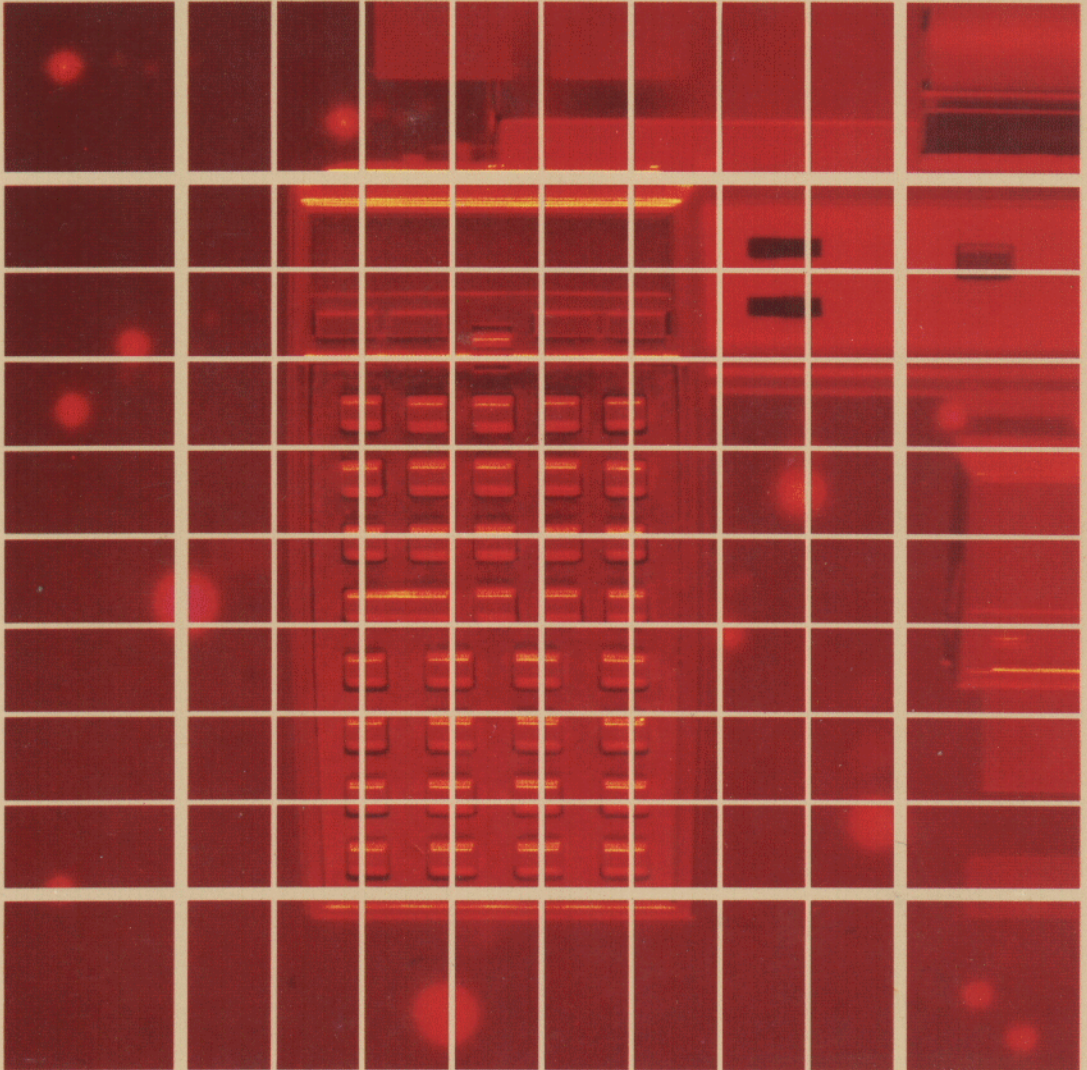


HEWLETT-PACKARD

HP-41CX

MANUEL D'UTILISATION

VOLUME 2: FONCTIONNEMENT EN DÉTAIL



Résumé des conventions typographiques utilisées dans le présent manuel

Notation (exemple)	Description
STO	<i>Cadre noir.</i> Fonction primaire sur le clavier.
10°	<i>Cadre jaune.</i> Fonction secondaire sur le clavier. Actionnez puis relâchez préalablement la touche jaune (■). Ce genre de fonction peut se trouver sur le clavier normal ou sur le clavier alphabétique.
DATE	<i>Cadre bleu.</i> Fonction non présente sur le clavier. Pour exécution en mode Alpha : actionnez XEQ puis introduisez le nom de la fonction sur le clavier alphabétique. Pour exécution en mode personnel : affectez la fonction à une touche du clavier personnel. (Pages 45, 46.)
ABC	<i>Caractères bleus.</i> Caractères alphabétiques.
123	<i>Caractères ou chiffres jaunes.</i> Caractères alphabétiques secondaires.
T	<i>Lettres noires encadrées.</i> Ce sont des fonctions spéciales et non des caractères alphabétiques ; elles ne sont opérantes que dans des conditions particulières. S'il s'agit d'une fonction secondaire, elle est précédée d'un carré jaune (symbole de la touche jaune qui sert de touche « préfixe » pour passer des fonctions primaires aux fonctions secondaires).
■ T	
■ T	
<i>paramètre</i>	<i>Le type de paramètre requis pour une fonction.</i>

Pour une description complète, reportez-vous au paragraphe « Représentation des touches », page 16.



HP-41CX

Manuel d'Utilisation

Volume 2

Fonctionnement en détail

Mars 1984

00041-90493

Présentation du volume 2

Ceci est le second volume du *manuel d'utilisation du HP-41CX*. Le volume 1, « *Principes de base* », constitue une introduction à la plupart des caractéristiques du HP-41CX. Le présent volume détaille *tous* les aspects du HP-41CX. Ces deux volumes constituent un seul manuel ; la pagination du présent volume fait donc suite à celle du volume 1, ainsi que les numéros des chapitres et des différentes parties.

Vous trouverez à la page 9 (volume 1) un paragraphe intitulé « Comment utiliser ce manuel », qui explique la structure du manuel et donne des conseils pour son utilisation. Jetez-y un coup d'œil pour avoir une vue d'ensemble des deux volumes. Le présent volume est destiné à servir de *référence* et à fournir des *renseignements exhaustifs*. Il contient toutes les annexes ainsi qu'un index de toutes les fonctions (en fin de volume) et un résumé complet de toutes les fonctions sous la forme de tableaux de fonction. Si vous connaissez déjà le HP-41, n'oubliez pas de lire l'annexe I, « Comparaison avec le HP-41C/CV ».

Volume 2 : Fonctionnement en détail

Table des matières

Partie II : Description détaillée des principes de base

Chapitre 9 : Le clavier et l'affichage	154
• Touches fonctionnant en bascule • Les claviers	
• Saisie de nombres et de caractères au clavier • Témoins d'état	
• Format d'affichage numérique • Messages et affichages standard	
• Défilement de l'affichage • Spécification de paramètres	
• Comment redéfinir le clavier personnel • Prévisualisation et annulation des fonctions • Les catalogues • Messages d'erreur	
Chapitre 10 : La pile opérationnelle	174
• Introduction • Calculs en notation polonaise inverse	
• Le registre LAST X • Autres opérations dans la pile opérationnelle	
Chapitre 11 : Fonctions numériques	184
• Introduction • Fonctions à un nombre (ou fonctions monadiques) • Fonctions à deux nombres (ou diadiques) • Statistiques	

Partie III : Description détaillée de la mémoire

Chapitre 12 : Mémoire principale	194
• Organisation • Mémoire de programmation • Mémoire des alarmes	
• Mémoire du clavier personnel • Opérations sur les registres de stockage des données	
Chapitre 13 : Mémoire annexe	204
• Introduction • Fichiers en mémoire annexe	
• Opérations sur les fichiers de programme • Création de fichiers de données et de texte	
• Pointeurs des fichiers de données et de texte • Opérations sur les fichiers de données • Opérations sur les fichiers de texte	
Chapitre 14 : L'éditeur de texte	228
• Introduction • L'affichage de l'éditeur de texte	
• Opérations dans l'éditeur de texte (ED). • Utilisation de ED dans un programme	

Partie IV : Description détaillée des fonctions d'horloge

Chapitre 15 : Les fonctions d'horloge	236
• Mise à la date et mise à l'heure • Affichage de l'heure • Opérations sur les valeurs horaires • Opérations sur la date • Calculs de dates • Limites et erreurs	
Chapitre 16 : Alarmes	246
• Types d'alarmes • Armement des alarmes • Activation et annulation d'alarmes-messages • Catalogue d'alarmes • Effacement d'alarmes de la mémoire • Alarmes périmées • Programmes armement d'alarme	
Chapitre 17 : Fonctionnement du chronomètre	266
• Clavier du chronomètre • Prises de temps intermédiaires • Fonctions programmables du chronomètre • Chronomètre en décompteur • Impression de temps intermédiaires • Exemple : Programme de chronomètre	

Partie V : Programmation en détail

Chapitre 18 : Bases de programmation	280
• Chargement d'un programme • Exécution d'un programme • Lignes de programme • Opérations non programmables • Déplacement du pointeur en mémoire programme • Mise au point d'un programme • Annulation de programmes	
Chapitre 19 : Indicateurs binaires	288
• Introduction • Types d'indicateurs binaires • Résumé de l'état des indicateurs binaires • Indicateurs binaires et registre X	
Chapitre 20 : Branchements	298
• Introduction • Branchement à un label • Appel d'un sous-programme • Fonctions conditionnelles • Boucles	
Chapitre 21 : Opérations alphabétiques et interactions	308
• Introduction • Registres X et ALPHA • Manipulation de chaînes alphabétiques • Demande d'intervention • Réponse à l'appui sur une touche • Sorties	
Chapitre 22 : Programmes d'enregistrement de temps	320
• Introduction • Exemples de programmes • Utilisation des programmes • Fichiers servant à l'enregistrement des temps • Explication de TR • Explication de SIGMA	

Annexes

Annexe A : Messages d'erreur et d'état	354
Annexe B : Pour en savoir davantage sur les alarmes périmées	360
Annexe C : Caractères nuls	366
Annexe D : Opérations d'impression	368
Annexe E : Modules d'extension mémoire	370
Annexe F : Spécifications de l'horloge	374
Annexe G : Piles, garantie et maintenance	380
Annexe H : Périphériques, extensions et HP-IL	392
Annexe I : Comparaison avec le HP-41C/CV	398
Annexe J : Codes-barres	404
Tableau de fonctions	414
Index	440
Index des fonctions	3 ^e de couverture

Liste des schémas et tableaux

Schémas

Le clavier alphabétique	157
Touches spéciales permettant de spécifier les paramètres	165
Le clavier personnel	167
Configuration de la mémoire principale	195
Le clavier de l'éditeur de texte	231
Organigramme des alarmes	249
Touches actives sur le clavier du catalogue des alarmes	256
Clavier du chronomètre	269
Codes des touches pour <code>GETKEY</code> et <code>GETKEYX</code>	316
Configurations possibles pour deux modules d'extension mémoire HP 82181A	370
Organisation de la mémoire annexe	373
La fonction <code>CORRECT</code>	376

Tableaux

Registres statistiques	191
Mise à l'heure	237
Formats d'affichage	239
Valeurs horaires	240
Adjonction d'un temps dans le registre Alpha	241
Options de format de la date	242
Ajout d'une date au registre Alpha	243
Résultat de l'annulation de l'alarme	254
Clavier du catalogue d'alarmes	257
Fonctionnement du clavier du chronomètre	268
Résumé des états des indicateurs binaires	293
Valeurs décimales et indicateurs binaires 00 à 07	294
Codes des caractères	310
Réponses des alarmes périmées	362
Le facteur de précision	375
Affichage d'une instruction de programme	396
Configuration de la mémoire	399
Termes équivalents	403

Table des matières du volume 1 : Principes de base

Comment utiliser ce manuel

Partie I : Utilisation du HP-41

Chapitre 1 : Utilisation du clavier

Chapitre 2 : L'affichage

Chapitre 3 : Mise en mémoire et rappel de nombres

Chapitre 4 : Comment exécuter les fonctions de votre HP-41

Chapitre 5 : Fonctions standard du HP-41

Chapitre 6 : Les fonctions d'horloge

Chapitre 7 : Programmation élémentaire

Chapitre 8 : Stockage de texte, données et programmes dans des fichiers

Liste des erreurs

Index

Index des fonctions

Le clavier et l'affichage

Table des matières

Touches fonctionnant en bascule	155
Les claviers	155
Le clavier normal	156
Le clavier personnel	156
Le clavier alphabétique	156
Claviers spéciaux	158
Saisie de nombres et de caractères au clavier	158
Saisie numérique	159
Saisie de caractères	159
Témoins d'état	160
Format d'affichage numérique	160
Format des nombres	160
Ponctuation	161
Messages et affichages standard	161
Défilement de l'affichage	162
Spécification de paramètres	162
Spécification indirecte de paramètres	162
Touches spéciales	164
Comment redéfinir le clavier personnel	166
Le catalogue du clavier personnel	168
Retour aux fonctions normales	168
Les deux rangées supérieures de touches	169
Prévisualisation et annulation des fonctions	169
Les catalogues	170
Principes d'utilisation des catalogues	170
Différents types de catalogues	171
Messages d'erreur	172

Touches fonctionnant en bascule

Juste au-dessous de l'affichage se trouvent quatre touches fonctionnant en bascule : **ON**, **USER**, **PRGM** et **ALPHA**. Elles commandent la façon dont votre HP-41 interprète les autres touches du clavier. Ces touches fonctionnent « en bascule », c'est-à-dire que, lorsque vous les actionnez, elles donnent au clavier une interprétation particulière jusqu'à ce que vous les actionniez à nouveau, ramenant alors le clavier à son état antérieur.

La touche ON. Cette touche met votre HP-41 sous et hors tension. Lorsqu'il n'a pas été utilisé pendant environ 10 minutes, le HP-41 se met automatiquement hors tension, afin de prolonger la durée de vie des piles ou batteries*. Pendant qu'il est hors tension, la mémoire permanente sauvegarde le contenu des mémoires principale et annexe, ainsi que l'état de certains indicateurs binaires. Pour réinitialiser le HP-41 (c'est-à-dire pour effacer les mémoires — principale et annexe (modules d'extension mémoire) — et ramener tous les indicateurs binaires à leur état par défaut) :

1. Mettez le HP-41 hors tension.
2. Appuyez sur la touche **+** et maintenez-la enfoncée.
3. Appuyez sur **ON**.
4. Relâchez **+**.

Le message **MEMORY LOST** (mémoire perdue) apparaît.

