîne ← entier sys. | #05616h | #05616h |
| SL (XLIB 2 5) | #1961Bh | #1961Bh |
| SL interne | #53D5Eh | #53D5Eh |
| SLB (XLIB 2 6) | #1963Bh | #1963Bh |
| SLB interne | #53D6Eh | #53D6Eh |
| SNEG (XLIB 2 325) | #2034Dh | #2034Dh |
| SNEG interne →1: nom global | #20370h | #20370h |
| SNEG interne →1: nom local | #20393h | #20393h |
| Sortie de boucle | #071ABh | #071ABh |
| SQ (XLIB 2 80) | #1B426h | #1B426h |
| SQ interne →1: expr. symb. | #54F9Ah | #54F9Ah |
| SQ interne →1: matrice | #36435h | #36444h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| SQ interne →1: complexe | #1B48Fh | #1B48Fh |
| SQ interne →1: réel_unité | #0F913h | #0F913h |
| SQ interne →1: réel | #1B47Bh | #1B47Bh |
| SQRT (XLIB 2 79) | #1B374h | #1B374h |
| SQRT interne →1: réel | #1B3F5h | #1B3F5h |
| SR (XLIB 2 7) | #1965Bh | #1965Bh |
| SR interne | #53D81h | #53D81h |
| SRB (XLIB 2 8) | #1967Bh | #1967Bh |
| SRB interne | #53D91h | #53D91h |
| SRECV interne →1: réel | #2EE97h | #2EE97h |
| START (XLIB 1792 9) | #23103h | #23103h |
| START interne →2: expr. symb. 1: réel | #23180h | #23180h |
| START interne →2: réel 1: réel | #23144h | #23144h |
| START interne →2: réel/exp.symb. 1: expr. symb. | #23167h | #23167h |
| STD (XLIB 2 141) | #1C486h | #1C486h |
| STD interne →1: entier sys. | #16707h | #16707h |
| STEP (XLIB 1792 12) | #23380h | #23380h |
| STEP interne →1: entier sys. | #073A5h | #073A5h |
| STEP interne →1: expr. symb. | #233A8h | #233A8h |
| STEP interne →1: réel | #233C1h | #233C1h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| STEQ (XLIB 2 250) | #1F14Eh | #1F14Eh |
| STEQ interne → 1: quelconque | #15717h | #15717h |
| STIME (XLIB 2 375) | #220A2h | #220A2h |
| STIME interne → 1: réel | #2EDF5h | #2EDF5h |
| STO (XLIB 2 341) | #20CCDh | #20CCDh |
| STO interne → 2: Bibliothèque/Sauvegarde 1: réel | #215BFh | #215BFh |
| STO interne → 2: Graphique 1: PICT | #4F37Ch | #4F37Ch |
| STO interne → 2: quelconque 1: nom local | #07D27h | #07D27h |
| STO interne → 2: quelconque 1: expression algébrique | #1F8CFh | #1F8CFh |
| STO interne → 2: quelconque 1: nom global | #18513h | #18513h |
| STO interne → 2: quelconque 1: Signé | #214F4h | #2150Fh |
| STO interne → 2: répertoire 1: nom global | #08D08h | #08D08h |
| STO* (XLIB 2 330) | #20753h | #20753h |
| STO* interne → 2: matrice 1: nom global | #2086Bh | #2086Bh |
| STO* interne → 2: nom global 1: réel/complexe | #2082Ah | #2082Ah |
| STO* interne → 2: nom global 1: matrice | #208ACh | #208ACh |
| STO* interne → 2: quelconque 1: nom global/Nom Local | #207C6h | #207C6h |
| STO* interne → 2: réel/complexe 1: nom global | #20802h | #20802h |
| STO+ (XLIB 2 327) | #2044Bh | #2044Bh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| STO+ interne →2:nom global/nom local 1:quelconque | #204C3h | #204C3h |
| STO+ interne →2:quelconque 1:nom global/nom local | #20482h | #20482h |
| STO- (XLIB 2 328) | #20538h | #20538h |
| STO- interne →2:matrice 1:nom global | #205BFh | #205BFh |
| STO- interne →2:nom global/nom local 1:quelconque | #205A1h | #205A1h |
| STO- interne →2:quelconque 1:nom global/nom local | #20583h | #20583h |
| STO/ (XLIB 2 329) | #2060Ch | #2060Ch |
| STO/ interne →2:nom global 1:réel/complexe | #206A7h | #206A7h |
| STO/ interne →2:nom global 1:matrice | #20729h | #20729h |
| STO/ interne →2:nom global/nom local 1:quelconque | #20689h | #20689h |
| STO/ interne →2:quelconque 1:nom global/nom local | #2066Bh | #2066Bh |
| STOALARM (XLIB 2 25) | #198FEh | #198FEh |
| <i>Stockage d'un entier système à l'adresse #70673h</i> →1: entier sys. | #04D0Eh | #04D0Eh |
| <i>Stockage des messages d'erreurs</i> →1:matrice 2:entier sys. | #0764Eh | #0764Eh |
| <i>Stockage des niveaux 1 à 5 dans 'ΣPAR'</i> | #2C6A2h/ | - |
| <i>Stocke 1er élément de la pile dans 'dvar' (Nom Local)</i> | #63A29h | #63A29h |
| <i>Stocke 1er élément de la pile dans 'LNAME' (Nom Local)</i> | #63A3Dh | #63A3Dh |
| STOF (flags système) | #1C731h | #1C731h |
| STOF (flags système) →1: entier bin. | #1C6E3h | #1C6E3h |
| STOF (flags utilisateur) →1: entier bin. | #1C6F7h | #1C6F7h |