La touche USER. Cette touche permet de valider et invalider le clavier personnel, qui est la version du clavier normal redéfinie par vous-même en fonction de vos besoins spécifiques. Lorsque le clavier personnel est validé, le témoin **USER** apparaît (et l'indicateur binaire 27 est armé).

La touche PRGM. Cette touche fait passer le HP-41 du mode exécution au mode programme et inversement. Lorsque vous mettez votre HP-41 sous tension, il se trouve en mode exécution — et vous pouvez donc exécuter fonctions et programmes. En mode programme, vous pouvez écrire et éditer (modifier) des programmes; les fonctions sont stockées comme des pas de programme, à exécuter ultérieurement lorsque le programme est exécuté en mode exécution. Le témoin **PRGM** indique que le HP-41 est en mode programme ou qu'un programme est en cours d'exécution.

La touche ALPHA. Cette touche valide et invalide le clavier alphabétique, qui comprend les caractères bleus figurant sur la face biseautée des touches. Le témoin **ALPHA** apparaît (et l'indicateur binaire 48 est armé) lorsque le clavier alphabétique est validé. En appuyant sur **ON** ou **PRGM**, vous invalidez le clavier alphabétique.



Les claviers




Dans le présent manuel, chaque nom de fonction est représenté dans une couleur qui indique comment exécuter la fonction. La vue d'ensemble des claviers, qui figure à la page ci-après, illustre cette utilisation des couleurs, ainsi que le rôle de chaque clavier.






* A moins que vous n'exécutiez la fonction **ON**, qui arme l'indicateur binaire 44 (mise sous tension continue). L'indicateur 44 est désarmé chaque fois que vous mettez le HP-41 sous tension.

Le clavier normal


Le clavier normal se compose des fonctions inscrites en blanc sur le dessus des touches et des fonctions inscrites en jaune au-dessus des touches. C'est le clavier par défaut — celui qui est opérant après que la mémoire permanente ait été effacée.


Lorsque vous appuyez sur , le témoin **SHIFT** apparaît, indiquant qu'une fonction secondaire va être exécutée. Ce témoin disparaît ensuite lorsque vous appuyez sur une seconde touche (pour exécuter une fonction secondaire) ou lorsque vous appuyez à nouveau sur la touche  (pour annuler la commande SHIFT, qui sélectionne les fonctions secondaires).

Dans le présent manuel, une fonction primaire est représentée par son nom en noir encadré de noir, et une fonction secondaire (avec SHIFT) par son nom en jaune encadré de jaune. Par exemple,  est la fonction primaire de la touche située dans l'angle supérieur droit, et  est la fonction secondaire de cette même touche. Cette règle s'applique aux autres claviers également ; ainsi,  est une fonction secondaire sur le clavier alphabétique.

Lorsqu'une touche a une fonction spéciale identifiée par une lettre sur sa face biseautée, cette touche est représentée par cette lettre en noir et encadrée de noir. Par exemple, la séquence qui a pour résultat  serait  , où  représente la touche .


Le clavier personnel

Le clavier personnel est votre version personnalisée du clavier normal. Vous pouvez affecter une fonction ou un label global à n'importe quelle touche, sauf les touches fonctionnant en bascule et la touche  (SHIFT). Vous pouvez ensuite accéder à cette fonction ou commencer l'exécution du programme à ce label global en validant le clavier personnel et en appuyant sur la touche ainsi redéfinie.

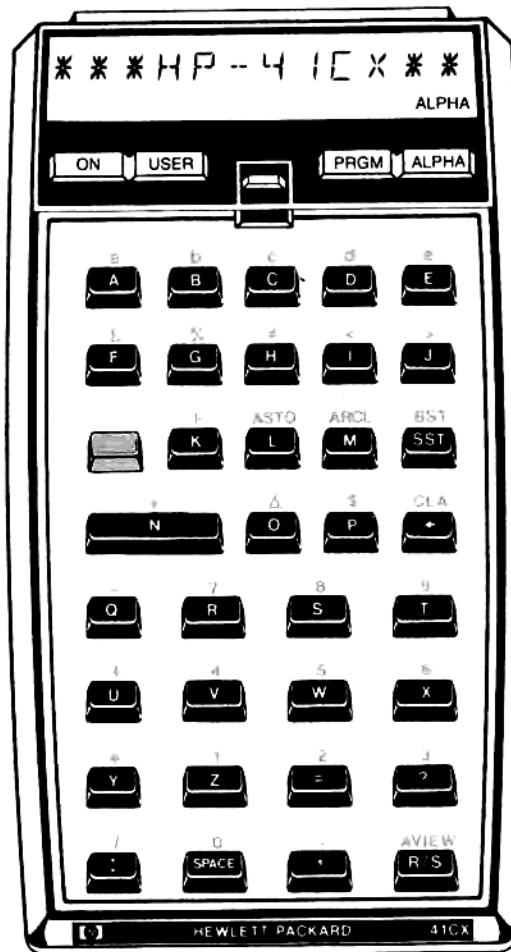
Comme vous pouvez également personnaliser les fonctions secondaires des touches, une touche peut exécuter quatre fonctions différentes selon que le clavier personnel est opérant ou non et que la touche  est actionnée ou non. Le fonctionnement du clavier personnel est décrit page 166.

De nombreuses fonctions ne figurent pas sur le clavier normal, mais peuvent être affectées au clavier personnel. Dans le présent manuel, les fonctions ne figurant pas sur le clavier sont représentées par leur nom en bleu et encadrées de bleu.

Le clavier alphabétique

Le clavier alphabétique comprend des lettres, des fonctions, des symboles et des chiffres qui sont traités comme des caractères alphabétiques. Les lettres et symboles bleus figurant sur la face biseautée des touches sont les caractères primaires. Les chiffres de 0 à 9 et les symboles arithmétiques sont des caractères secondaires qui figurent sur le dessus des touches. Vous trouverez à la page ci-après une vue d'ensemble du clavier alphabétique qui comprend les symboles et fonctions secondaires (obtenus après pression de ).

Le clavier alphabétique



Le clavier alphabétique a deux utilisations distinctes ; il peut servir :

- à inscrire le nom d'une fonction ou un label global en tant que paramètre d'une fonction **[ASM]** , **[CLP]** , **[COPY]** , **[GTD]** , **[LBE]** ou **[XEQ]** . Dans ce cas, les caractères deviennent une partie de l'instruction ;
- à introduire des caractères dans le registre alphabétique. Ils sont alors conservés dans ce registre jusqu'à ce que d'autres caractères viennent les écraser et que vous effaciez le registre. Le registre alphabétique sert à afficher vos propres messages, à spécifier des noms de fichiers et des labels globaux pour certaines fonctions, et à manipuler des données.

Dans le présent manuel, les caractères primaires du clavier alphabétique sont représentés en bleu, et les caractères secondaires en jaune. Un chiffre en jaune représente donc un caractère alphabétique, alors qu'un chiffre en noir représente un chiffre du clavier normal.

Claviers spéciaux

Outre les claviers validés par les touches **[USER]** et **[ALPHA]** , il existe des claviers spéciaux, qui sont validés par des fonctions. Les principaux sont les suivants :

- Le clavier de l'éditeur de texte, validé par **[ED]** . Basé sur le clavier alphabétique, ce clavier comprend des commandes permettant de manipuler un fichier de texte en mémoire annexe. La fonction **[ED]** est décrite au chapitre 14.
- Le clavier du catalogue d'alarmes, validé par **[ALMCAT]** et **[CATALOG]** 5. Ces fonctions vous permettent d'examiner et modifier des alarmes en mémoire. Les alarmes sont décrites au chapitre 16.
- Le clavier du chronomètre, validé par **[SW]** et **[SWPT]** . Les fonctions du chronomètre sont décrites au chapitre 17.

Saisie de nombres et de caractères au clavier

L'introduction de nombres dans le registre X et de caractères dans le registre alphabétique s'effectue de manière identique dans les deux cas :

- Lorsque vous introduisez le premier chiffre ou caractère, l'affichage montre ce chiffre ou ce caractère suivi de *l'indicateur de saisie* (—).
- L'indicateur de saisie montre que la prochaine saisie effectuée au clavier va être annexée à la chaîne de chiffres ou de caractères déjà affichée.
- Tant que l'indicateur de saisie est affiché, vous pouvez corriger les données introduites en appuyant sur **[←]** pour supprimer le chiffre ou caractère le plus à droite*. L'indicateur de saisie se déplace alors d'une position vers la gauche.
- Si l'indicateur de saisie n'est pas affiché, cela signifie que la saisie est terminée et que le prochain chiffre ou caractère introduit constituera le début d'un nouveau nombre ou d'une nouvelle chaîne alphanumérique*.

* Si vous introduisez 10 chiffres ou un exposant de 10 à deux chiffres (fonction **[EEEX]**) dans le registre X, l'indicateur de saisie disparaît car l'affichage ne peut plus contenir de chiffres supplémentaires. Cela ne signifie pas que la saisie est terminée ; le chiffre suivant *ne constituera pas* le début d'un nouveau nombre, et une pression sur **[→]** *supprimera* le chiffre le plus à droite.

Saisie numérique

Vous pouvez introduire jusqu'à 10 chiffres dans le registre X — les chiffres supplémentaires sont ignorés. Les seules touches utilisées pour la saisie numérique sont les touches numériques [0] à [9], [=], [CHS] (*change sign* = changement de signe), [EEX] (*enter exponent* = entrer exposant), et [+]. Toute pression sur une touche autre qu'une touche numérique, [ON] ou [USER] met fin à la saisie numérique, et les chiffres introduits ensuite constituent un nouveau nombre.

En appuyant sur [0], vous remplacez le nombre se trouvant dans le registre X par zéro ; si vous introduisez ensuite un nouveau nombre, il vient remplacer le zéro. Si l'affichage ne comprend qu'un seul chiffre, ou si la saisie numérique est terminée, [=] a le même effet que [0].

Introduction d'un exposant. Pour introduire un nombre sous la forme $a \times 10^b$, introduisez d'abord au clavier les chiffres et le point décimal de a , puis appuyez sur [CHS] si le nombre est négatif. Si a comporte plus de 8 chiffres, vous devez introduire un point décimal quelque part à gauche du 9^e chiffre.

Appuyez ensuite sur [EEX]. Tout chiffre situé à droite du huitième chiffre disparaît de l'affichage, mais reste en mémoire. Introduisez un ou deux chiffres pour l'exposant b et appuyez sur [CHS] si b est négatif. Si vous appuyez sur [EEX] sans avoir préalablement introduit une valeur pour a , ce paramètre prend la valeur 1 par défaut.

Introduction de π . Une pression sur [PI] équivaut à l'introduction de 3,141592654 et entraîne automatiquement la fin de la saisie numérique.

Saisie de caractères

En mode exécution. Si le clavier alphabétique est validé et si vous n'êtes pas en train d'introduire un paramètre, les caractères vont dans le registre alphabétique. Pour introduire au clavier des caractères dans le registre alphabétique sous contrôle d'un programme, exécutez [AON] avant que le programme ne s'interrompe pour la saisie, puis [AOFF] lorsque son exécution reprend.

Le registre alphabétique peut contenir jusqu'à 24 caractères. Lorsque vous introduisez le 24^e caractère, le HP-41 émet un bip pour vous avertir que le registre alphabétique est plein. Si vous introduisez un caractère de plus, le caractère le plus à gauche disparaît de l'affichage et est perdu.

Vous mettez fin à la saisie des caractères en appuyant sur [AL-PO], [BKT], [SST], [APPRW], [RS] ou lorsque vous invalidez le clavier alphanumérique. Vous pouvez revenir à la saisie de caractères en appuyant sur [I-] (*append* = annexer) ou [AVCL].

Une pression sur [CLA] supprime tous les caractères du registre alphabétique. Si la saisie est terminée, une pression sur [=] a le même effet que [CLA].

En mode programme. Une ligne de programme peut contenir jusqu'à 15 caractères, qui sont précédés d'un ' sur l'affichage. Les caractères suivants vont dans le registre alphabétique lorsque le programme est exécuté. Pour ajouter une chaîne de caractères au registre alphabétique sans remplacer le contenu existant, commencez la chaîne avec [I-]. Par exemple, vous pouvez charger plus de 15 caractères dans le registre alphabétique en utilisant une seconde ligne de programme commençant par [I-]. (Le caractère ' apparaît seulement une fois que la ligne de programme est affichée ; la fonction d'annexion s'exécute au cours de l'exécution du programme).

Remarque : Lorsqu'elles sortent sur une imprimante ou un moniteur vidéo, les chaînes alphanumériques apparaissent entre guillemets. Seules les lignes de programme qui commencent et finissent par des guillemets sont des chaînes alphanumériques ; si une ligne de programme *ne sort pas* entre guillemets, il s'agit d'une fonction. Ne confondez pas un nom de fonction mal connu et une chaîne alphanumérique — n'oubliez pas d'appuyer sur **[XEQ]** avant d'introduire le nom d'une fonction.

Témoins d'état

Ces témoins apparaissent en bas de l'affichage, comme les témoins **USER**, **PRGM**, **ALPHA** et **SHIFT** mentionnés plus haut.

- **BAT** indique que l'alimentation par piles ou batteries commence à fléchir. Avec des piles alcalines, il vous reste environ 5 à 15 jours d'utilisation après l'apparition du témoin **BAT**. Avec les batteries rechargeables HP 82120A, il vous reste entre 2 et 50 minutes de fonctionnement. Si vous utilisez le lecteur de cartes magnétiques HP 82104A ou le crayon-lecteur optique HP 82153A, la durée d'utilisation restante est encore plus réduite. Pour plus de renseignements sur les piles et batteries, reportez-vous à l'annexe G.
- **GRAD** ou **RAD** indique que les fonctions trigonométriques et rectangulaires / polaires sont exprimées en grades ou en radians. Si aucun de ces deux témoins n'est affiché, les angles sont exprimés en degrés.
- Les chiffres **0 1 2 3 4** indiquent que l'indicateur binaire correspondant (00, 01, 02, 03, 04) est armé.

Certains témoins d'état ont des significations particulières lorsque **[ED]** est validé. Ils reprennent leur signification initiale lorsque vous invalidez **[ED]**.

Format d'affichage numérique

Chaque nombre est représenté en mémoire sous la forme $a \times 10^b$, a étant un nombre à 9 chiffres après la virgule, avec $1 \leq |a| < 10$, et b étant un nombre entier à deux chiffres, avec $0 \leq |b| < 100$. Vous pouvez changer la façon dont les nombres sont affichés, sans que leur représentation en mémoire ne soit modifiée (si vous voulez modifier le nombre en mémoire pour qu'il corresponde à l'affichage, reportez-vous à la fonction **[RND]** page 186). Le format et la ponctuation que vous spécifiez sont conservés en mémoire permanente.