| Fonction | Adresse | |
|--|-----------------|-----------------|
| | HP-48 S/SX | HP-48 G/GX |
| STOF (XLIB 2 151) | #1C67Fh/#1C67Fh | #1C67Fh/#1C67Fh |
| STOF interne →1: liste | #1C6A2h/#1C6A2h | #1C6A2h/#1C6A2h |
| STOF interne →2: entier bin. 1: entier bin. | #1C6CFh/#1C6CFh | #1C6CFh/#1C6CFh |
| STOKEYS (XLIB 2 380) | #22514h/#22514h | #22514h/#22514h |
| STOKEYS interne →1: liste | #41AA1h/#41AA1h | #41AA1h/#41AA1h |
| STOKEYS interne →1: nom global/nom local | #41BA5h/#41BA5h | #41BA5h/#41BA5h |
| STOΣ (XLIB 2 283) | #1FD0Bh/#1FD0Bh | #1FD0Bh/#1FD0Bh |
| STOΣ interne →1: quelconque | #2C1F3h/#2C1F3h | #2C1F3h/#2C1F3h |
| STR→ (XLIB 2 163) | #1CB26h/#1CB26h | #1CB26h/#1CB26h |
| STR→ interne →1: chaîne | #14137h/#14137h | #14137h/#14137h |
| STWS (XLIB 2 148) | #1C5C5h/#1C5C5h | #1C5C5h/#1C5C5h |
| STWS interne →1: entier sys. | #53CAAh/#53CAAh | #53CAAh/#53CAAh |
| STWS interne →1: réel | #53C96h/#53C96h | #53C96h/#53C96h |
| SUB (XLIB 2 156) | #1C85Ch/#1C85Ch | #1C85Ch/#1C85Ch |
| SUB interne →3:chaîne 2:réel 1:réel | #1C8BBh/#1C8BBh | #1C8BBh/#1C8BBh |
| SUB interne →3:Graphique 2:complexe 1:complexe | #4FBC4h/#4FBC4h | #4FBC4h/#4FBC4h |
| SUB interne →3:Graphique 2:liste 1:liste | #4FB74h/#4FB74h | #4FB74h/#4FB74h |
| SUB interne →3:liste 2:réel 1:réel | #1C8CFh/#1C8CFh | #1C8CFh/#1C8CFh |
| SUB interne →3:PICT 2/1: liste ou complexe | #4FBF6h/#4FBF6h | #4FBF6h/#4FBF6h |
| SWAP (XLIB 2 271) | #1FBBDh/#1FBBDh | #1FBBDh/#1FBBDh |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| SWAP des niveaux 2 et 3 interne | #60EE7h | #60EE7h |
| SWAP DROP et DUP interne | #62830h | #62830h |
| SWAP DROP et SWAP interne | #6284Bh | #6284Bh |
| SWAP DROP et VRAI interne | #21660h | #21660h |
| SWAP DROP SWAP DROP interne | #6112Ah | #6112Ah |
| SWAP DROP SWAP DROP SWAP DROP interne | #6113Ch | #6113Ch |
| SWAP et 4 PICK interne | #63C7Ch | #63C7Ch |
| SWAP et 4 ROLL interne | #63C2Ch | #63C2Ch |
| SWAP et DROP interne | #60F9Bh | #60F9Bh |
| SWAP et DUP interne | #62747h | #62747h |
| SWAP et DUP2 interne | #6386Ch | #6386Ch |
| SWAP et FAUX interne | #2164Ch | #2164Ch |
| SWAP et LIST → interne | #631F5h | #631F5h |
| SWAP et NEXT (boucle interne) interne | #6344Dh | #6344Dh |
| SWAP et OVER interne | #61380h | #61380h |
| SWAP et R → C interne | #632A9h | #632A9h |
| →2: réel 1: réel | | |
| SWAP interne | #03223h | #03223h |
| SWAP interne | #62794h | #62794h |
| →2/1: entier sys. | | |
| SYSEVAL (XLIB 2 49) | #1A52Eh | #1A52Eh |
| SYSEVAL interne | #1A547h | #1A547h |
| →1: entier bin. | | |
| Système Binaire → réel long interne | #63B96h | #63B96h |
| →1: entier sys. | | |
| TAN (XLIB 2 83) | #1B55Eh | #1B55Eh |
| TAN interne | #54FE5h | #54FE5h |
| →1: expr. symb. | | |
| TAN interne | #2AC17h | #2AC17h |
| →1: réel long | | |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| TAN interne → 1: complexe | #525B7h | #525B7h |
| TAN interne → 1: réel_unité | #0F674h | #0F674h |
| TAN interne → 1: réel | #2AC91h | #2AC91h |
| TANH (XLIB 2 86) | #1B655h | #1B655h |
| TANH interne → 1: expr. symb. | #55030h | #55030h |
| TANH interne → 1: complexe | #5265Ch | #5265Ch |
| TANH interne → 1: réel | #2ADEDh | #2ADEDh |
| TAYLR (XLIB 2 339) | #20B20h | #20B20h |
| TAYLR interne → 3: expr. symb. 2: nom global 1: réel | #595DDh | #595DDh |
| TEXT | #1E606h | #1E606h |
| TEXT (XLIB 2 217) | #1E606h | #1E606h |
| TEXT interne | #1314Dh | #1314Dh |
| THEN (XLIB 1792 1) | #22EFAh | #22EFAh |
| THEN (XLIB 1792 24) | #2371Fh | #2371Fh |
| THEN (XLIB 1792 26) | #237A8h | #237A8h |
| THEN interne → 1: expr. symb. | #22F4Fh | #22F4Fh |
| THEN interne → 1: réel | #22F22h | #22F22h |
| TICKS (XLIB 2 18) | #1982Dh | #1982Dh |
| TICKS interne | #0EB81h | #0EB81h |
| TIME (XLIB 2 16) | #197F7h | #197F7h |
| TIME interne | #0CBFAh | #0CBFAh |
| TLINE | #1E3C2h | #1E3C2h |
| TLINE (XLIB 2 207) | #1E3C2h | #1E3C2h |

| Fonction | Adresse | |
|---|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| TLINE interne →2: complexe 1: complexe | #4F598h | #4F598h |
| TLINE interne →2: liste 1: liste | #4F539h | #4F539h |
| TMENU (XLIB 2 347) | #2115Dh | #2115Dh |
| TMENU interne →1: réel | #41679h | #41679h |
| TOT (XLIB 2 300) | #1FEEAh | #1FEEAh |
| TOT interne | #2C5BCh | #2C5BCh |
| TRANSIO (XLIB 2 372) | #2204Ch | #2204Ch |
| TRANSIO interne →1: réel | #2ED10h | #2ED10h |
| TRN (XLIB 2 175) | #1D392h | #1D392h |
| TRN interne →1: matrice | #3811Fh | #3811Fh |
| TRN interne →1: nom global | #1D3BFh | #1D3BFh |
| TRN interne →1: nom local | #1D3E2h | #1D3E2h |
| TRNC (XLIB 2 109) | #1BDD1h | #1BDD1h |
| TRNC interne →2: expr. symb. 1: réel | #55DFh | #55DFh |
| TRNC interne →2: matrice 1: réel | #35EA9h | #35EA9h |
| TRNC interne →2: complexe 1: réel | #35F17h | #35F17h |
| TRNC interne →2: réel_unité 1: réel | #0FD8Bh | #0FD8Bh |
| TRNC interne →2: réel 1: réel | #2B53Dh | #2B53Dh |
| TRUTH (XLIB 2 224) | #1E6E1h | #1E6E1h |
| TSTR (XLIB 2 29) | #19992h | #19992h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| TSTR interne | #0D304h | #0D304h |
| TVARS (XLIB 2 37) | #1A1AFh | #1A1AFh |
| TVARS interne →1: liste | #18706h | #18706h |
| TYPE (XLIB 2 166) | #1CB86h | #1CB86h |
| TYPE interne →1: matrice | #1CDB1h | #1CDB1h |
| TYPE interne →1: Programme RPL | #1CDD4h | #1CDD4h |
| TYPE interne →1: quelconque | #1CB90h | #1CB90h |
| UBASE (XLIB 2 14) | #19771h | #19771h |
| UBASE interne →1: expr. symb. | #55256h | #55256h |
| UBASE interne →1: réel_unité | #0F945h | #0F945h |
| UFACT (XLIB 2 15) | #197A5h | #197A5h |
| UFACT interne | #197C8h | #197C8h |
| UNIT (XLIB 2 13) | #1974Fh | #1974Fh |
| UNIT interne | #0F33Ah | #0F33Ah |
| UNTIL (XLIB 1792 8) | #230EDh | #230EDh |
| UNTIL interne | #633C6h | #633C6h |
| UPDIR (XLIB 2 35) | #1A15Bh | #1A15Bh |
| UPDIR interne | #08309h | #08309h |
| UPDIR interne | #1A16Fh | #1A16Fh |
| UTPC (XLIB 2 308) | #2001Ah | #2001Ah |
| UTPC interne →2: réel 1: réel | #2C149h | #2C12Eh |
| UTPF (XLIB 2 310) | #2005Ah | #2005Ah |
| UTPF interne →3:réel 2:réel 1:réel | #2C174h | #2C15Eh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| UTPN (XLIB 2 309) | #2003Ah | #2003Ah |
| UTPN interne → 3:réel 2:réel 1:réel | #2C09Fh | #2C09Fh |
| UTPT (XLIB 2 311) | #2007Ah | #2007Ah |
| UTPT interne → 2:réel 1:réel | #2C19Ah | #2C189h |
| UVAL (XLIB 2 12) | #1971Bh | #1971Bh |
| UVAL interne → 1: réel_unité | #05089h | #05089h |
| V→ (XLIB 2 180) | #1DD06h | #1DD06h |
| V→ interne → 1: matrice | #1DD3Dh | #1DD3Dh |
| V→ interne → 1: complexe | #1DD29h | #1DD29h |
| VAR (XLIB 2 301) | #1FF05h | #1FF05h |
| VAR interne | #2C5D5h | #2C5D5h |
| VARs (XLIB 2 36) | #1A194h | #1A194h |
| VARs interne | #18779h | #18779h |
| VRAI | #03A81h | #03A81h |
| VRAI et FAUX interne | #634F7h | #634F7h |
| VTYP (XLIB 2 167) | #1CE28h | #1CE28h |
| VTYP interne → 1: nom global/nom local | #1CE55h | #1CE55h |
| VTYP interne → 1: Signé | #1CE82h | #1CE82h |
| WAIT (XLIB 2 55) | #1A71Fh | #1A71Fh |
| WAIT Interne → 1: réel > 0 | #1A7B5h | #1A7B5h |
| WAIT Interne → 1: réel | #1A738h | #1A738h |
| WHILE (XLIB 1792 5) | #23033h | #23033h |
| WSLOG (XLIB 2 19) | #19848h | #19848h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| WSLOG <i>Interne</i> | #0D2A3h | #0D2A3h |
| XCOL (XLIB 2 306) | #1FFDAh | #1FFDAh |
| XCOL <i>interne</i> →1: réel | #2C6C5h | #2C6B6h |
| XLIB 2 260 | #1F640h | #1F640h |
| XLIB 2 261 | #1F996h | #1F996h |
| XLIB 2 262 | #1F9AEh | #1F9AEh |
| XLIB → | #08CCCh | #08CCCh |
| XMIT <i>interne</i> →1: chaîne | #2EE6Fh | #2EE6Fh |
| XOR (XLIB 2 232) | #1E8F6h | #1E8F6h |
| XOR <i>interne</i> →1: booléen 2: booléen ← booléen | #03ADAh | #03ADAh |
| XOR <i>interne</i> →2: chaîne 1: chaîne | #1889Bh | #1889Bh |
| XOR <i>interne</i> →2: chaîne 1: chaîne | #18904h | #18904h |
| XOR <i>interne</i> →2: entier bin. 1: entier bin. | #53D26h | #53D26h |
| XOR <i>interne</i> →2: expr. symb. 1: expr. symb. | #55A62h | #55A62h |
| XOR <i>interne</i> →2: expr. symb. 1: réel | #55A30h | #55A30h |
| XOR <i>interne</i> →2: réel 1: expr. symb. | #55A49h | #55A49h |
| XOR <i>interne</i> →2: réel 1: réel | #1E946h | #1E946h |
| XPON (XLIB 2 105) | #1BC45h | #1BC45h |
| XPON <i>interne</i> →1: expr. symb. | #551F2h | #551F2h |
| XPON <i>interne</i> →1: réel | #2AE39h | #2AE39h |