Format des nombres

Il existe trois formats d'affichage des nombres, que vous choisissez à l'aide des fonctions **[FIX]**, **[SCI]** et **[ENV]**.

[FIX] n. Dans ce format, les nombres sont affichés avec n chiffres après la virgule ($0 \leq n \leq 9$). Si la partie entière d'un nombre comporte plus de $(10 - n)$ chiffres, l'affichage montrera moins de n chiffres après la virgule. Par exemple, le format par défaut est **[FIX] 4**, qui affiche quatre chiffres après la virgule ; mais si la partie entière d'un nombre comporte huit chiffres, votre HP-41 n'affichera que deux chiffres après la virgule.

Le dernier chiffre affiché est arrondi si le premier chiffre non affiché est supérieur ou égal à 5. Si un nombre comporte moins de n chiffres après la virgule, des zéros supplémentaires sont affichés après les derniers chiffres. Si un nombre est trop grand ou trop petit pour l'affichage, le format passe automatiquement et temporairement à **[SCI] n**.

[90] n. Dans ce format, les nombres sont affichés avec un chiffre avant et n chiffres après la virgule ($0 \leq n \leq 9$), et multipliés par une puissance de 10. Pour $n \leq 7$, le nombre est arrondi à n chiffres après la virgule. Comme un maximum de 7 chiffres après la virgule peut être affiché, l'arrondi a lieu en dehors de l'affichage lorsque vous spécifiez **[8] 8** ou **[9] 9** (ces formats peuvent être utiles lorsque les nombres sont imprimés).

[n] n. Dans ce format, un nombre affiché comporte les mêmes *chiffres* que dans le format **[n] n**, mais avec un exposant qui est toujours un multiple de trois. La virgule décimale se déplace vers la droite pour compenser les changements d'exposant.

Ponctuation

Les indicateurs binaires 28 et 29 permettent de contrôler l'utilisation des virgules et des points dans l'affichage. Aux Etats-Unis, le point sert de séparation entre la partie entière d'un nombre et ses décimales (c'est le point décimal), et la virgule sert à séparer les groupes de chiffres dans un nombre long. Dans d'autres pays, en Europe en particulier, la virgule sert de séparateur décimal et le point ponctue les groupes de trois chiffres.

L'indicateur binaire 28 détermine le rôle des points et des virgules. Par défaut, il est *armé*, ce qui donne un affichage conforme à la norme américaine (point décimal et virgule pour séparer les groupes de chiffres). En désarmant cet indicateur, vous inversez le rôle des points et des virgules, qui correspondent alors à l'usage européen.

L'indicateur binaire 29 détermine si un séparateur de groupe de chiffres est affiché ou non, indépendamment du type de ce séparateur (virgule ou point). Par défaut, l'indicateur 29 est *armé*, ce qui correspond à l'affichage du séparateur. En désarmant l'indicateur 29, vous supprimez tous les séparateurs de groupes de chiffres, et dans le cas particulier du format **[0] 0**, vous supprimez l'affichage du séparateur décimal.

Messages et affichages standard

Votre HP-41 affiche soit l'affichage standard, soit un message. L'affichage standard est constitué par le contenu du registre X, sauf si :

- le clavier alphabétique est validé (sans que vous soyez en train d'introduire un paramètre), auquel cas c'est le contenu du registre alphabétique qui est l'affichage standard ;
- votre HP-41 est en mode programme, auquel cas c'est la ligne de programme en cours qui est l'affichage standard ;
- un programme est en cours d'exécution, auquel cas c'est l'indicateur d'exécution de programme (P) qui est l'affichage standard.

Tout autre affichage est un message. Le chapitre 14 comprend des exemples relatifs à l'éditeur de texte, le chapitre 15 des exemples relatifs à l'horloge, le chapitre 17 des exemples relatifs au chronomètre, et le chapitre 21 des exemples de messages programmés destinés à l'utilisateur. Les exemples du présent chapitre présentent les messages apparaissant lors de la spécification des paramètres, de la prévisualisation des fonctions, de l'exécution des catalogues, et les messages d'erreur. Lorsque l'affichage contient un message, l'indicateur binaire 50 est armé.

Défilement de l'affichage

Pour que vous puissiez visualiser plus de caractères que l'affichage ne peut en contenir, votre HP-41 fait « défiler » les caractères sur l'affichage jusqu'à ce que le dernier caractère apparaisse. Pendant que les caractères défilent, vous pouvez appuyer sur n'importe quelle touche pour mettre fin au défilement et visualiser directement l'affichage final. La fonction correspondant à la touche sur laquelle vous avez appuyé n'est pas exécutée.

Spécification de paramètres

Certaines fonctions doivent être complétées par des paramètres. Lorsque votre HP-41 affiche le nom de la fonction suivi d'un ou plusieurs indicateurs de saisie (—), vous devez introduire un paramètre.

- **Dans le cas d'un paramètre numérique** (adresse de registre, numéro d'indicateur binaire, label local numérique, numéro de ligne de programme, etc.), le nombre d'indicateurs de saisie représente le nombre de chiffres que vous devez introduire (il se peut que vous deviez introduire des zéros complémentaires en tête de nombre, par exemple 042 pour désigner la ligne 42).
- **Dans le cas d'un paramètre alphabétique** (nom de fonction ou label global), appuyez sur **ALPHA** pour valider le clavier alphanumérique, inscrivez ensuite le nom ou le label, et appuyez à nouveau sur **ALPHA** pour mettre fin à l'opération.

Spécification indirecte de paramètres

Pour la plupart des fonctions, les paramètres peuvent être spécifiés indirectement: au lieu d'introduire le paramètre lui-même en réponse à l'indicateur de saisie, vous pouvez introduire l'adresse d'un registre (le « registre indirect ») contenant ce paramètre. Cette possibilité est particulièrement utile lorsque la valeur du paramètre dépend de calculs effectués précédemment par un programme ou lorsqu'un sous-programme est exécuté à répétition pour accéder à des registres séquentiels. En outre, les adresses des registres $R_{(100)}$ à $R_{(318)}$ de la mémoire principale doivent être spécifiés indirectement.

Pour spécifier un paramètre indirectement :

1. Exécutez la fonction
2. En réponse à l'indicateur de saisie, appuyez sur **■** . L'affichage fait suivre le nom de la fonction de **IND _ _ .**
3. Spécifiez le registre indirect.

Les exemples suivants montrent comment fonctionne la spécification indirecte pour trois types de paramètres. Dans chaque exemple, R_{10} est le registre indirect contenant un paramètre de 5; dans le premier exemple, 5 est simplement un nombre, dans le second exemple, 05 est une adresse, et dans le troisième exemple, 05 est un label.

Exemple. Supposons que R_{10} contient 5. Si vous exécutez $\boxed{\text{TONE}} \text{ IND } 10$, le nombre se trouvant dans R_{10} devient le paramètre de la fonction $\boxed{\text{TONE}}$. Par conséquent, $\boxed{\text{TONE}} \text{ IND } 10$ est l'équivalent de $\boxed{\text{TONE}} \text{ 5}$ lorsque R_{10} contient le chiffre 5.

$\boxed{\text{TONE}} \text{ IND } 10$	}	
+		$\boxed{\text{TONE}} \text{ 5}$
$R_{10} \boxed{5}$		

Exemple. Supposons que R_{10} contient 5. Si vous exécutez $\boxed{\text{STO}} \text{ IND } 10$, l'adresse se trouvant dans le registre R_{10} devient le paramètre de la fonction STO. Par conséquent, $\boxed{\text{STO}} \text{ IND } 10$ est l'équivalent de $\boxed{\text{STO}} \text{ 05}$ lorsque R_{10} contient le chiffre 5.

$\boxed{\text{STO}} \text{ IND } 10$	}	
+		$\boxed{\text{STO}} \text{ 05}$
$R_{10} \boxed{5}$		

La spécification indirecte d'une adresse — appelée *adressage indirect* — est l'utilisation la plus courante de la spécification indirecte de paramètres, et la pratique la plus courante en matière d'adressage indirect consiste à accéder à une série de registres grâce à un sous-programme d'itération contenu dans un programme. Par exemple, une boucle contenant $\boxed{\text{RCL}} \text{ IND } 10$, $\boxed{1/x}$, $\boxed{\text{STO}} \text{ IND } 10$ remplacera le nombre contenu dans R_{05} par son inverse lorsque R_{10} contient 5 (comme ci-dessus). La boucle peut ensuite incrémenter d'une unité l'adresse se trouvant dans R_{10} , en la faisant passer de 5 à 6, puis recommencer l'opération en remplaçant cette fois le nombre se trouvant dans R_{06} par son inverse, puis incrémenter à nouveau d'une unité l'adresse se trouvant dans R_{10} et ainsi de suite (les boucles sont décrites au chapitre 20, « Branchements »).

Exemple. Supposons que R_{10} contient 5. Si vous exécutez $\boxed{\text{XEQ}} \text{ IND } 10$, le label se trouvant dans R_{10} devient le paramètre de la fonction $\boxed{\text{XEQ}}$. Par conséquent, $\boxed{\text{XEQ}} \text{ IND } 10$ est l'équivalent de $\boxed{\text{XEQ}} \text{ 05}$ lorsque R_{10} contient le chiffre 5.

$\boxed{\text{XEQ}} \text{ IND } 10$	}	
+		$\boxed{\text{XEQ}} \text{ 05}$
$R_{10} \boxed{5}$		

Vous pouvez aussi spécifier indirectement n'importe quel label global figurant sur le catalogue 1 ou n'importe quelle fonction programmable ou label global figurant sur le catalogue 2, à condition que le label n'excède pas six caractères.

Vous pouvez spécifier indirectement les paramètres pour les fonctions suivantes :

- Fonctions dont les paramètres sont des adresses de registres.

STO, **RCL**

STO **+**, **STO** **-**, **STO** **x**, **STO** **÷**

←STO, **←MCL**

IRG, **DSE**

X <, **VIEW**, **ΣREG**

- **XEQ**, **GTU**

- **SP**, **CP**, **FT?**, **FC?**, **FS?C**, **FC?C**

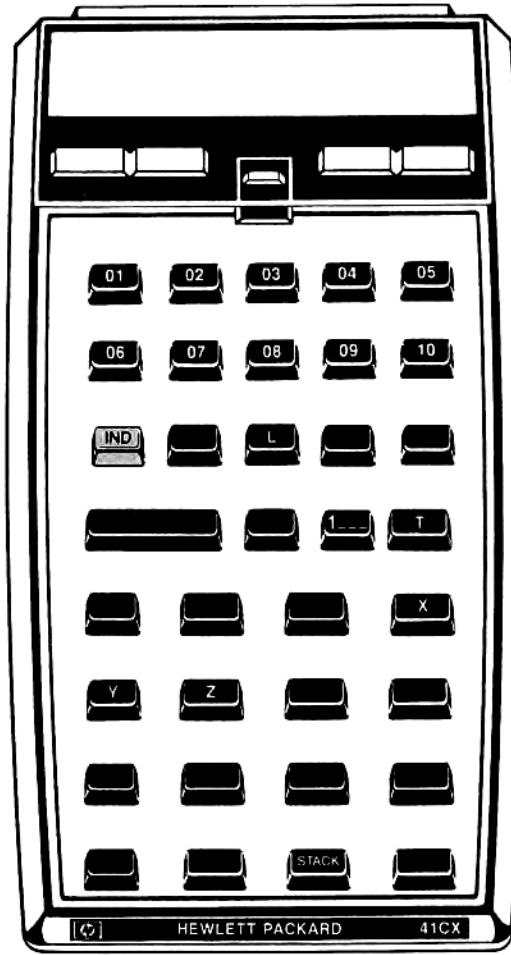
- **FIX**, **SCI**, **ENG**

- **TONE**

Touches spéciales

La figure ci-après montre les touches qui ont une signification particulière lorsque vous spécifiez un paramètre destiné à des fonctions du catalogue 3.

Touches spéciales permettant de spécifier les paramètres



Adresses des registres de la pile opérationnelle. Pour spécifier un registre de la pile opérationnelle ou le registre LAST X, appuyez sur \square , puis sur \square , \square , \square , \square , ou \square .

Numéros de lignes de programme. Pour spécifier des numéros de lignes supérieurs à 999, appuyez sur \square . Le HP-41 affiche alors 1 _____. Introduisez ensuite les trois chiffres requis.

Spécification d'un paramètre par pression sur une seule touche. Pour faciliter l'introduction des paramètres, vous pouvez spécifier un paramètre à un chiffre (0 à 9), à deux chiffres ou à trois chiffres (touches 1 à 10) en appuyant sur la (ou les) touche(s) appropriées dans les deux rangées supérieures du clavier. Par exemple, lorsqu'un, deux ou trois indicateurs de saisie sont affichés et que vous appuyez sur la touche \square , vous introduisez le paramètre 1, 01 ou 001 selon le nombre d'indicateurs. Si un seul indicateur de saisie est affiché et que vous appuyez sur \square , vous introduisez le paramètre 0; mais si deux ou trois indicateurs de saisie sont affichés, vous introduisez le paramètre 10 ou 010.

Comment redéfinir le clavier personnel

Deux fonctions permettent d'affecter labels globaux et fonctions aux touches du clavier personnel : \square (*assign* = affecter à) et \square (*programmable assign* = affecter par programme). Utilisez \square lorsque vous affectez des fonctions manuellement; cette fonction est plus facile à utiliser mais n'est pas programmable. Utilisez \square pour effectuer les affectations sous commande d'un programme.

Pour affecter une fonction manuellement :

1. Exécutez \square .
2. Appuyez sur \square , introduisez le nom de la fonction ou le label global au clavier, et appuyez à nouveau sur \square .
3. Appuyez sur la touche (ou sur \square et la touche) à redéfinir.

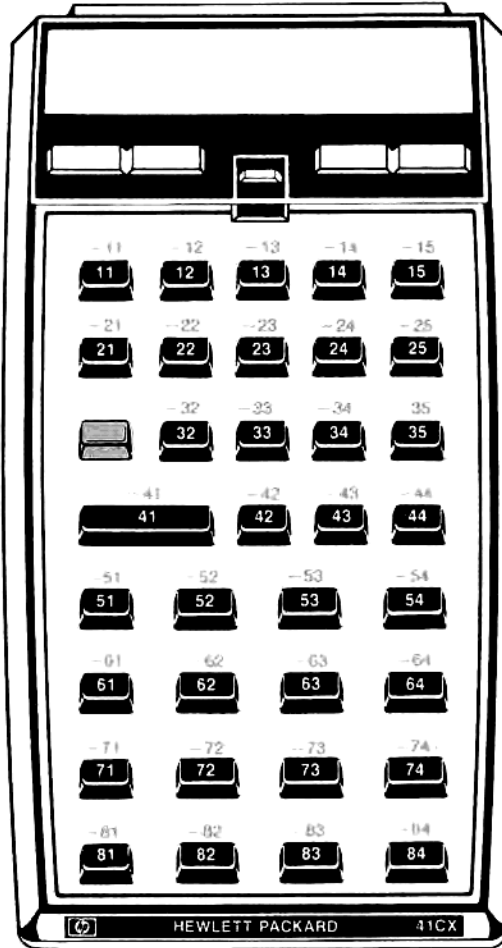
Pour affecter une fonction sous commande d'un programme :

1. Introduisez le nom de la fonction ou le label global dans le registre alphabétique.
2. Introduisez le code numérique de la touche à redéfinir (en fonction de la figure de la page ci-après) dans le registre X.
3. Exécutez \square .

La figure ci-après montre les codes numériques des touches du clavier personnel. Vous remarquerez que :

- Tous les codes ont deux chiffres.
- Les codes des fonctions secondaires sont négatifs.
- Vous ne pouvez redéfinir ni les touches fonctionnant en bascule, ni la touche jaune donnant accès aux fonctions secondaires.
- Vous pouvez redéfinir la touche \square . La nouvelle fonction que vous affectez à cette touche annule et remplace la fonction « Run » (marche) en mode exécution et la fonction \square en mode programme, mais vous pouvez toujours arrêter un programme en cours en appuyant sur \square .

Le clavier personnel



Lorsque vous affectez une fonction faisant partie du catalogue 2 ou 3, ou un label global faisant partie du catalogue 2, l'affectation est stockée dans la mémoire du clavier personnel (cette mémoire est comprise dans la mémoire principale et est décrite au chapitre 12); Cependant, lorsque vous affectez un label global faisant partie du catalogue 1, cette affectation est stockée comme une partie du label lui-même. Si le label est supprimé de la mémoire de programmation, l'affectation est annulée. Si le programme contenant le label ainsi affecté est stocké en mémoire annexe (module d'extension mémoire), et si le clavier personnel est validé (indicateur 27 armé) lorsque le programme est rappelé, l'affectation faisant partie du label se trouve re-validée.

Le catalogue du clavier personnel

En exécutant **[CATALOG]** 6, vous affichez toutes les fonctions et tous les labels globaux affectés au clavier personnel. Ces fonctions sont classées par ordre croissant des codes numériques des touches correspondantes, les codes négatifs apparaissant après les codes positifs. Vous pouvez interrompre le défilement du catalogue en appuyant sur **[RS]**; vous pouvez ensuite:

- Appuyer sur **[SST]** pour voir l'affectation suivante.
- Appuyer sur **[BSY]** pour voir l'affectation précédente.
- Appuyer sur **[C]** pour effacer l'affectation affichée.
- Appuyer sur **[RS]** pour que le défilement reprenne.

S'il n'y a pas d'affectation, ou si la seule affectation est effacée par pression sur **[C]**, le HP-41 affiche **CAT EMPTY** (catalogue vide).

Retour aux fonctions normales

Vous pouvez utiliser l'une des méthodes suivantes pour annuler la fonction affectée à une touche:

- Exécutez **[ASH]**, appuyez sur **[ALPHA]** deux fois de suite, puis sur la touche voulue.

ou

- Effacez le registre alphabétique, introduisez le code numérique voulu dans le registre X, puis exécutez **[PASN]**.

ou

- Appuyez sur **[C]** pendant que le catalogue 6 est interrompu et affiche l'affectation de la fonction.

Pour annuler toutes les affectations en cours, exécutez **[CLKEYS]**. Vous remarquerez que les affectations stockées avec les labels globaux en mémoire annexe seront revalidées si le clavier personnel est opérant (indicateur binaire 27 armé) lorsque vous rappelez le programme.

Les deux rangées supérieures de touches

Il existe un type particulier de label de programme, le label local alphabétique, qui est introduit à l'aide des deux rangées supérieures de touches du clavier personnel. Le nom de chaque label est un caractère alphabétique se trouvant sur l'une des deux rangées supérieures de touches : A à E sur la première rangée, F à J sur la seconde, et a à e en position secondaire (avec la touche jaune) sur la première rangée. Le chapitre 20 « Branchements » indique comment programmer à l'aide de ces labels; vous ne trouverez ci-après que les conditions requises pour exécuter un label local alphabétique sur le clavier personnel. Ces conditions sont les suivantes :

- Le clavier personnel est validé.
- Le programme en cours d'exécution doit contenir le label local alphabétique.
- Vous n'avez pas redéfini la touche qui correspond au label local alphabétique.


Combinées avec les règles générales d'utilisation du clavier personnel, ces conditions entraînent les priorités ci-dessous. Lorsque vous appuyez sur une touche de l'une des deux rangées supérieures du clavier personnel :

1. Si vous avez affecté une fonction ou un label global à cette touche, la fonction est exécutée ou le programme commence à s'exécuter à partir du label global.
2. Si vous n'avez pas redéfini la touche et si le label local alphabétique existe dans le programme en cours, l'exécution commence à partir de ce label local alphabétique.
3. Si aucune des deux conditions ci-dessus n'est vérifiée, c'est la fonction du clavier normal (inscrite sur la touche ou au-dessus) qui est exécutée.

L'exécution d'une fonction du clavier normal peut nécessiter plus de temps lorsque le clavier personnel est validé, du fait que votre HP-41 vérifie d'abord quelles sont les priorités. Pour éviter cet inconvénient, vous pouvez invalider le clavier personnel avant d'appuyer sur la touche, ou affecter la fonction du clavier normal à cette touche.

Prévisualisation et annulation des fonctions

Vous pouvez visualiser, sur l'affichage, la fonction d'une touche, sans pour autant l'exécuter, en maintenant cette touche enfoncée. Cette possibilité est particulièrement utile sur le clavier personnel, lorsque vous ne vous rappelez plus quelles touches vous avez redéfinies.

- Si la fonction nécessite un paramètre (apparition d'un ou plusieurs indicateurs de saisie), relâchez la touche. Si vous désirez annuler la fonction, appuyez sur .
- Si la fonction ne nécessite pas l'introduction d'un paramètre, vous pouvez soit relâcher la touche pour exécuter la fonction, soit maintenir la pression sur la touche jusqu'à ce que la fonction soit annulée : le message **NULL** (annulée) apparaît.

En outre, il y a prévisualisation d'une ligne de programme dans quatre cas (lorsque vous relâchez la touche avant que le message **NULL** ne soit affiché) :

- Si le clavier personnel est validé et si vous appuyez sur une touche à laquelle vous avez affecté un label global, ce label est affiché et l'exécution du programme commence à partir de ce label.
- Si le clavier personnel est validé et si vous appuyez sur une touche qui correspond à un label alphabétique local dans le programme en cours, le message **XEQ label** apparaît et l'exécution du programme commence à partir de ce label.
- Si vous appuyez sur **[R/S]**, la ligne de programme en cours est affichée et l'exécution du programme commence à cette ligne.
- Si vous appuyez sur **[SST]**, la ligne de programme en cours est affichée et elle seule est exécutée.

Les catalogues

Six catalogues vous permettent de visualiser le contenu de la mémoire. La fonction **[CATALOG]** n'est pas programmable, mais il existe des fonctions programmables équivalentes aux catalogues 4 et 5. Vous trouverez ci-après les règles de fonctionnement communes à tous les catalogues, puis une présentation générale de chaque catalogue.

Principes d'utilisation des catalogues

Exécutez **[CATALOG] n** pour lancer le défilement du catalogue *n*.

Pendant que le catalogue défile :

- Une pression sur toute autre touche que **[R/S]** et **[ON]** accélère le défilement.
- Une pression sur **[R/S]** interrompt le défilement.

Pendant que le catalogue est arrêté :

- Une pression sur **[SST]** affiche la rubrique suivante du catalogue.
- Une pression sur **[BFT]** affiche la rubrique précédente du catalogue.
- Une pression sur **[R/S]** relance le défilement.
- Une pression sur **[◀]** permet de sortir du catalogue.

Les catalogues 4, 5 et 6 consomment autant d'énergie électrique qu'un programme en cours d'exécution, même lorsque le défilement du catalogue est interrompu. C'est pourquoi le HP-41 invalide les fonctions de catalogue lorsqu'elles sont interrompues pendant plus de deux minutes environ.

Une imprimante doit être en mode Trace pour imprimer un catalogue.

Différents types de catalogues

Catalogue 1 : Programmes de l'utilisateur. Liste de tous les labels globaux et instructions **[END]**. Chaque instruction **[END]** s'accompagne du nombre de multipliants du programme, et l'instruction **.END.** permanente (qui apparaît la dernière) est accompagnée du nombre de registres encore disponibles pour de nouveaux programmes.

Vous pouvez vous servir du catalogue 1 pour faire de n'importe quel programme le programme en cours : appuyez sur **[RS]** pour interrompre le défilement sur l'instruction **[END]** ou le label global voulu, puis appuyez sur **[←]** pour invalider la fonction de catalogue (page 284).

Catalogue 2 : Fonctions auxiliaires. Liste de toutes les fonctions et programmes disponibles sur votre HP-41 à partir des périphériques connectés et des modules enfichés au moment considéré, plus toutes les fonctions d'horloge et mémoires annexes. Un ' précède les labels globaux sur l'affichage, pour les distinguer des fonctions.

Les fonctions et les programmes sont groupés par sources. Initialement, le catalogue n'affiche que les entrées principales (titres) de chaque groupe. Pour obtenir l'affichage de toutes les entrées, appuyez sur **[RS]** pour interrompre le défilement du catalogue, attendez que l'affichage clignote, puis appuyez sur **[ENTER ↑]**. Pour revenir à l'affichage des seuls titres, appuyez sur **[RS]** pour interrompre le catalogue, puis sur **[ENTER ↑]** (page 394).

Catalogue 3 : Fonctions standard. Liste alphabétique des fonctions standard du HP-41. Ce catalogue contient les noms alphabétiques de chaque fonction, qui peuvent parfois être différents des noms figurant sur le clavier. Vous devez connaître le nom alphabétique d'une fonction si vous voulez l'affecter à une touche du clavier personnel ou interpréter correctement les lignes de programme.

Catalogue 4 : Répertoire de la mémoire annexe. Liste de tous les fichiers se trouvant en mémoire annexe (modules d'extension mémoire), avec indication du nom, du type et du nombre de registres de chaque fichier. Après avoir énuméré tous les fichiers existants, votre HP-41 affiche le nombre de registres encore disponibles pour de nouveaux fichiers. La fonction **[EMMR]** permet à ce catalogue d'être exécuté par programme.

Vous pouvez vous servir du catalogue 4 pour faire de n'importe quel fichier le fichier en cours : appuyez sur **[RS]** pour interrompre le catalogue sur le fichier voulu, puis appuyez sur **[←]** pour invalider le catalogue (page 206).

Catalogue 5 : Catalogue des alarmes. Liste de toutes les alarmes en mémoire, avec indication de l'heure, de la date, et du message pour chaque alarme. Vous pouvez supprimer des alarmes, ré-armer des alarmes à répétition, et examiner des parties spécifiques d'une alarme en utilisant le clavier du catalogue d'alarmes. La fonction **[ALMGAT]** permet à ce catalogue d'être exécuté par programme (page 255).

Catalogue 6 : Catalogue du clavier personnel. Liste de toutes les fonctions et labels globaux affectés aux touches du clavier personnel, avec affichage du nom de la fonction ou du label global, ainsi que du code numérique de la touche correspondante, à partir du code 11 (**[Σ+]**) et jusqu'au code -84 (**[VIEW]**).

Vous pouvez utiliser le catalogue 6 pour annuler n'importe quelle affectation : appuyez sur **[RS]** pour interrompre le catalogue sur l'affectation voulue, puis appuyez sur **[■] [C]** (page 168).

Messages d'erreur

Une opération interdite n'est jamais exécutée. S'il s'agit d'une instruction de programme, le HP-41 arrête l'exécution du programme et affiche un message d'erreur*.

- Pour effacer le message d'erreur de l'affichage, appuyez sur \square .
- Pour exécuter une autre fonction, il vous suffit d'appuyer sur la touche correspondante — inutile de commencer par effacer le message d'erreur.
- Pour trouver quelle instruction est à l'origine de l'erreur, appuyez sur $\boxed{\text{PRGM}}$ pour passer en mode programme. L'affichage montre alors la ligne contenant l'instruction erronée (ou un numéro XROM dans le cas où le message **NONEXISTENT** provient du fait qu'un module n'ait pas été enfiché).

Vous trouverez en Annexe A une liste des messages d'erreur et d'état. De nombreux modules qui s'enfichent dans le HP-41 possèdent leurs propres messages, qui apparaissent sur l'affichage du HP-41. Reportez-vous à la documentation fournie avec ces modules.

*Les indicateurs binaires 24 et 25 peuvent empêcher certaines erreurs prévues d'arrêter l'exécution d'un programme. Ces indicateurs binaires sont décrits page 290.

La pile opérationnelle

Table des matières

Introduction	174
Calculs en notation polonaise inverse	175
Montée et descente des nombres dans la pile	175
Utilisation de ENTER ↑	175
Autorisation/inhibition des mouvements de montée	176
Ordre de la saisie	176
Remplissage de la pile opérationnelle	177
Le registre LAST X	179
Correction d'erreurs	179
Opérations arithmétiques avec constantes	180
Autres opérations dans la pile opérationnelle	180
Permutation du contenu de deux registres	181
Permutation circulaire de l'ensemble de la pile	181
Stockage et rappel	182
Arithmétique directe dans les registres	182
Effacement de la pile	183

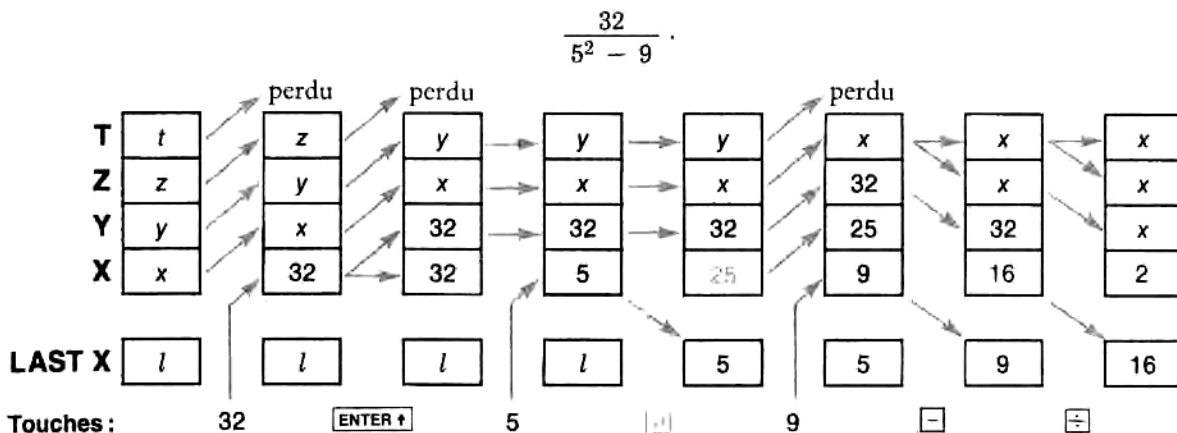
Introduction

Les fonctions numériques utilisent quatre registres qui constituent la *pile opérationnelle*. Les nombres se déplacent *automatiquement* d'un registre à l'autre dans la pile opérationnelle à mesure que vous les introduisez et effectuez des calculs. La logique utilisée est la notation polonaise inverse qui diminue le nombre de touches à actionner et permet l'affichage de tous les résultats intermédiaires. Si vous ne connaissez pas la notation polonaise inverse, reportez-vous à la page 20.