| Fonction | Adresse | |
|--------------------------|-----------------|-------------------------|
| | HP-48 S/SX | HP-48 G/GX |
| XRNG (XLIB 2 218) | #1E621h/#1E621h | |
| XRNG interne | #47A1Ah/#47A1Ah | |
| →2: réel | | 1: réel |
| XROOT (XLIB 2 74) | #1B185h/#1B185h | |
| XROOT (XLIB 2 75) | #1B1CAh/#1B1CAh | |
| XROOT interne | #0F8FAh/#0F8FAh | |
| →1: réel_unité | | |
| XROOT interne | #2AA81h/#2AA81h | |
| →2: réel | | 1: réel |
| YCOL interne | #2C6DEh/#2C6CFh | |
| →1: réel | | |
| YRNG | #1E641h/#1E641h | |
| YRNG (XLIB 2 219) | #1E641h/#1E641h | |
| YRNG interne | #47A42h/#47A42h | |
| →2: réel | | 1: réel |
| ^ (XLIB 2 73) | #1B02Dh/#1B02Dh | |
| ^ interne | #55F12h/#55F12h | |
| →2: expr. symb. | | 1: expr. symb. |
| ^ interne | #55EE0h/#55EE0h | |
| →2: expr. symb. | | 1: réel/cplx/réel_unité |
| ^ interne | #2AA5Fh/#2AA5Fh | |
| →2: réel long | | 1: réel long |
| ^ interne | #52374h/#52374h | |
| →2: complexe | | 1: complexe |
| ^ interne | #52360h/#52360h | |
| →2: complexe | | 1: réel |
| ^ interne | #52342h/#52342h | |
| →2: réel | | 1: complexe |
| ^ interne | #1B124h/#1B124h | |
| →2: réel | | 1: réel |
| ^ interne | #2AA70h/#2AA70h | |
| →2: réel | | 1: réel |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| ^ <i>interne</i> → 2: réel/cplx/réel_unité 1: exp. symb. | #55EF9h | #55EF9h |
| ^ <i>interne</i> → 2: réel/réel_unité 1: réel/réel_unité | #0F878h | #0F873h |
| _ (XLIB 2 267) | #1FAEBh | #1FAEBh |
| _ <i>interne</i> → 1: expr. symb. | #28A38h | #28A38h |
| _ <i>interne</i> → 1: réel/réel_unité | #1FB31h | #1FB31h |
| (<i>usage algébrique</i>) (XLIB 2 256) | #1F3F3h | #1F3F3h |
| (<i>usage dans la pile</i>) (XLIB 2 255) | #1F354h | #1F354h |
| <i>interne (usage dans la pile)</i> → 2: expr. symb. 1: liste | #1F38Bh | #1F38Bh |
| ≠ (XLIB 2 234) | #1EA9Dh | #1EA9Dh |
| ≠ <i>interne</i> → 2: réel/cplx/réel_unité 1: exp. symb. | #55ADFh | #55ADFh |
| ≠ <i>interne</i> → 2: entier bin. 1: entier bin. | #544ECh | #544ECh |
| ≠ <i>interne</i> → 2: exp. symb. 1: réel/cplx/réel_unité | #55AC6h | #55AC6h |
| ≠ <i>interne</i> → 2: expr. symb. 1: expr. symb. | #55AF8h | #55AF8h |
| ≠ <i>interne</i> → 2: complexe 1: réel | #1EB97h | #1EB97h |
| ≠ <i>interne</i> → 2: quelconque 1: quelconque | #1EB51h | #1EB51h |
| ≠ <i>interne</i> → 2: quelconque 1: quelconque ← booléen | #635C4h | #635C4h |
| ≠ <i>interne</i> → 2: réel 1: complexe | #1EB8Dh | #1EB8Dh |
| ≠ <i>interne</i> → 2: réel 1: réel ← booléen | #2A8CCh | #2A8CCh |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| \neq interne →2:réel/réel_unité 1:réel/réel_unité | #0F598h | #0F598h |
| \neq interne →2:signé/quelconque 1:signé/quelconque | #1EB65h | #1EB65h |
| #MATCH (XLIB 2 265) | #1FA59h | #1FA59h |
| #MATCH interne →2:réel/complexe/exp. symb. 1:liste | #1FABAh | #1FABAh |
| \leq (XLIB 2 237) | #1ECFCh | #1ECFCh |
| \leq interne →2: entier bin. 1: entier bin. | #5453Fh | #5453Fh |
| \leq interne →2: expr. symb. 1: expr. symb. | #55BD9h | #55BD9h |
| \leq interne →2:expr. symb. 1:réel/réel_unité | #55BA7h | #55BA7h |
| \leq interne →2: réel 1: réel | #1ED7Eh | #1ED7Eh |
| \leq interne →2: réel 1: réel ← booléen | #2A8ABh | #2A8ABh |
| \leq interne →2: réel 1: réel ← booléen | #2A8B6h | #2A8B6h |
| \leq interne →2:réel/réel_unité 1: expr. symb. | #55BC0h | #55BC0h |
| \leq interne →2:réel/réel_unité 1:réel/réel_unité | #0F5D4h | #0F5D4h |
| \geq (XLIB 2 238) | #1ED9Bh | #1ED9Bh |
| \geq interne →2: chaîne 1: chaîne | #142BAh | #142BAh |
| \geq interne →2: entier bin. 1: entier bin. | #5452Ch | #5452Ch |
| \geq interne →2: expr. symb. 1: expr. symb. | #55C24h | #55C24h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| \geq interne →2: expr. symb. 1: réel/réel_unité | #55BF2h | #55BF2h |
| \geq interne →2: réel 1: réel | #1EE1Dh | #1EE1Dh |
| \geq interne →2: réel 1: réel ← booléen | #2A895h | #2A895h |
| \geq interne →2: réel 1: réel ← booléen | #2A8A0h | #2A8A0h |
| \geq interne →2: réel/réel_unité 1: expr. symb. | #55C0Bh | #55C0Bh |
| \geq interne →2:réel/réel_unité 1:réel/réel_unité | #0F5E8h | #0F5E8h |
| Σ (XLIB 2 254) | #1F2C9h | #1F2C9h |
| Σ interne →4:exp.symb. 3:exp.symb. →2:exp.symb. 1:quelconque | #56949h | #56949h |
| Σ interne →4: expr. symb. 3: expr. symb. →2: réel 1: quelconque | #56A06h | #56A06h |
| Σ interne →4: expr. symb. 3: réel →2: réel 1: quelconque | #56AC9h | #56AC9h |
| Σ interne →4: expr. symb. 3: réel →2: expr. symb. 1: quelconque | #56A4Ch | #56A4Ch |
| $\Sigma+$ (XLIB 2 286) | #1FD61h | #1FD61h |
| $\Sigma+$ interne →1: matrice | #2C32Eh | #2C32Eh |
| $\Sigma+$ interne →1: réel | #2C2D9h | #2C2D9h |
| $\Sigma-$ (XLIB 2 287) | #1FD8Bh | #1FD8Bh |
| $\Sigma-$ interne | #2C423h | #2C423h |