- Le premier paragraphe ci-après, « Calculs en notation polonaise inverse » prend pour exemple le calcul d'une expression numérique type et décrit les principes régissant l'utilisation de la pile opérationnelle. Vous y trouverez également une méthode pour les calculs avec constante (remplissage de la pile avec la constante).
- Le second paragraphe, « Registre LAST X », traite d'un registre particulier, qui est en liaison étroite avec les registres de la pile. Le registre LAST X sert à corriger les erreurs et offre une seconde méthode de calcul avec des constantes.
- Le troisième paragraphe décrit les autres opérations possibles dans la pile, qui vous permettent une plus grande souplesse d'utilisation; là aussi, l'accent est mis sur le recours répété à une constante.

Calculs en notation polonaise inverse

Le schéma ci-dessous montre le contenu de la pile opérationnelle et du registre LAST X à chaque étape d'un calcul effectué selon la logique de la notation polonaise inverse. x , y , z , t , et l représentent les nombres se trouvant initialement dans la pile. L'expression à calculer est



Toutes les explications sur la façon dont le pile fonctionne et sur la façon de l'utiliser efficacement se référeront à l'exemple ci-dessus.

Montée et descente des nombres dans la pile

Les mouvements automatiques du contenu des registres sont des montées (déplacement vers le haut dans le schéma) ou des descentes (déplacements vers le bas dans le schéma).

Montée. Elle se produit généralement lorsqu'un nombre est introduit dans le registre X. Les nombres se trouvant dans les registres Y et Z se trouvent alors décalés dans les registres Z et T; le nombre se trouvant dans le registre T est perdu. Dans l'exemple, la montée se produit lorsque vous introduisez 32, lorsque vous appuyez sur \square (ce qui copie 32 dans le registre Y), et lorsque vous introduisez 9.

Descente. Elle se produit généralement lorsqu'une fonction combine les nombres des registres X et Y. Les nombres se trouvant dans les registres Z et T descendent alors dans les registres Y et Z; le nombre se trouvant dans le registre LAST X est perdu. Dans l'exemple, la descente se produit lorsque vous exécutez \square et \square .

Utilisation de \square

En appuyant sur \square , vous séparez deux nombres introduits l'un après l'autre (32 et 5 dans l'exemple). Le nombre se trouvant dans le registre X (32) est alors copié dans le registre Y. La copie restant dans le registre X est remplacée par le nombre qui est ensuite introduit (5), car l'action sur \square inhibe les mouvements de montée dans la pile.

Autorisation/inhibition des mouvements de montée

Presque toutes les fonctions autorisent des montées dans la pile ; il se produit une montée chaque fois que vous placez un nombre dans le registre X après avoir exécuté une fonction autorisant une montée. Cependant, il existe quatre fonctions qui inhibent les montées, et d'autres qui sont neutres.

Fonctions inhibant les montées dans la pile. Ces quatre fonctions sont **ENTER** ↑, **OL** □, **Σ+** □, et **Σ-** □ . Si vous exécutez l'une de ces fonctions, puis introduisez un nombre dans le registre X, ce nombre remplace le contenu du registre X et les registres Y, Z, et T ne sont affectés. Dans le schéma de la page précédente, le registre X est représenté en bleu lorsque les montées sont inhibées et que son contenu sera remplacé.

Fonctions neutres. Les fonctions suivantes n'ont pour résultat ni d'autoriser ni d'inhiber les montées dans la pile, mais de maintenir l'état antérieur :

- Les touches fonctionnant en bascule (**ON** □, **USER** □, **PRGM** □, **ALPHA** □).
- La touche d'espace arrière (**←** □) pendant la saisie de chiffres ou de caractères.
- La touche préfixe donnant accès aux fonctions secondaires (**■** □).
- Les catalogues 1, 2, 3 et 6.

Ordre de la saisie

Deux considérations principales décident de l'ordre dans lequel vous devez introduire les opérandes. Vous aurez beaucoup moins de touches à actionner si vous respectez les règles ci-après, même si vous devez parfois faire un choix parmi ces règles.

Parenthèses imbriquées. Lorsque des expressions numériques comprennent des parenthèses imbriquées, calculez d'abord la partie comprise dans les parenthèses les plus « intérieures », puis utilisez ce résultat dans l'expression ainsi simplifiée. Si vous devez calculer deux expressions entre parenthèses, puis les combiner, la pile opérationnelle sauvegarde le résultat de la première expression pendant que vous calculez la seconde. L'exemple indiqué ci-dessous au paragraphe « Expressions polynomiales » illustre cette règle.

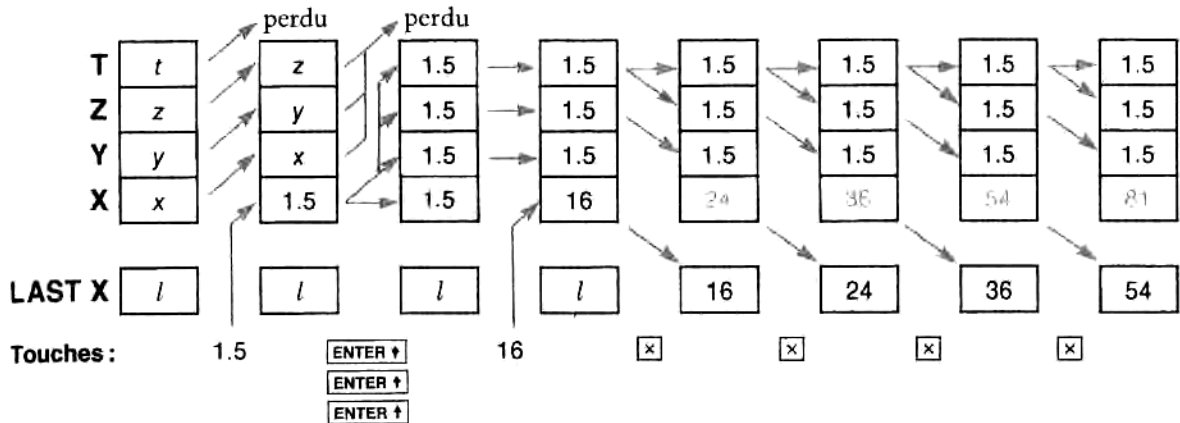
Fonctions non commutatives. Les fonctions comme la soustraction et la division ne sont pas commutatives et l'ordre des opérandes est donc essentiel : $5 - 3 \neq 3 - 5$, et $5 \div 3 \neq 3 \div 5$. Pour les expressions comprenant des fonctions non commutatives, introduisez ou calculez le nombre qui doit être dans le registre Y avant d'introduire ou de calculer le nombre qui doit être dans le registre X. L'exemple précédent illustre cette règle à deux reprises :

- Le numérateur (32) est introduit avant que le dénominateur ($5^2 - 9$) ne soit calculé.
- Le terme 5^2 est calculé avant que 9 ne lui soit soustrait.

Remplissage de la pile opérationnelle

Vous remarquerez que, dans les trois dernières étapes de l'exemple précédent, x se déplace du registre T dans les registres Y et Z. Cette conséquence de la descente d'un paramètre dans la pile peut permettre de maintenir une constante dans le registre Y, comme le montrent les deux exemples ci-après. Cette méthode est particulièrement appropriée lorsque la constante doit se trouver dans le registre Y pour les opérations non commutatives comme \square et \square (alors que LAST X , elle, permet d'avoir une constante dans le registre X).

Croissance cumulée. Supposons que vous vouliez calculer la croissance d'une quantité commençant à la valeur 16 et augmentant de 50 % à chaque période. Commencez par remplir la pile avec le coefficient de croissance (1,5) et introduisez la valeur de départ (16) dans le registre X. Appuyez ensuite sur \square pour calculer la valeur après la première période, et à nouveau sur \square pour chaque nouvelle période.



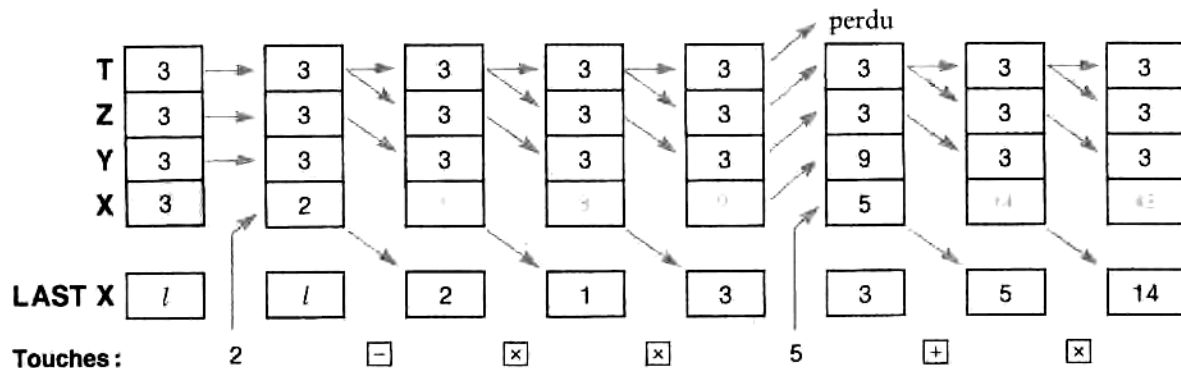
Expressions polynomiales. En remplissant la pile, vous pouvez calculer un polynôme, qui nécessite plusieurs copies de la variable. Pour une plus grande efficacité, utilisez la méthode de Horner pour réécrire le polynôme, avec des parenthèses imbriquées pour éliminer des exposants supérieurs à 1. Supposons que vous vouliez calculer

$$x^4 - 2x^3 + 5x$$

avec $x = 3$. Commencez par réécrire le polynôme de manière à éliminer les exposants.

$$\begin{aligned} x^4 - 2x^3 + 5x &= (x^3 - 2x^2 + 5)x \\ &= ((x^2 - 2x)x + 5)x \\ &= (((x - 2)x)x + 5)x \end{aligned}$$

Remplissez ensuite la pile avec la variable en appuyant sur 3, ENTER, ENTER, ENTER, et exécutez les opérations ci-dessous. Vous remarquerez que le calcul commence par la parenthèse la plus intérieure.



Une fois familiarisé avec la méthode de Horner, vous pouvez introduire les différents éléments d'un polynôme sans réécrire ce dernier. Par exemple, pour calculer le polynôme

$$ax^5 + bx^4 + cx^3 + dx^2 + ex + f$$

les étapes sont les suivantes, après que la pile ait été remplie avec la variable :

$$a, \boxed{x}, b, \boxed{+}, \boxed{x}, c, \boxed{+}, \boxed{x}, d, \boxed{+}, \boxed{x}, e, \boxed{+}, \boxed{x}, f, \boxed{+}.$$

- Vous remarquerez que, à l'exception du premier et du dernier, tous les coefficients sont suivis de $\boxed{+}$ et \boxed{x} (en effet, il n'y a pas de résultat précédent à ajouter au premier coefficient, et le dernier coefficient n'est pas multiplié par une puissance de la variable).
- Si le premier coefficient est 1, commencez directement au second coefficient (la variable se trouve déjà dans le registre X).
- Pour les coefficients négatifs, vous pouvez introduire une valeur positive et remplacer $\boxed{+}$ par $\boxed{-}$ immédiatement après cette valeur.
- Lorsque x n'est pas élevé à une puissance, appuyez simplement sur \boxed{x} (ce qui introduit en fait un coefficient 0 pour cette puissance).

Résultats non cumulés. Vous pouvez aussi utiliser la méthode des constantes pour effectuer une série d'opérations sans liaison entre elles (non cumulées) et faisant intervenir une constante. Après chaque calcul, appuyez sur $\boxed{=}$ pour effacer le registre X avant d'introduire l'opérande suivant. Cela a pour effet d'inhiber la montée des nombres dans la pile, empêchant ainsi que le résultat précédent ne chasse la constante du registre Y.

Le registre LAST X

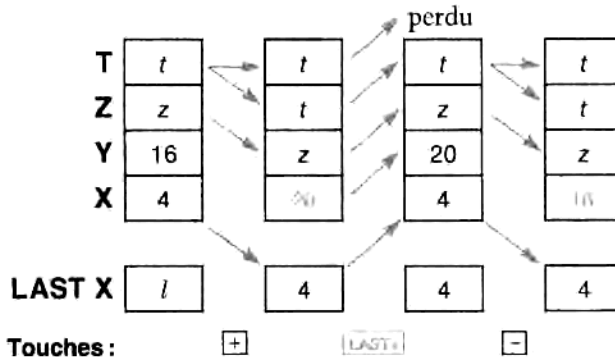
Le registre X contient l'opérande x de la fonction numérique précédente (sauf pour **CHS**). Pour rappeler ce nombre dans le registre X, appuyez sur **LAST X**. Cela vous permet de corriger une erreur et de rappeler un opérande pour un autre calcul.

Correction d'erreurs

Fonctions à un nombre (monadiques). Si vous exécutez par erreur une fonction à un nombre, vous pouvez corriger cette erreur de la façon suivante :

1. Appuyez sur **0**. Cela remplace le résultat erroné par un zéro et inhibe la montée des nombres dans la pile.
2. Appuyez sur **LAST X**. Cela rappelle votre opérande, qui remplace alors le zéro se trouvant dans le registre X.
3. Reprenez votre calcul avec la fonction voulue.

Fonctions à deux nombres (diadiques). Si vous faites une erreur avec une fonction comme **+** ou **-**, vous pouvez utiliser **LAST X** et la fonction inverse (**-** ou **x**) pour corriger votre erreur. Supposant que vous faites une erreur dans l'addition de deux nombres. Appuyez sur **LAST X** puis sur **-** comme indiqué ci-dessous. La nature de votre erreur détermine la façon dont vous devez continuer ; les différentes possibilités sont énumérées après le schéma ci-dessous.