| Fonction | Adresse | |
|--|------------|------------|
| | HP-48 S/SX | HP-48 G/GX |
| Σ LINE (XLIB 2 314) | #200F3h | #200F3h |
| Σ LINE interne | #4A16Ch | #48B9Dh |
| Σ X (XLIB 2 291) | #1FDF7h | #1FDF7h |
| Σ X interne | #2C94Fh | #2C940h |
| Σ X*Y (XLIB 2 295) | #1FE63h | #1FE63h |
| Σ X*Y interne | #2C9BDh | #2C9C2h |
| Σ X^2 (XLIB 2 293) | #1FE2Dh | #1FE2Dh |
| Σ X^2 interne | #2C977h | #2C972h |
| Σ Y (XLIB 2 292) | #1FE12h | #1FE12h |
| Σ Y interne | #2C963h | #2C959h |
| Σ Y^2 (XLIB 2 294) | #1FE48h | #1FE48h |
| Σ Y^2 interne | #2C99Ah | #2C99Ah |
| π (XLIB 2 63) | #1AABDh | #1AABDh |
| π interne (3.14159265359) | #54D35h | #54D35h |
| \int (algébrique) (XLIB 2 253) | #1F223h | #1F223h |
| \int (sur la pile) (XLIB 2 252) | #1F1D4h | #1F1D4h |
| \int interne (algébrique) | #1F27Ah | #1F27Ah |
| \int interne (sur la pile) | #1F201h | #1F201h |
| $\sqrt{\quad}$ interne → 1: expr. symb. | #54F81h | #54F81h |
| $\sqrt{\quad}$ interne → 1: réel long | #2AAEAh | #2AAEAh |
| $\sqrt{\quad}$ interne → 1: complexe | #52107h | #52107h |
| $\sqrt{\quad}$ interne → 1: réel_unité | #0F92Ch | #0F92Ch |
| $\sqrt{\quad}$ interne → 1: réel | #2AB09h | #2AB09h |

Glossaire

@ Symbole généralement utilisé pour représenter une adresse en mémoire.

A (champ) Ce champ correspond aux 5 quartets de poids faible d'un registre de travail.

A (registre) Registre de travail offrant de nombreuses possibilités de traitement.

Accès aléatoire Accès immédiat à un point donné de la mémoire représenté par son adresse.

Accès séquentiel Accès à un point donné d'une mémoire après avoir parcouru toute cette mémoire depuis son début jusqu'au point recherché.

Access pointer Voir *pointeur étendu*.

Adresse Numéro d'un quartet en mémoire. Les adresses sont codées en mémoire à l'aide de 5 quartets de #00000h à #FFFFFh.

Algorithme Décomposition d'un travail à effectuer en travaux élémentaires compte tenu de l'exécutant.

Alpha-lock Verrouillage du clavier en mode alphabétique.

And Voir *Et logique*.

Arborescence (menus) Hiérarchie des menus permettant l'appel d'un sous-menu à partir d'un article d'un autre menu.

Arborescence (répertoires) Emboîtement des répertoires (dossiers) les uns dans les autres.

Arité Nombre d'arguments ou de paramètres qu'attend un opérateur, une commande ou une fonction.

ASCII *American Standard Code for Information Interchange* Table faisant correspondre la valeur d'un octet (2 quartets) avec un caractère usuel (lettre, chiffre, symbole, etc.).

Assembleur (langage) Langage de programmation permettant de contrôler la machine au plus bas niveau en intervenant directement au niveau du microprocesseur. L'assembleur est une représentation « lisible » du langage machine qui lui n'est constitué que de 0 et de 1. L'assembleur permet de représenter ces 0 et ces 1 par des chiffres hexadécimaux (représentations d'un groupe de 4 bits, ou quartets) ou par des mnémoniques.

Assembleur (programme) Programme chargé de remplacer les mnémoniques utilisés en langage assembleur par les codes hexadécimaux correspondants, ou mieux, par le programme exécutable. Un assembleur est un outil de programmation en langage assembleur (on dit aussi langage d'assemblage).

B (champ) Deux quartets de poids faible d'un registre de travail.

B (registre) L'un des quatre registres de travail.

Balayage Rafraîchissement ligne par ligne du contenu de l'écran.

Bank switching Sélection de l'une des portions de 128 ko d'une carte en port 2 sur HP-48 GX.

Base 10 Base de numération décimale usuelle.

Base 16 Base de numération hexadécimale utilisée en informatique car un chiffre hexadécimal peut représenter toutes les combinaisons possibles de 4 bits.

Base 2 Base de numération binaire. Les seuls chiffres autorisés sont 0 et 1 représentant les deux seuls états logiques reconnus en électronique numérique.

Base de numération Détermine les chiffres utilisables (0 et 1 en base 2, de 0 à 9 en base 10, de 0 à 9 et de A à F en base 16).

BCD Voir *DCB*.

Bibliothèque Objet particulier réunissant d'autres objets nommés. Une bibliothèque a une structure à rapprocher de celle d'un répertoire mais doit aussi être considérée comme un ensemble de commandes.

Binaire Voir *Base 2*.

Binary digit Voir *Bit*.

Bit Nombre binaire. Un bit vaut 0 ou 1. Un groupe de 4 bits est un quartet, un groupe de 8 bits est un octet.

Bit armé Bit ayant la valeur 1.

Bit désarmé Bit ayant la valeur 0.

Branchement Reprise de l'exécution d'un programme à partir d'une instruction appelée « point de branchement » et ne se trouvant pas immédiatement après la dernière instruction exécutée.

Buffer Voir *Mémoire tampon*.

Byte Voir *Octet*.

BYTES Commande accessible en RPL et permettant de connaître le nombre de quartets occupés en mémoire par un objet ainsi que le checksum (somme de contrôle) de celui-ci.

C Registre de travail offrant le plus de possibilités.

Caractère Lettre, chiffre ou symbole codé sur 1 octet (c'est-à-dire 2 quartets ou 8 bits).

Chaîne de caractères Ensemble de caractères juxtaposés. L'objet chaîne de caractères accessible à l'utilisateur est délimité par " et ".

Champ d'un registre Partie nommée d'un registre de travail. Une opération sur un registre peut être limitée à l'un de ses champs.

Checksum Somme de contrôle ou CRC permettant de vérifier l'intégrité d'un objet. La commande BYTES en RPL renvoie - entre autres - le checksum d'un objet de l'utilisateur.

Chemin courant Liste des répertoires ouverts pour accéder aux variables actuellement disponibles (affichées après une pression sur [VAR]). Le chemin courant est affiché en haut à gauche de l'écran. Le premier répertoire du chemin courant est le répertoire racine (HOME) alors que le dernier répertoire du chemin courant est le répertoire courant (celui dont les variables sont actuellement disponibles).

Chemin d'accès Liste des répertoires à ouvrir pour accéder à une variable.

Chip « Puce électronique » ou module électronique, composant électronique prenant en charge une tâche déterminée.

Code de redondance cyclique CRC, voir *Checksum*.

Coma Mode de fonctionnement « basse consommation » de la HP-48.