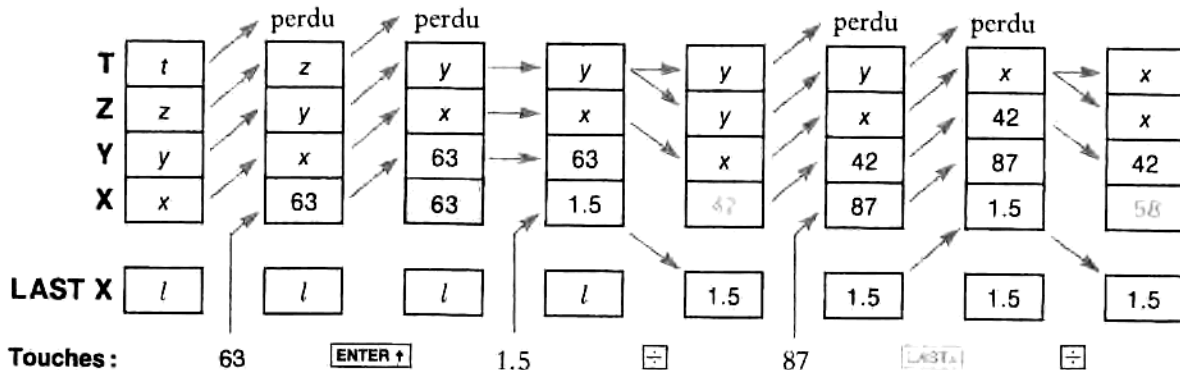


- Si vous vouliez multiplier au lieu d'ajouter, exécutez **LAST X** à nouveau pour ramener la pile à son état initial, puis exécutez votre multiplication.
- Si vous ne vouliez pas ajouter le nombre 16, appuyez sur **0** pour effacer 16, introduisez le nombre voulu, exécutez **LAST X** pour rappeler 4, puis appuyez sur **+**.
- Si c'est le nombre 4 qui est erroné dans l'addition, introduisez le nombre voulu, puis appuyez sur **+**.

Les erreurs avec d'autres types de fonctions à deux nombres sont encore plus faciles à corriger. Vous pouvez par exemple annuler une pression sur **P+R** en exécutant **R=P**, et corriger des erreurs avec **%** et **%CH** comme vous le feriez pour une fonction à un nombre. Pour corriger des erreurs dans d'autres fonctions, vous devez d'abord déterminer comment la fonction affecte la pile, puis inverser les déplacements.

Opérations arithmétiques avec constantes

L'exemple ci-dessous montre comment rappeler une constante en vue de calculs ultérieurs. Supposons que vous vouliez diviser 63 et 87 par 1,5. Cette constante doit être introduite en seconde position (après 63) de manière à se trouver dans le registre X pour le premier calcul, puis elle doit être conservée dans le registre LAST X.



Cette méthode est particulièrement appropriée lorsque la constante doit se trouver dans le registre X pour des opérations non commutatives comme \square et \ominus (au contraire, lorsque des opérations arithmétiques utilisent la descente des nombres dans la pile, la constante se trouve dans le registre Y).

Autres opérations dans la pile opérationnelle

Vous pouvez considérer les quatre registres de la pile comme deux paires de registres. Les registres X et Y sont le centre de presque toutes les activités, alors que les registres Z et T jouent le rôle de registres de stockage reliés aux registres plus actifs X et Y par les mouvements de montée et de descente. Si vous faites une copie supplémentaire d'un nombre pendant qu'il se trouve dans le registre X, ou si vous rappelez une copie du registre LAST X, vous pouvez stocker temporairement cette copie dans l'un des registres supérieurs de la pile, pour le rappeler ultérieurement.

Pour tirer pleinement partie des registres Z et T, prévoyez à l'avance les différentes étapes qui résulteront d'une série de calculs. Déterminez à quel endroit les opérandes doivent se trouver à chaque étape, partez du calcul final et remontez en arrière, et utilisez les opérations présentées ci-après pour relier les résultats d'un calcul aux entrées nécessaires pour le calcul suivant. Cette utilisation optimale de la pile économise la mémoire de programmation et réduit les besoins en registres de stockage.

Permutation du contenu de deux registres

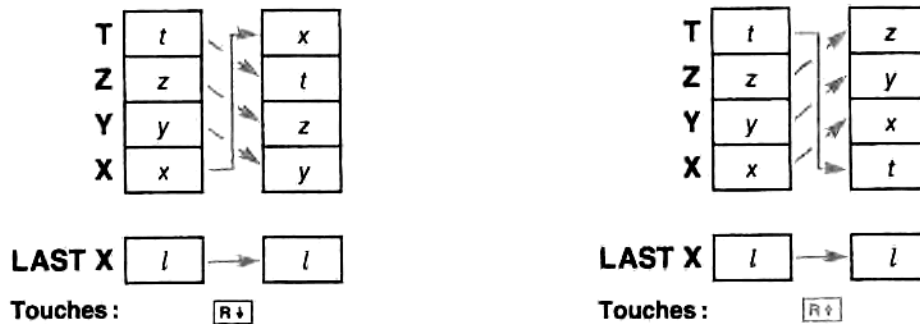
Permutation des registres X et Y. En exécutant $\boxed{x\leftrightarrow y}$, vous permutez le contenu des registres X et Y. Cette fonction a plusieurs objets ; elle permet

- d'examiner le contenu du registre Y. Appuyez sur $\boxed{x\leftrightarrow y}$, examinez l'affichage, puis appuyez à nouveau sur $\boxed{x\leftrightarrow y}$ pour remettre les nombres dans leur ordre initial. Ceci est utile lorsqu'une fonction envoie des résultats à la fois dans le registre X et le registre Y, comme les fonctions statistiques et les conversions de coordonnées polaires/rectangulaires par exemple ;
- de permuter des nombres qui ne sont pas dans le bon ordre pour des opérations non commutatives, comme la soustraction et la division ;
- de réorganiser le contenu de la pile en combinaison avec $\boxed{R\downarrow}$ ou $\boxed{R\uparrow}$; cf. le paragraphe « Permutation circulaire » ci-dessous.

Permutation du registre X et d'un autre registre de la pile. Pour permuter le contenu du registre X avec celui d'un autre registre de la pile ou avec celui du registre LAST X, exécutez $\boxed{x\leftrightarrow}$ et appuyez sur $\boxed{\cdot}$ puis sur \boxed{Y} , \boxed{Z} , \boxed{T} ou \boxed{L} . Reportez-vous au paragraphe « Arithmétique directe dans les registres » ci-après pour trouver un exemple de cette fonction.

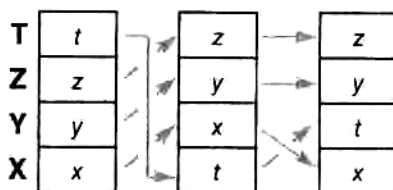
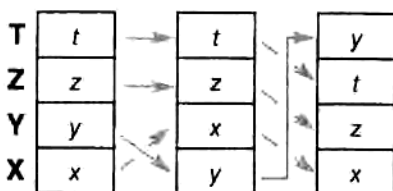
Permutation circulaire de l'ensemble de la pile

Les fonctions $\boxed{R\downarrow}$ (déplacement vers le bas) et $\boxed{R\uparrow}$ (déplacement vers le haut) réalisent une rotation du contenu de la pile dans un sens ou un autre, sans duplication ni perte de données.



Vous remarquerez que le registre LAST X n'est pas modifié. Pour revoir tous les nombres stockés dans la pile, appuyez soit quatre fois de suite sur $\boxed{R\downarrow}$ ou sur $\boxed{R\uparrow}$. Chaque nombre est affiché au moment où il passe dans le registre X, et la pile se retrouve dans son état initial.

Utilisez $\boxed{R\downarrow}$ et $\boxed{R\uparrow}$ en combinaison avec $\boxed{x\uparrow y}$ pour permuter des registres de la pile autres que le registre X. Ces fonctions vous permettent de réorganiser la pile dans n'importe quel ordre ; en voici deux exemples.



Touches :

$\boxed{x\uparrow y}$

$\boxed{R\downarrow}$



Touches :

$\boxed{R\uparrow}$

$\boxed{x\uparrow y}$

Stockage et rappel

Vous pouvez copier n'importe quel nombre dans la pile en exécutant \boxed{STO} ou \boxed{RCL} , puis en spécifiant un registre de la pile. Le résultat de ces deux fonctions est que le registre X et le registre spécifié contiennent le même nombre.

Stockage. Pour copier le contenu du registre X dans un registre de la pile ou dans le registre LAST X, appuyez sur \boxed{STO} \square , puis sur \boxed{Y} , \boxed{Z} , \boxed{T} ou \boxed{L} . Le nombre qui se trouvait dans le registre spécifié est perdu.

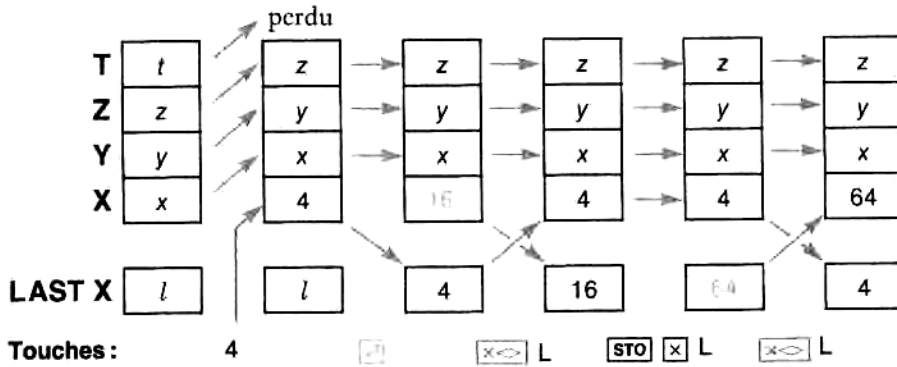
Rappel. Pour copier le contenu d'un registre de la pile ou du registre LAST X dans le registre X, appuyez sur \boxed{RCL} \square , puis sur \boxed{Y} , \boxed{Z} , \boxed{T} ou \boxed{L} . Le nombre se trouvant dans le registre T est perdu lorsque l'ensemble des contenus se décalent vers le haut (sauf si les mouvements de montée sont inhibés).

Arithmétique directe dans les registres

Vous pouvez combiner le contenu du registre X avec celui de n'importe quel registre de la pile en appuyant sur \boxed{STO} \square \square , \boxed{STO} \square \square , \boxed{STO} \square \square , ou \boxed{STO} \square \square , puis sur \boxed{X} , \boxed{Y} , \boxed{Z} , \boxed{T} ou \boxed{L} . N'oubliez pas que l'ordre des opérandes est essentiel pour la soustraction et la division ; l'opérande se trouvant dans le registre que vous spécifiez correspond ici à celui qui se trouve normalement dans le registre Y. L'arithmétique directe dans les registres diffère par plusieurs points de l'arithmétique normale dans la pile :

- Le résultat va dans le registre que vous spécifiez.
- Le registre X demeure inchangé (sauf si vous le spécifiez comme étant le paramètre).
- Le registre LAST X demeure inchangé (sauf si vous le spécifiez comme étant le paramètre).
- Il ne se produit aucun mouvement de descente dans la pile.

Le programme ci-dessous élève au cube le nombre se trouvant dans le registre X et place la valeur initiale de ce dernier dans le registre LAST X sans modifier le contenu des autres registres de la pile.



Effacement de la pile

Pour placer des zéros dans les registres X, Y, Z et T, exécutez **CLST** (clear stack = effacement pile). Le registre LAST X demeure inchangé.

Fonctions numériques

Table des matières

Introduction	184
Fonctions à un nombre (ou fonctions monadiques)	185
Fonctions générales	185
Fonctions modifiant des nombres	186
Opérations trigonométriques	186
Conversions	187
Fonctions logarithmiques et exponentielles	187
Fonctions à deux nombres (ou fonctions diadiques)	187
Opérations arithmétiques de base	188
Opérations arithmétiques sur les temps	188
Pourcentages	188
Conversions polaires/rectangulaires	189
Autres fonctions à deux nombres	189
Statistiques	190
Registres statistiques	190
Saisie des données	191
Moyenne	192
Ecart-type	192

Introduction

Le présent chapitre décrit les fonctions numériques de votre HP-41. Toutes les fonctions à un ou deux nombres s'exécutent dans la pile opérationnelle; leur fonctionnement est expliqué sur des schémas représentant la pile. Bien que les données destinées aux fonctions statistiques soient introduites à partir de la pile, elles sont préalablement accumulées dans les registres statistiques de la mémoire principale. Les résultats des opérations sur ces données accumulées reviennent ensuite dans la pile. Certaines autres fonctions, impliquant des calculs mais ne renvoyant pas nécessairement les résultats dans la pile (par exemple l'arithmétique directe dans les registres et la fonction \overline{ISG}), ne figurent pas dans le présent chapitre.

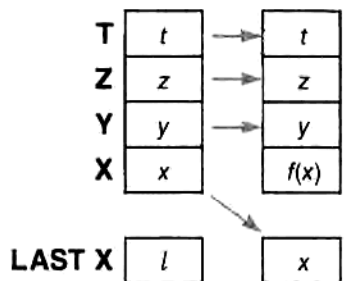
Les fonctions numériques peuvent générer trois situations d'erreur :

1. Si vous essayez un calcul avec un opérande interdit pour la fonction considérée (par exemple une division lorsque $x = 0$), le message **DATA ERROR** (erreur de donnée) apparaît.
2. Si vous essayez un calcul avec un opérande qui n'est pas un nombre, le message **ALPHA DATA** (donnée alphanumérique) apparaît. N'oubliez pas qu'une chaîne de chiffres issus du registre alphabétique n'est pas considérée comme un nombre.
3. Si vous essayez un calcul ayant pour résultat un nombre supérieur à $9,99999999 \times 19$, le message **OUT OF RANGE** (hors limites) apparaît (les sommations statistiques $\Sigma+$ et $\Sigma-$ font exception à cette règle).

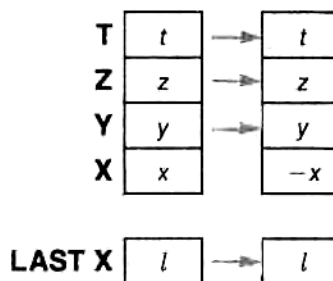
Votre HP-41 n'exécute pas une fonction lorsqu'elle génère une situation d'erreur. Sauf si l'indicateur binaire 25 est armé, l'apparition du message d'erreur **DATA ERROR** ou **ALPHA DATA** s'accompagne d'une interruption de programme (si un programme était en cours) ; et l'apparition du message d'erreur **OUT OF RANGE** interrompt également l'exécution d'un programme, sauf si l'indicateur binaire 24 ou 25 est armé.