Complément Inversion de bits (voir les chapitres d'initiation à la logique).

CRC Code de redondance cyclique, voir *Checksum*.

D L'un des 4 registres de travail.

D0 Registre contenant une adresse.

D1 Registre contenant une adresse.

Daisy chain Chaîne logique entre les différents modules de mémoire. Détermine la configuration de la mémoire.

DCB Décimal codé binaire. Représentation sur un quartet des chiffres de 0 à 9. Comme un quartet peut aussi représenter les chiffres hexadécimaux A à F, le DCB consiste à interdire ces représentations pour se limiter aux nombres de la base 10.

Décimal codé binaire Voir *DCB*.

Décrémentation Diminution, généralement par pas de 1.

Désassembleur Programme chargé de transformer un code exécutable en code hexadécimal pouvant être édité.

Drapeau Bit (ou, plus rarement, un quartet) chargé d'interdire ou d'autoriser une fonction. Les drapeaux peuvent être comparés à des interrupteurs paramétrant le fonctionnement de la calculatrice en « allumant » ou en « éteignant » certaines fonctions.

Drapeau armé Le bit représentant le drapeau vaut 1. Voir aussi *Drapeau*.

Drapeau désarmé Le bit représentant le drapeau vaut 0. Voir aussi *Drapeau*.

Editer Créer ou modifier des données. A l'origine le verbe éditer est synonyme de corriger.

Editeur Programme utilisé pour créer ou modifier des données.

Entrées-sorties Ensemble de moyens dont dispose la machine pour échanger des informations.

Environnement Ensemble de variables visibles à un instant donné. Dans le cadre d'une utilisation classique de la HP-48, les variables visibles sont celles du répertoire courant et celles de tous les répertoires du chemin courant. L'utilisation des applications intégrées telles que Equation Writer ou Matrix vous place dans un environnement particulier.

Epilogue Ensemble de quartets (5 généralement) chargé de signaler la fin du code décrivant un objet.

Equation Writer Application intégrée à la HP-48 et chargée de faciliter l'introduction

d'expressions mathématiques. Cette application se charge de la conversion depuis la notation algébrique directe vers la notation polonaise inversée (RPN). L'application Equation Writer constitue un environnement personnalisé. Voir *RPN*.

Et logique Opération logique (produit logique) dont la table de vérité est :

0 and 0 = 0

0 and 1 = 0

1 and 0 = 0

1 and 1 = 1

Evaluation Obtention de la valeur d'une variable. Dans le cas d'une variable non exécutable (nombre, caractère, chaîne de caractères, etc.), l'évaluation de la variable correspond à la valeur constante de la variable (selon le cas, le nombre, le caractère, etc.). Si la variable est exécutable (programme RPL, code exécutable, etc.), son évaluation correspond à l'exécution du programme.

Flag Voir *Drapeau*.

Flag armé Voir *Drapeau armé*.

Flag désarmé Voir *Drapeau désarmé*.

Garbage collection Opération effectuée régulièrement par un système informatique afin de libérer les espaces occupés par des variables devenues inutiles. La zone où sont stockés les objets temporaires est débarrassée des objets temporaires inutilisés lors d'une Garbage collection.

Glaneur de cellules Voir *Garbage Collection*.

GROB Objet graphique en mode point (objet graphique bitmap). Il s'agit du codage pixel par pixel d'une image destinée à l'affichage. Voir *Pixel*.

GROB courant Objet graphique pour l'ins-tant affiché. En fonctionnement normal, la

HP-48 peut passer de l'affichage de la pile à celui de l'environnement graphique GRAPH ou PICTURE (selon la version). C'est donc soit le GROB pile, soit le GROB environnement graphique qui est affiché et est donc le GROB courant.

GROB environnement graphique Objet graphique décrivant pixel par pixel l'affichage à l'écran de l'environnement GRAPH ou PICTURE (selon la version). Voir *Pixel*.

GROB menus Objet graphique décrivant pixel par pixel le contenu de la barre des menus en bas de l'écran. Voir *Pixel*.

GROB pile Objet graphique décrivant pixel par pixel la représentation à l'écran du contenu de la pile. Voir *Pixel*.

Hexa Voir *Hexadécimal*.

Hexadécimal Utilisation de la base 16 pour les calculs. Voir aussi *Base 16*.

HOME Nom donné au répertoire racine de la HP-48. Voir *Répertoire racine*.

HP-48 G Version apparue en 1993, dotée de 32 ko de mémoire vive interne et de 512 ko de ROM mais dépourvue de lecteur de carte.

HP-48 GX Version apparue en 1993, dotée de 128 ko de mémoire vive interne, de 512 ko de ROM et de deux lecteurs de cartes dont l'un supporte jusqu'à 4 Mo.

HP-48 S Version simplifiée de la HP-48 SX apparue en 1991. Dépourvue de lecteur de cartes.

HP-48 SX Première calculatrice de la série HP-48 apparue en 1989. Dotée de 32 ko de mémoire vive interne, de 256 ko de ROM et de deux lecteurs de cartes.

I/O Input/Output Voir *Entrées/sorties*.

IN et OUT (registres) Registres utilisés pour connaître les touches pressées.

Incrémentation Augmentation, généralement par pas de 1.

Indicateur Voir *Témoin*.

Kermit Protocole de transmission utilisé par la HP-48 pour échanger des données via le câble série (RS 232) ou les diodes infrarouge. Voir aussi *RS 232*.

Kilo octet 1024 octets (2^{10}). Voir *Octet*.

Langage machine Langage constitué d'ordres directement compréhensibles par le microprocesseur codés par des 0 et des 1 (c'est un langage binaire). Le langage assembleur n'est qu'une représentation « humainement lisible » du langage machine.

LCD Procédé d'affichage par cristaux liquides utilisé pour les écrans de la plupart des calculatrices.

Liaison (communication) Lien physique (câble) ou optique (faisceau infrarouge) entre la HP-48 et un autre dispositif (un micro-ordinateur, une autre HP-48, l'imprimante HP, etc.).

Liaison (variable) Association d'un objet à un nom (placé entre ' et ') afin de créer une variable qui n'est qu'un couple constitué d'un objet et de son nom (ce nom est aussi un objet). Un environnement est un ensemble de liaisons, et donc, un ensemble de variables. Voir *Environnement*, *Variable*.

Library Voir *Bibliothèque*.

Masquage des mémoires Le SATURN a une capacité d'adressage limitée à #FFFFh adresses différentes. Cela impose l'utilisation d'une même adresse au sein de plusieurs modules de mémoire différents. Pour

éviter les confusions, il existe une hiérarchie entre les modules de mémoire. Si deux modules utilisent la même adresse, celle-ci, lorsqu'elle employée, désigne un quartet du module de mémoire le plus prioritaire. Un quartet de même adresse dans un autre module moins prioritaire est donc masqué. Un module de mémoire masqué est accessible après une reconfiguration de la mémoire. Voir *Mémoire*.

Matrix Writer Application HP d'édition de matrices (et de vecteurs).

Méga octet 1048576 octets (2^{20}). Voir *Octet*.

Mémoire Dispositif électronique de sauvegarde des données informatiques. Il existe six modules de mémoire dans les HP-48 (G/GX) : la mémoire vive des entrées/sorties, la mémoire vive interne, le gestionnaire de partitions, la carte en port 1, la carte en port 2 et la mémoire morte.

Mémoire masquée Voir *Masquage des mémoires*.

Mémoire morte Mémoire dont le contenu a été figé lors de la construction de la machine, il est donc impossible d'y écrire des données. Elle contient toutes les données et toutes les fonctions disponibles en standard ainsi que certains programmes (notamment, pour le démarrage de la calculatrice). Sa capacité est de 256 ko sur les HP-48 S/SX et de 512 ko sur HP-48 G/GX.

Mémoire tampon Mémoire de faible capacité (en général, il s'agit d'une partie de la RAM système ou de la RAM I/O) chargée d'absorber un flot de données entrantes ou de conserver peu de temps des données à émettre vers un périphérique.

Mémoire vive Mémoire où il est à la fois possible d'écrire et de lire des données. Les premiers quartets de la mémoire vive cor-

respondent à la mémoire vive réservée au système. Les quartets suivants contiennent - entre autres - les variables créées par l'utilisateur (programmes, données diverses, etc.). La mémoire vive interne (celle de la HP-48 proprement dite) peut être complétée par des cartes de mémoire vive.

Mémoire vive des entrées/sorties Module de mémoire ayant le plus haut niveau de priorité. Contient des données relatives aux entrées/sorties (avec l'écran entre autres).

Mémoire vive interne Mémoire vive de la HP-48 sans ajout de carte de mémoire vive.

Mémoire vive réservée au système Les premiers quartets de la mémoire vive sont réservés à la HP-48 qui y stocke de nombreux paramètres.

Menu courant Menu affiché à l'instant considéré.

Microprocesseur Organe d'un système micro-informatique chargé des traitements.

Mnémotechniques Représentations mnémotechniques des codes hexadécimaux utilisés en assembleur. Ces codes hexadécimaux sont eux-mêmes la représentation directe des bits (des 0 et des 1) constituant le langage machine.

Mode courant Mode d'utilisation de la calculatrice à l'instant considéré.

Nibble Voir *Quartet*.

Nombre binaire voir *Bit*.

Non logique Opération logique consistant à remplacer tous les bits à 1 d'un mot binaire par des bits à 0 et inversement.

Not Voir *Non logique*.

Objet Donnée « encapsulée » dans la mémoire. Un objet est d'un type défini par son prologue et peut être associé à un objet de type nom pour créer une variable.

Octet Groupe de 8 bits, c'est-à-dire, de 2 quartets. Voir *Bit*.

OR Voir *OU logique*.

Ou logique Opération logique (somme logique) dont la table de vérité est :

0 or 0 = 0

0 or 1 = 1

1 or 0 = 1

1 or 1 = 1

Ou logique exclusif Opération logique dont la table de vérité est :

0 xor 0 = 0

0 xor 1 = 1

1 xor 0 = 1

1 xor 1 = 0

P (champ) Le registre P contient un nombre compris entre #0h et #Fh. Ce nombre est le numéro du quartet constituant le champ P des registres de travail.

P (registre) Le nombre compris entre #0h et #Fh que contient ce registre détermine le numéro du quartet constituant le champ P des registres de travail.

Partition L'une des tranches de 128 ko d'une carte pour le port 2 des HP-48 GX.

PC Registre du SATURN contenant l'adresse de la prochaine instruction à exécuter.

Pile Structure de données de type LIFO (*Last In First Out*), « le dernier introduit est le premier sorti ».

Pixel *Picture Element*. Point de l'écran.

Poids Valeur d'une portion d'un mot binaire par rapport au reste du mot. Les bits de

poids fort (MSB, bits les plus significatifs) se trouvent à gauche du mot binaire. Les bits de poids faible (LSB, bits les moins significatifs) sont à droite du mot binaire. De même, on peut par exemple diviser un octet en deux quartets, l'un de poids fort, l'autre de poids faible.

Pointeur Registre ou champ en mémoire contenant une adresse en mémoire.

Pointeur étendu Objet créé en associant l'adresse d'un objet placé dans une mémoire masquée et l'adresse d'une routine rendant accessible ladite mémoire masquée.

Priorité des mémoires Elle détermine le module de mémoire utilisé lorsqu'une même adresse existe dans plusieurs modules. En configuration normale et sur HP-48 G/GX, les modules sont, par ordre de priorité décroissante : la RAM I/O, la RAM interne, le gestionnaire de partitions, la carte du port 1, celle du port 2 et la ROM. Voir *Mémoire*.

Processeur Voir *Microprocesseur*.

Program Counter Voir *PC*.

Prologue Champ d'un objet déterminant le type d'un objet. Plus généralement, entête représentative d'une zone de mémoire.

Quartet Groupe de 4 bits. Un quartet est représenté par un nombre hexadécimal. Voir aussi *Base 16, Bit, Hexadécimal* et *Octet*.

R0, R1, R2, R3 et R4 sont les noms des registres de sauvegardes. On y sauvegarde les registres de travail.

Racine Voir *Répertoire racine*.

Rafraîchissement Renouvellement régulier du contenu de l'écran à l'aide d'une image identique à la précédente si l'affichage n'a pas été modifié.

RAM Voir *mémoire vive*.

RAM I/O Voir *mémoire vive des entrées/sorties*.

RAM système Voir *mémoire vive réservée au système*.

Ramasse-miettes Voir *Garbage Collection*.

Récupérateur de mémoire Voir *Garbage Collection*.

Registre Petite mémoire interne du microprocesseur. Le microprocesseur effectue ses traitements à partir du contenu des registres. Voir *registre*.

Registre de calcul Voir *Registre de travail*.

Registre de sauvegarde Voir *R0, R1, R2...*

Registre de travail Registres A, B, C et D. Ce sont les registres qui offrent le plus de possibilités de traitement.

Répertoire Sorte de dossier dans lequel se trouve un ensemble de variables. Un répertoire peut contenir d'autres répertoires qui peuvent, eux-mêmes, en contenir encore d'autres. C'est cet emboîtement de répertoires qui définit l'arborescence de la mémoire. En fait, un répertoire est une variable contenant des noms de variables.

Répertoire racine Ce répertoire contient tous les autres et se trouve donc à la « racine » de l'arborescence des répertoires. Ce répertoire s'appelle HOME dans le cas de la HP-48. Voir *Répertoire*.

Retour de sous-programme Saut vers la dernière adresse placée dans la pile RSTK.

ROM Voir *mémoire morte*.

RPL (environnement RPL) Nom souvent donné à l'interface de la HP-48 dans le

cadre de son utilisation normale (pile, langage RPL, fonctions et outils intégrés, etc.).

RPL (langage) *Rom-based programming language* Langage de programmation standard de la HP-48. A la fois apparenté au LISP et au FORTH. Utilise la notation polonaise inversée (RPN). Voir *RPN*.

RPN *Reverse Polish Notation* Notation post-fixe imposant l'introduction des arguments d'une fonction avant celle-ci.

RS 232 Norme de communication en série utilisée pour la communication par câble entre la HP-48 et un micro-ordinateur.

RSTK *Return Stack* Pile des retours contenant les adresses de retours de sous-programme. Cette pile est apparentée à un registre de 8 niveaux de 5 quartets. Chaque niveau peut accueillir une adresse. Deux de ses niveaux doivent rester disponibles en permanence

SATURN Microprocesseur de la HP-48. Voir *Microprocesseur*.

Saut Voir *Branchement*.

ST *Status Bits* Registre de 16 bits. Chacun de ces bits a un rôle de drapeau pour le fonctionnement du SATURN. Voir *Drapeau Registre* et *SATURN*.

Table de vérité Tableau résumant le comportement d'une fonction logique pour toutes les valeurs possibles de ses arguments.

Tampon Voir *Mémoire tampon*.

Témoin Symbole qui peut être affiché en haut de l'écran pour indiquer un mode de saisie particulier (touche α , touches shift), une sonnerie, une transmission en cours, etc.

Test Comparaison du contenu d'un registre de travail avec zéro ou avec le contenu d'un autre registre de travail. Le test peut vérifier l'égalité, l'inégalité, la supériorité stricte ou large et l'infériorité stricte ou large. Si la condition testée est remplie, un saut relatif court est exécuté. Voir *Saut*.

Tétrade Voir *Quartet*.

Variable Association d'un objet à un nom afin de stocker en mémoire cet objet sous le nom qui vient de lui être affecté.

Visibilité des mémoires La mémoire visible est la mémoire accessible à l'instant considéré. Voir *Masquage des mémoires*.

Visibilité des variables Les seules variables visibles sont celles appartenant aux répertoires du chemin courant. Voir *Répertoire*, *Variable*, *Chemin d'accès* et *Chemin courant*.

W (champ) Ce champ correspond aux 16 quartets d'un registre de travail (le champ W correspond à l'intégralité du registre).

WP (champ) Ce champ est l'ensemble de quartets depuis le quartet n°0 d'un registre de travail jusqu'à son quartet n°P. P est ici la valeur (comprise entre #0h et #Fh) que contient le registre P. Voir aussi *P (champ)* et *P (registre)*.