Fonctions à un nombre (ou fonctions « monadiques »)

Les fonctions à un nombre remplacent l'opérande se trouvant dans le registre X par le résultat, sauvegardent l'opérande dans le registre LAST X, et laissent les registres Y, Z et T inchangés. Dans le schéma de la pile ci-dessous, $f(x)$ représente le résultat. La seule exception à cette règle est la fonction **CHS** (*change sign* = changement de signe), représentée à droite, qui ne sauvegarde pas l'opérande.



Fonctions à un nombre



CHS

Fonctions générales

Inverse. En exécutant $\frac{1}{x}$, vous obtenez l'inverse de x .

Carré et racine carrée. En exécutant \square , vous obtenez le carré de x , et en exécutant \sqrt{x} , vous obtenez la racine carrée de x (si x est positif).

Factorielle. n étant un nombre entier positif, en exécutant **FACT** n vous obtenez $n! = n (n - 1) (n - 2) \dots 1$.

Fonctions modifiant des nombres

Valeur absolue et signe d'un nombre. En exécutant **ABS** vous obtenez $|x|$. En exécutant **SIGN** vous obtenez :

$$\begin{aligned} & 1 \text{ si } x \geq 0, \\ & -1 \text{ si } x < 0, \text{ ou} \\ & 0 \text{ si le registre X contient des données alphabétiques.} \end{aligned}$$

Partie entière et partie décimale. Ces fonctions réduisent un nombre à sa partie entière ou à sa partie décimale. Par exemple, si le registre X contient 777,888, vous obtenez 777 en exécutant **INT** et 0,888 en exécutant **FRC**.

Arrondi. N'oubliez pas que les fonctions commandant le format d'affichage d'un nombre affectent uniquement la façon dont il apparaît sur l'affichage, et non pas la façon dont il est représenté en mémoire. Pour arrondir en mémoire un nombre se trouvant dans le registre X :

1. Choisissez un format d'affichage ayant le nombre de chiffres décimaux que vous voulez attribuer au nombre arrondi.
2. Exécutez **RND**.

Par exemple, pour arrondir un nombre à l'entier le plus proche, exécutez **FIX** 0 puis **RND**.

Opérations trigonométriques

Modes angulaires. Le mode angulaire détermine la façon dont l'ordinateur interprète les nombres en tant qu'angles. Le mode angulaire que vous choisissez est sauvegardé en mémoire permanente. Ces fonctions modifient seulement l'unité dans laquelle les angles sont exprimés, et non pas les nombres stockés en mémoire.

- Exécutez **RAD** pour obtenir des radians. Le témoin **RAD** apparaît, indiquant que les nombres seront interprétés comme des angles exprimés en radians (il y a 2π radians dans un cercle).
- Exécutez **GRAD** pour obtenir des grades. Le témoin **GRAD** apparaît, indiquant que les nombres seront interprétés comme des angles exprimés en grades (il y a 400 grades dans un cercle).
- Exécutez **DEG** pour obtenir des degrés décimaux. C'est l'unité angulaire par défaut ; lorsque les témoins **RAD** ou **GRAD** ne sont pas affichés, les nombres sont interprétés comme des angles exprimés en degrés. Les chiffres situés après le séparateur décimal dans l'argument sont interprétés comme une fraction décimale d'un degré, et non comme des minutes et secondes.

Fonctions trigonométriques

- **SIN** (sinus) et **SIN⁻¹** (arc sinus)
- **COS** (cosinus) et **COS⁻¹** (arc cosinus)
- **TAN** (tangente) et **TAN⁻¹** (arc tangente)

Conversions

Conversions degrés/radians. Exécutez $\boxed{D\text{-}R}$ pour convertir un nombre exprimant un angle en degrés décimaux en un nombre exprimant le même angle en radians. Pour la conversion inverse, exécutez $\boxed{R\text{-}D}$.

Conversions décimal/sexagésimal. Les heures et les degrés peuvent être exprimés en format HMS (*heures - minutes - secondes*) plutôt que dans le format décimal habituel. Les deux premiers chiffres suivant le séparateur décimal sont alors interprétés comme des minutes, les deux chiffres suivants comme des secondes, et les chiffres suivants comme une *fraction* décimale de seconde. Par exemple :

$$HH.MMSSssss = HH \text{ heures} + MM \text{ minutes} + SS,ssss \text{ secondes (format HMS)}$$

$$HH + MM/60 + SS,ss/3600 \text{ (format décimal)}$$

Pour convertir au format HMS un nombre exprimé en format décimal, exécutez \boxed{HMS} (*conversion vers le format HMS*). Pour la conversion inverse, exécutez \boxed{HR} (*conversion vers le format sexagésimal*).

Conversion décimal/octal. Pour convertir un entier décimal en son équivalent octal (base 8), exécutez \boxed{OCT} (*vers le format octal*). Pour convertir un entier octal en son équivalent décimal (base 10), exécutez \boxed{DEC} (*vers le format décimal*).

Fonctions logarithmiques et exponentielles

Log décimal et puissance de dix. Appuyez sur \boxed{LOG} pour calculer le logarithme décimal (à base 10) du nombre se trouvant dans le registre X. Appuyez sur $\boxed{10^x}$ pour calculer 10 élevé à la puissance du nombre stocké dans le registre X.

Log népérien et exponentielle. Appuyez sur \boxed{LN} pour calculer le logarithme népérien (à base e) du nombre se trouvant dans le registre X. Appuyez sur $\boxed{e^x}$ pour calculer e élevé à la puissance du nombre stocké dans le registre X.

Les fonctions hyperboliques, hyperboliques inverses, et certaines fonctions financières calculent les expressions $\ln(1+x)$ et $e^x - 1$ pour des arguments proches de zéro et avec des résultats également proches de zéro. Pour permettre une plus grande précision, $\boxed{LN1+X}$ et $\boxed{E^X-1}$ calculent ces expressions directement :

- $\boxed{LN1+X}$ calcule $\ln(1+x)$,
- $\boxed{E^X-1}$ calcule $e^x - 1$.

Fonctions à deux nombres (ou fonctions « diadiques »)

Toutes les fonctions à deux nombres utilisent des opérandes dans les registres X et Y ; la plupart envoient un seul résultat dans le registre X, entraînant ainsi un mouvement de descente dans la pile (les pourcentages et les conversions de coordonnées polaires/rectangulaires font exception).

Opérations arithmétiques de base

Les schémas de la pile pour $+$, $-$, \times , et \div figurent au chapitre précédent. N'oubliez pas l'ordre de saisie des valeurs pour la soustraction et la division : x étant dans le registre X et y dans le registre Y, vous obtenez :

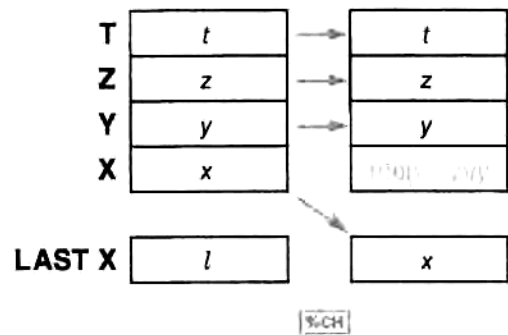
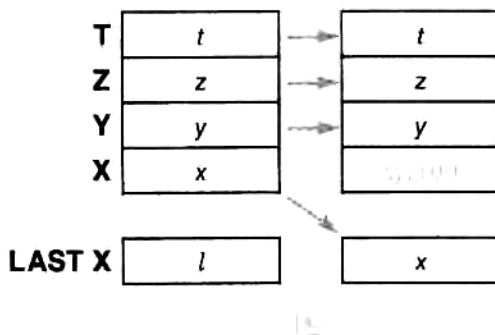
- $y - x$ avec la soustraction (et non pas $x - y$).
- y/x avec la division (et non pas x/y).

Opérations arithmétiques sur les temps

Pour ajouter ou soustraire des nombres qui sont en format HMS (heures-minutes-secondes), utilisez `HMS+` ou `HMS-`. L'ordre de saisie des valeurs et les mouvements de descente dans la pile sont les mêmes que pour l'addition et la soustraction normales.

Pourcentages

Les deux fonctions de pourcentage utilisent le nombre du registre Y comme base et remplacent le nombre introduit dans le registre X par le résultat de l'opération. Le nombre se trouvant dans le registre Y n'est donc pas modifié et aucun mouvement de descente ne se produit dans la pile.



Pourcentage. Pour calculer un pourcentage, placez le nombre de base dans le registre X et le taux du pourcentage dans le registre X, puis exécutez `%I`.

Différence relative en pourcent. Pour calculer l'augmentation ou la diminution d'un nombre par rapport à un autre, introduisez le premier nombre dans le registre Y et le second dans le registre X, puis exécutez `%CH`. Vous obtenez l'augmentation ou la diminution comme un pourcentage positif ou négatif du premier nombre.

Rapport de deux nombres, en pourcent. Pour calculer le pourcentage que représente un nombre par rapport à un autre :

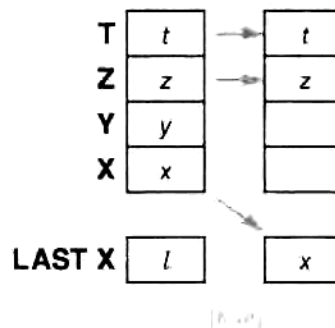
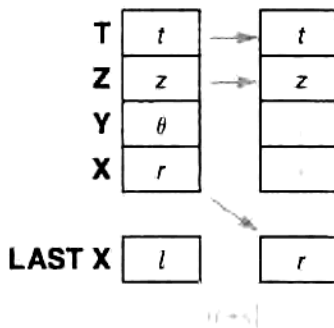
1. Placez dans le registre Y le nombre auquel vous voulez rapporter l'autre (nombre de base), et dans le registre X le nombre que vous voulez convertir en un pourcentage.
2. Exécutez $\boxed{1/x}$.
3. Exécutez $\boxed{\times}$.
4. Exécutez $\boxed{1/x}$.

Conversions polaires/rectangulaires

Un point dans un plan peut être décrit soit par des coordonnées polaires, soit par des coordonnées rectangulaires. Les coordonnées polaires sont le module r et l'argument θ (angle); les coordonnées rectangulaires sont l'abscisse x (horizontale) et l'ordonnée y (verticale). Ces coordonnées sont illustrées page 54. Deux fonctions, $\boxed{r \rightarrow \theta}$ et $\boxed{\theta \rightarrow r}$, permettent d'effectuer les conversions dans les deux sens.

- Pour convertir les coordonnées polaires en coordonnées rectangulaires, exécutez $\boxed{r \rightarrow \theta}$.
- Pour convertir des données rectangulaires en coordonnées polaires, exécutez $\boxed{\theta \rightarrow r}$. L'argument θ obtenu a le même signe que la valeur introduite comme coordonnée y .

Qu'il soit une entrée ou une sortie, l'argument θ est interprété en fonction de l'unité angulaire choisie. Dans les schémas ci-dessous, vous remarquerez l'ordre des coordonnées dans la pile, et le fait qu'aucun mouvement de descente ne se produit. Appuyez sur $\boxed{x \rightarrow y}$ pour visualiser le résultat stocké dans le registre Y.



Autres fonctions à deux nombres

Élévation d'un nombre à une puissance. Pour élever un nombre à une puissance, placez ce nombre dans le registre Y et la puissance dans le registre X, puis exécutez $\boxed{\wedge}$. Le mouvement de descente dans la pile est le même que pour les fonctions arithmétiques. Les valeurs autorisées pour x dépendent de la valeur de y :

- Si y est positif, x peut être un nombre quelconque.
- Si y est négatif, x doit être un entier.

- Si y est égal à zéro, x doit être positif.

Toute autre combinaison entraîne l'apparition du message **DATA ERROR** (erreur de donnée).

Calcul des racines. Pour calculer la racine n ème d'un nombre :

1. Placer ce nombre dans le registre Y.
2. Placer n dans le registre X.
3. Exécutez $\boxed{1/x}$.
4. Exécutez $\boxed{\sqrt{}}$.

Modulo. x et y étant des entiers positifs se trouvant respectivement dans les registres X et Y, la fonction $\boxed{\text{MOD}}$ calcule le reste de l'opération y divisé par x (« $y \bmod x$ »). Par exemple, vous pouvez vérifier si y est multiple de x en exécutant la fonction $\boxed{\text{MOD}}$ et en regardant si le résultat est zéro. Les mouvements de descente dans la pile sont les mêmes que pour les fonctions arithmétiques.

Vous pouvez également utiliser $\boxed{\text{MOD}}$ avec des nombres qui ne sont pas des entiers positifs. L'équation générale de $y \bmod x$ est $y - x \langle y/x \rangle$, où $\langle y/x \rangle$ représente le plus grand entier inférieur à y/x . Lorsque vous effectuez $y \bmod x$ avec $x = 0$, vous obtenez y .

Statistiques

Les calculs statistiques comportent deux étapes. Vous introduisez d'abord les données à partir de la pile ; votre HP-41 fait la sommation des résultats intermédiaires sur la base de ces données. Ensuite, vous exécutez les calculs statistiques ; votre HP-41 utilise les résultats intermédiaires pour calculer les résultats d'ensemble, qui sont ensuite stockés dans la pile. Les opérations statistiques de base sont décrites aux pages 55 à 58.

Registres statistiques

Les registres statistiques sont un bloc de six registres de données se trouvant dans la mémoire principale, et qui contiennent les résultats intermédiaires constituant la sommation des données que vous avez introduites. Lorsque la mémoire de votre HP-41 est réinitialisée, les registres statistiques sont les registres R_{11} à R_{16} .

- Vous pouvez affecter à d'autres registres de stockage la fonction de registres statistiques en exécutant $\boxed{\Sigma\text{REG}}$ et en spécifiant l'adresse du premier registre dans le bloc que vous choisissez. Cette affectation est alors sauvegardée en mémoire permanente.
- Pour vérifier les affectations des registres, exécutez $\boxed{\Sigma\text{REG?}}$. Vous obtenez dans le registre X * l'adresse du premier registre du bloc de registres statistiques.
- Pour placer des zéros dans les six registres statistiques, exécutez $\boxed{\text{CLT}}$.

* La fonction $\boxed{\Sigma\text{REG?}}$ ne vérifie pas si les registres affectés aux calculs statistiques existent réellement (c'est-à-dire si une quantité suffisante de mémoire a été affectée au stockage des données), mais fournit uniquement l'adresse de ces registres.

Les registres statistiques font la sommation des statistiques intermédiaires suivantes à partir des données introduites dans les registres X et Y.