XOR Voir *OU logique exclusif*.

Bibliographie



Ci-dessous, la liste des livres consultés lors de la rédaction du présent ouvrage.

P. Courbis, S. Lalande. – *Voyage au centre de la HP-48*, Angkor, 1991.

P. Courbis. – *Voyage au centre de la HP-48 G/GX*, Angkor, 1993.

J.-M. Ferrard. – *Les secrets de la HP-48*, D3i, 1993.

M. Cornillault. – *HP-48 : Mathématiques en prépa*, Dunod, 1994.

Hewlett-Packard – *HP-48 : Advanced user's reference manual*, 1993.

Ci-dessous, la liste des bulletins et périodiques consultés lors de la rédaction du présent ouvrage.

Haute Performance, n°2 à 11, édité par le club FFJM,

Poche Computers, n°2 et 3, édité par Microcards,

CHP, bulletin du club hexagone programmation,

48SXTANT, bulletin de Robert Pulluard.

Textes divers en provenance de BBS.

De très nombreux documents (adresses SYSEVAL, instructions, tables de codes, etc.) ont été consultés sur différents serveurs WEB d'Internet. Nous avons recensé plus de mille adresses (ou liens) où des informations de référence sont disponibles.

Les clubs d'utilisateurs



■ Clubs francophones d'utilisateurs de HP-48

48SX Tant

Adresse : Lijsterlaan 31, 2665 TH Bleiswijk (Pays-Bas)

Prix au numéro : 35 FF

Activité : bulletin d'information et de programmation

Remarque : l'un des clubs francophones les plus sérieux, n'hésitez pas à le contacter bien qu'il se trouve en Hollande.

Club Calculatrices

Adresse : SETORGA, BP 203-16, 75765 Paris Cedex 16

Cotisation : aucune

Activité : bulletin d'information et de programmation

Remarque : club ouvert à tous les types de calculatrices

CHP - Club Hexagone Programmation

Adresse : CHP, 9 route d'Orléans, 63200 Riom

Cotisation : 30 FF par numéro

Activité : bulletin d'information et de programmation

Remarque : petit club très sympathique

■ Serveur Minitel

3615 CALCULATOR

Editeur : Suite 1024

Coût de la connexion : 0,13 F puis 3 U.T./min.

Remarque : le plus important serveur dédié aux calculatrices

Assistance technique

La « hot-line » des utilisateurs de calculatrices

**Un problème dans le cadre de l'utilisation de ce livre ?
Une question urgente ?**

Le **CLUB CALCULATRICES** met à votre disposition un service d'assistance technique et répond à vos questions en 48 heures.

Voici comment utiliser ce service :

- à l'aide d'un simple minitel, connectez-vous au service télématique 3615 CALCULATOR,
- sur ce service créez une boîte aux lettres (B.A.L.) pour pouvoir recevoir des messages,
- posez votre question en laissant un message dans la boîte aux lettres CLUB,
- vous trouverez dans votre boîte aux lettres la réponse à votre question (délai à prévoir : 24 à 48 heures).

Ne restez pas bloqué par un petit problème ! Le CLUB est là pour vous aider !

Correspondance

Contact :

*Loïc Fieux / Collection "Calculatrices"
Dunod
15 rue Gossin
92543 Montrouge Cedex*

Attention ! Toutes vos questions concernant cet ouvrage doivent être posées au service d'assistance technique et non en écrivant à cette adresse.

VOTRE CALCULATRICE A TROUVE SON MAÎTRE !

- Plus de 12.000 écrans !
- Des milliers de programmes !
- Informations, Scoops...
- Trucs et Astuces
- Messagerie et Contacts
- Petites Annonces
- Echanges, Achats, Ventes
- Micro-Informatique
- Casio™, HP™, TI™, Sharp™, etc.



Éditrice : SUITE 1024 • 0 13 1 6 • puis 3 0 1 1 mini

3615

Le minitel de votre HP !

CALCULATOR

GRATUIT !

Le CLUB CALCULATRICES

Vous avez des idées, vous trouvez des astuces, vous programmez et vous souhaitez savoir ce que d'autres ont pu découvrir...

Quel que soit votre niveau, sachez qu'il existe un club d'utilisateurs de calculatrices graphiques (fx, TI, hp, EL...). Pour plus d'informations sur le club, connectez-vous au service télématique 3615 CALCULATOR, créez une boîte aux lettres (BAL) pour pouvoir recevoir des messages et écrivez un message de demande d'informations à destination de la boîte aux lettres CLUB.

Cochez les cases !

- Oui, je souhaite devenir gratuitement membre du **Club Calculatrices**, je ne prends aucun engagement en envoyant ce coupon.
- Je vous envoie des astuces et/ou des programmes afin qu'ils soient publiés sur le service télématique 3615 CALCULATOR. Je ne vous envoie ni programme ni astuce.

Nom : Prénom :

Adresse :

Code postal Ville : Pays :



TI JAG

Complétez le coupon ci-dessus et envoyez-le à :

SETORGA • CLUB CALCULATRICES • BP 203-16 • 75765 Paris Cedex 16

Initiation et parascolaire sur HP-48 !

- ***pour bien débiter***

HP-48 : Permis de conduire

Acquérir "l'esprit HP-48" facilement et en quelques instants... C'est possible avec ce nouveau livre permettant au débutant de maîtriser sa HP-48 G/GX très rapidement. Oubliez les épais manuels ! En 192 pages, ce livre très illustré (plus de 300 copies d'écrans !) vous présente tous les grands principes pour dompter et exploiter votre HP-48 G/GX.

192 pages, 98 FF

- ***au lycée***

HP-48 pour le bac

Trois livres en un :

- pour le débutant, quelques notions pour exploiter facilement les capacités de base de la HP-48,
- un véritable concentré du cours de maths de Terminale scientifique, chapitre par chapitre, avec toutes les notions nécessaires à une bonne compréhension des problèmes que vous devrez savoir traiter à l'examen,
- un recueil d'annales du bac scientifique, avec de nombreux sujets très récents, accompagnés de leurs solutions rédigées comme vous aurez à le faire à l'examen. Et avec, en plus, tout ce que va vous permettre votre HP-48 pour gagner des points !

256 pages, 125 FF

Vous le trouverez dans...

HP-48

Maths en prépa

par M. Cornillault, M. de Courville et E. Lesueur (DUNOD), 512 pages avec disquette.

■ **Programmes de base** : viewer, recherche de chaîne, décomposition et recombinaison de programme, rappel d'un objet en mémoire, assembleur, désassembleur, gestion de l'aide en ligne, redéfinition du clavier...

■ **Arithmétique** : décomposition en facteurs premiers, PGCD, PPCM, transformation en fraction, transformation de deux nombres en fraction, addition, différence, produit et quotient de deux fractions, réduction au même dénominateur, exponentiation d'une fraction, fraction opposée, recherche du plus grand nombre, carré d'une fraction rationnelle...

■ **Calculs sur les polynômes** : simplification, valuation, valeur, addition, soustraction, produit, division, produit par une constante, puissance, division par puissances croissantes, PGCD de deux polynômes, composée de deux polynômes, translation d'un polynôme, dérivée d'un polynôme, intégration d'un polynôme, conversion des coefficients d'un polynôme...

■ **Racines** : décomposition en diviseurs, racines d'un polynôme...

■ **Algorithmes** : Bezout, Newton, tri par insertion de liste...

■ **Développements limités (DL) à coefficients algébriques** : opposé d'un DL, calcul de développements limités (Taylor), addition, différence, produit, quotient de deux DL, valuation d'un DL, DL de $(1+X)^a$ en zéro, simplification d'un DL, composée de deux DL, exponentiation d'un DL, DL de fonctions (*sinus, cosinus, sinus hyperbolique, cosinus hyperbolique, exponentielle, logarithme népérien, arc sinus, arc sinus hyperbolique, arc tangente, arc tangente hyperbolique, tangente, tangente hyperbolique*) en zéro ou en un selon le cas...

■ **Calculs matriciels** : trace d'une matrice rationnelle, transformation d'une matrice entière et son dénominateur en une matrice rationnelle, transformation d'une matrice rationnelle en une matricé entière et son dénominateur, addition, différence, multiplication de deux matrices, algorithme de Leverrier, inversion d'une matrice, comatrice, polynôme caractéristique...

■ **Matrices algébriques** : transformation d'une matrice classique en matrice algébrique, transformation d'une matrice algébrique en matrice classique, dimensions d'une matrice, opérations sur les matrices, somme de deux matrices, différence de deux matrices, produit d'une matrice par une constante, produit de deux matrices, trace d'une matrice carrée, transposée d'une matrice...

■ **Applications des matrices** : Jordanisation, *n*-ième ligne d'une matrice, PGCD des éléments d'une matrice, transformation d'une liste de vecteurs en matrice, fabrication d'une famille de vecteurs libres, fabrication d'une matrice d'après une formule générique donnée, algorithme de Gauss, résolution symbolique d'un système quelconque, équations des espaces propres d'une matrice, etc.

■ **Equations différentielles** : résolution graphique des équations différentielles du premier et du deuxième ordre scalaire, etc.

■ **Géométrie et trigonométrie** : traceur d'enveloppes de droites, traceur de développée, décomposition de coniques, linéarisation d'une expression trigonométrique...

■ **Physique** : tracé et étude de fonctions de transfert, calcul de l'enthalpie...

■ **Chimie** : calcul du pH d'une solution, calcul de la masse moléculaire d'une molécule donnée...

■ **Mais aussi...**

■ **sujets des concours** corrigés avec les programmes du livre,

■ **trucs et astuces** : remplacement des messages d'erreurs, menus, ports, la commande PKT, programme PEEK, répertoires cachés, la commande WSLOG, fonctionnement du calculateur, version de votre HP-48, etc.

■ **assembleur** : introduction à l'assembleur, visite des registres, les types, les mnémoniques, etc.

■ **internals** : la programmation par internals, liste d'internals par familles, etc.

■ **extensions** : fabrication de bibliothèques, décomposition de bibliothèques, fabrication de matrices, fabrication d'un affichage, compression de données, décompression de données, etc.

Vous le trouverez dans...
HP-48
Physique et chimie
en prépa

par H. Canon, A. Ramboux et L. Fieux (DUNOD), 256 pages avec disquette.

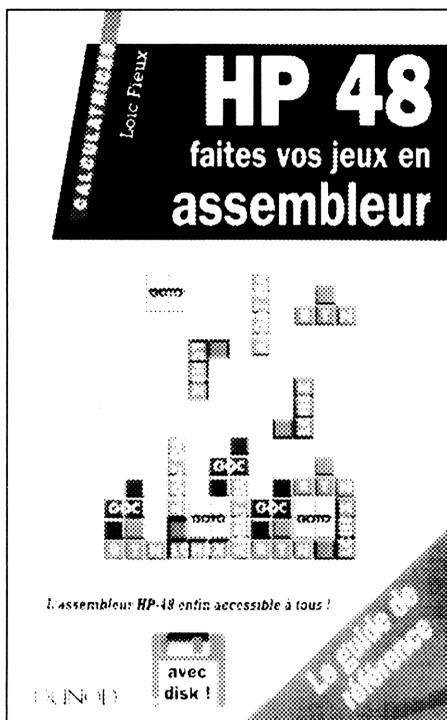
- Equilibres chimiques en solution aqueuse
- Dosages acidobasiques
- Dosages acidobasiques par conductimétrie
- Diagramme de prédominance acidobasique
- Pouvoir tampon d'une solution
- Dosage de complexes
- Dosage de complexes par conductimétrie
- Piles et couples d'oxydoréduction
- Cinétique : ordre d'une réaction
- Tableau périodique des éléments
- Diagramme de Bode

- Equations différentielles
- Tracés polaires et paramétriques
- Satellites
- Oscillateurs harmoniques
- Circuits électriques
- Résonance d'un circuit RLC
- Optique

Passez à la vitesse supérieure !

HP-48 : faites vos jeux en assembleur !

Découvrez le langage assembleur sur HP-48 et, aidé par ce guide de référence, réalisez vos premiers jeux en assembleur ! Plus de 700 pages !



La collection

CALCULATRICES EFFICACES

*Les grands atouts
de la « petite informatique »*

- Casio fx 7900/9900 : permis de conduire
- Votre Casio fx 6800G
- Casio fx/CFX : 500 programmes
- Casio fx/CFX : le top des jeux !
- Casio fx/CFX : faites vos jeux !
- Casio fx/CFX : jeux et graphisme
- Casio fx pour le bac !
- Casio fx : programmez votre succès ! (Casio fx 6000 6500 6300 6800 7000 7200 7500 7700 7800 8000 8500 8700 8800)
- Casio fx : maths au lycée (Casio fx 6000 6500 6300 6800 7000 7200 7500 7700 7800 8000 8500 8700 8800)
- Casio fx : programmation efficace
- HP-48 G/GX : permis de conduire
- HP-38/48 G/GX pour le bac !
- HP-48 G/GX/S/SX : mathématiques en prépa
- HP-48 G/GX/S/SX : physique et chimie en prépa
- HP-48 G/GX/S/SX : faites vos jeux en assembleur !
- HP-48 G/GX/S/SX : le livre du programmeur en RPL et RPL-système
- TI-81 : programmez votre succès !
- TI-81 : jeux et graphisme
- TI-81 : le "top" des jeux !
- TI-82 : permis de conduire
- TI-82 : mathématiques au lycée
- TI-82 : sciences physiques au lycée
- TI-82 : programmes pour le lycée
- TI-82 : le "top" des jeux !
- TI-82/85 : jeux et graphisme
- TI-85 : permis de conduire
- TI-85 du lycée à la prépa
- TI-85 : le "top" des jeux !
- TI-92 : du lycée à la prépa
- TI-92 : les programmes
- TI-92 : le top des jeux
- Sharp EL 9200/9300 : faites vos jeux !

SNEL S.A.
Rue Saint-Vincent 12 - 4020 Liège
tél. 32(0)4 343 76 91 - fax 32(0)4 343 77 50
septembre 1996

Dépôt légal : septembre 1996

HP 48

Le livre du programmeur en RPL et RPL-système

Dans la même collection :

- HP-48 permis de conduire
- HP-48 pour le bac
- HP-48 : maths en prépa
- HP-48 : physique et chimie en prépa
- HP-48 : faites vos jeux en assembleur

La sécurité du RPL,
la rapidité
du RPL-système !

Une HP 48 sur mesure !

Profitez de toutes les fonctions intégrées de votre calculatrice en créant rapidement de puissants programmes en RPL ! Conçu pour le débutant cet ouvrage vous explique pas à pas comment concevoir et réaliser un programme. Vous y trouverez de nombreux exemples de structures de base facilement adaptables à vos besoins.

Chaque instruction du langage RPL pouvant être décomposée en sous-instructions appelées primitives, la programmation en RPL-système vous permet de créer des instructions sur mesure à partir de ces primitives. A mi-chemin entre le symbolisme du RPL et la complexité de l'assembleur, le RPL-système sait rester simple tout en autorisant la réalisation de programmes rapides et une maîtrise de la machine plus poussée qu'en RPL. Vous trouverez dans cet ouvrage les explications pour bien débiter en RPL-système.

Toutes les informations nécessaires pour programmer en RPL ou en RPL-système vous sont fournies : liste des instructions RPL classées par familles, guide alphabétique des instructions RPL, structure en mémoire des objets élémentaires, plus de 2000 adresses utilisables en RPL-système, codes des menus, codes ASCII, fonctions des indicateurs-système, variables réservées, codes des messages système, etc.

Bref, de l'initiation au développement, tout pour tirer le meilleur de votre HP-48 !



9 782100 030378

ISBN 2 10 003037 X Code 043 037

Environnement : **Calculateurs
Hewlett-Packard HP 48 G, GX, S et SX**

Thème : **Langage RPL et RPL-système**

Utilisateur : **Etudiant, Ingénieur, etc.**



E D I T E U R