Registres statistiques

Registre	Contenu	
R ₁₁	Σx	Sommation des valeurs x .
R ₁₂	Σx^2	Sommation des carrés des valeurs x .
R ₁₃	Σy	Sommation des valeurs y .
R ₁₄	Σy^2	Sommation des carrés des valeurs y .
R ₁₅	Σxy	Sommation des produits des valeurs x et y .
R ₁₆	n	Nombre d'observations statistiques (ou couples de données) accumulées. (Affiché.)

Saisie de données

Sommation des couples de données. Lorsque vous appuyez sur $\boxed{\Sigma+}$:

- Les résultats des calculs sur les nombres des registres Y et Y sont ajoutés aux cinq premiers registres statistiques. Si le contenu d'un registre vient ainsi à dépasser $+ 9,99999999 \times 10$, il n'y a pas d'erreur pour dépassement de capacité ; le registre concerné contient la valeur $+ 9,99999999 \times 10$.
- Le nombre n de couples de données (ou observations) contenu dans le sixième registre est incrémenté et sa valeur ainsi actualisée est envoyée dans le registre X.
- Le nombre se trouvant précédemment dans le registre X est sauvegardé dans le registre LAST X.
- La montée des données dans la pile est inhibée, de sorte que la prochaine donnée introduite remplacera n dans le registre X.

Vous pouvez faire la sommation d'observations comportant une ou deux valeurs, comme indiqué dans la partie I du présent manuel. Si vous faites seulement la sommation des valeurs x , commencez par effacer le registre Y (0 $\boxed{\text{ENTER} \uparrow}$). Comme $\boxed{\Sigma+}$ et $\boxed{\downarrow}$ inhibent les mouvements de montée dans la pile, le registre Y restera effacé pendant que vous ferez la sommation des valeurs x .

Correction d'erreurs. Pour corriger des données erronées dont vous avez fait la sommation :

1. Ré-introduisez les données erronée. Si vous en avez seulement fait la sommation, il vous suffit d'appuyer sur $\boxed{\text{LAST} \times}$ pour les rappeler (la valeur y erronée est toujours dans le registre Y et la valeur x erronée est sauvegardée dans le registre LAST X).
2. Appuyez sur $\boxed{\downarrow}$. Cette fonction joue le même rôle que $\boxed{\Sigma+}$, sauf que les résultats sont soustraits des cinq premiers registres statistiques (et non ajoutés), et que le sixième registre est décrémenté d'une unité (au lieu d'être incrémenté).
3. Introduisez les données correctes.
4. Appuyez sur $\boxed{\Sigma+}$.

Limitations de la valeur des données. Votre HP-41 peut se trouver dans l'impossibilité d'effectuer certains calculs statistiques si la différence entre vos données est relativement faible. Pour éviter cet inconvénient, vous devez normaliser vos données en les introduisant sous la forme d'une différence par rapport à une valeur de référence (la moyenne par exemple). Il vous faut ensuite rajouter cette différence aux calculs sur la moyenne. Par exemple, si les valeurs successives de x sont 665999, 666000 et 666001, vous devez introduire les données sous la forme -1 , 0 , et 1 , puis ajouter 666000 aux résultats correspondants.

Moyenne

En exécutant $\overline{x \pm y}$, vous obtenez la moyenne arithmétique \bar{x} des valeurs x sommées dans le registre X, et la moyenne arithmétique \bar{y} des valeurs y sommées dans le registre Y, conformément aux formules ci-dessous.

$$\bar{x} = \frac{\sum x}{n}, \quad \bar{y} = \frac{\sum y}{n}.$$

Appuyez sur $\overline{x \pm y}$ pour afficher la valeur obtenue pour \bar{y} . Le nombre se trouvant précédemment dans le registre X est alors sauvegardé dans le registre LAST X; le nombre précédemment dans le registre Y est perdu.

Ecart-type

En exécutant $s_{x \pm y}$, vous obtenez l'écart-type s_x des valeurs sommées dans le registre X et l'écart-type s_y des valeurs sommées dans le registre Y, conformément aux formules ci-dessous :

$$s_x = \sqrt{\frac{n\sum(x^2) - (\sum x)^2}{n(n-1)}}, \quad s_y = \sqrt{\frac{n\sum(y^2) - (\sum y)^2}{n(n-1)}}.$$

Appuyez sur $\overline{x \pm y}$ pour afficher la valeur \bar{y} obtenue. Le nombre précédemment dans le registre X est sauvegardé dans le registre LAST X et celui précédemment dans le registre Y est perdu.

Mémoire principale

Table des matières

Organisation	194
Mémoire de programmation	196
Lignes de programme	197
Multiplats nuls	197
Compactage	198
Mémoire des alarmes	198
Mémoire du clavier personnel	198
Registres de stockage des données	199
Affectation	199
Registres au-dessus de R_{99}	199
Opérations sur les registres de stockage de données	199
Stockage et rappel	199
Arithmétique directe dans les registres	201
Permutation	201
Opérations sur des blocs de registres	201
Effacement des registres	202

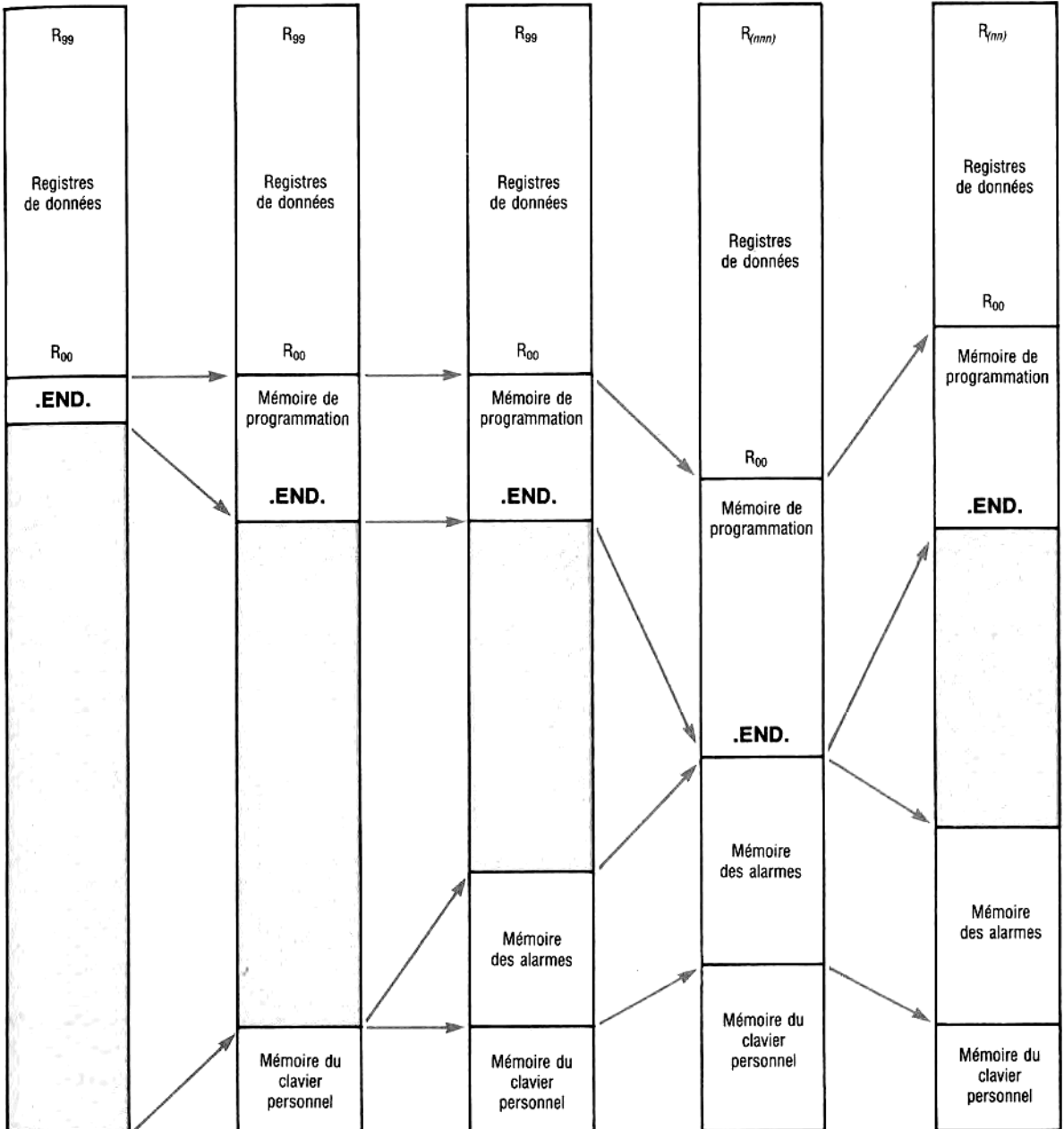
Organisation

La mémoire principale contient 319 registres (un registre contient 7 multiplats, et la mémoire comporte donc 2233 multiplats). Ces registres sont divisés en deux grands groupes* :

- Un groupe comprend les registres de stockage des données. Le nombre de registres de la mémoire principale affectés au stockage des données ne change que lorsque vous exécutez une fonction particulière pour spécifier ces registres.
- L'autre groupe comprend les programmes, les alarmes, les redéfinitions des touches du clavier personnel, et les registres non affectés. Les registres non affectés constituent une « réserve » qui est automatiquement affectée aux programmes, alarmes et redéfinitions de touches à mesure qu'ils nécessitent de nouveaux registres. Toutefois, la taille de ce groupe ne change que lorsque vous modifiez le nombre de registres affectés au stockage des données.

* La mémoire principale contient en fait 320 registres, mais la mémoire de programmation comprend toujours au moins un registre pour l'instruction **.END.** permanente.

Configurations de la mémoire principale



Le schéma de la page précédente montre cinq configurations de la mémoire principale — chaque colonne représentant la totalité de la mémoire principale. La colonne la plus à gauche représente la configuration par défaut, avec 100 registres pour le stockage des données, et 219 pour toutes les autres utilisations. Les colonnes situées à droite de cette première colonne montrent quatre stades ultérieurs de la mémoire principale. Votre HP-41 gère la plupart de ces détails automatiquement, mais une bonne compréhension de la mémoire principale peut vous aider à l'utiliser plus efficacement.

La première colonne représente la configuration par défaut après que la mémoire principale ait été effacée. 100 registres sont consacrés au stockage des données, le registre ayant le numéro le plus élevé se trouvant en haut. Le 1^{er} registre immédiatement en dessous du bloc de stockage des données contient l'instruction permanente `.END.`, qui marque la fin de la mémoire de programmation. Les registres non affectés, en dessous de cette instruction, sont disponibles pour les programmes, des alarmes, et des redéfinitions de touches du clavier personnel.

La seconde colonne montre la mémoire principale après que vous ayez introduit des programmes et affecté des fonctions aux touches du clavier personnel. La mémoire de programmation utilise les registres non affectés à mesure que l'instruction permanente `.END.` est poussée vers le bas par les nouvelles lignes de programme. La mémoire du clavier personnel utilise un registre non affecté pour deux redéfinitions de touches.

La troisième colonne montre la mémoire des alarmes après que vous ayez armé des alarmes. La quatrième colonne montre la mémoire après que vous ayez affecté des registres supplémentaires au stockage des données, écrit de nouveaux programmes, armé d'autres alarmes, et redéfini de nouvelles touches du clavier personnel. Lorsque tous les registres sont affectés, toute opération ayant besoin de nouveaux registres en mémoire principale génère une situation d'erreur. Si vous armez une alarme, le HP-41 affiche `NO ROOM` (plus d'espace). D'autres opérations font apparaître `PACKING` (compactage) puis `TRY AGAIN` (essayez à nouveau).

Si le compactage ne libère pas un nombre suffisant de registres non affectés, vous devez réduire la taille du bloc de registres consacrés au stockage des données ou effacer d'autres parties de la mémoire. Vous pouvez passer en revue le contenu de la mémoire de programmation, d'alarmes, et du clavier personnel en exécutant respectivement `⇐`, `⇐`, `⇐`, 1, 5 ou 6; n'oubliez pas que vous pouvez aussi stocker des données et des programmes dans des fichiers de données ou de programmes en mémoire annexe.

La cinquième colonne montre que des registres non-affectés sont à nouveau disponibles après que vous ayez réduit le nombre de registres consacrés au stockage des données et effacé une partie de la mémoire.

Mémoire de programmation

Lorsque la mémoire permanente est effacée, la mémoire de programmation ne contient plus que l'instruction permanente `.END.`. Si vous appuyez sur `⇐` `□` `□` et introduisez un programme au clavier, chaque instruction de ce programme est insérée immédiatement devant l'instruction permanente `.END.`, qui se déplace à chaque fois vers le bas pour faire de la place. Par conséquent :

- La première instruction du programme se situe en haut de la mémoire de programmation.
- La dernière instruction précède le `.END.` permanent, en bas de la mémoire de programmation.

Le catalogue 1 montre le nombre de registres non affectés et l'instruction permanente `.END.` (`.END. REG nnn`). Il peut y avoir encore six multipléts disponibles (presque un registre complet) en plus des *nnn* registres.

Lignes de programme

Dans un programme, chaque fonction, nombre, ou chaîne alphanumérique est considérée comme constituant une ligne de programme. Le nombre de *lignes de programme* dépend du nombre de fonctions, nombres et chaînes alphanumériques contenues dans le programme; le nombre de *registres* et *multipléts* occupés par ces lignes de programme dépend des fonctions et de la longueur des ombres et chaînes alphabétiques :

- Les fonctions requièrent entre un et quatre multipléts, selon la fonction et le paramètre, si elle en comprend un. Le nombre exact de multipléts requis pour chaque fonction figure dans les tableaux de fonctions, à la fin du présent manuel.
- Les fonctions dont le paramètre est un label global nécessitent un multiplét par caractère en plus de leur longueur normale.
- Les nombres nécessitent un multiplét par chiffre, plus un multiplét supplémentaire pour chaque `□`, `[CHS]` ou `[EEX]` introduit avec le nombre.
- Les chaînes alphabétiques nécessitent un multiplét par caractère, plus un multiplét supplémentaire pour la totalité de la chaîne.

Multipléts nuls

Généralement, le premier multiplét d'une instruction succède immédiatement au dernier multiplét de l'instruction précédente. Cependant, il peut parfois y avoir des multipléts nuls entre deux instructions. Ces multipléts ont pour origine :

Suppression d'une instruction. Lorsque vous supprimez une instruction, les multipléts qu'elle occupait sont remplacés par des multipléts nuls.

Insertion d'une instruction dans un programme. Si des multipléts nuls ne sont pas déjà disponibles à l'endroit où vous voulez insérer une nouvelle instruction, sept multipléts nuls viennent s'insérer à cet endroit et toutes les instructions suivantes se trouvent repoussées plus bas dans la mémoire. La nouvelle instruction remplace ensuite les multipléts nuls et, si elle occupe moins de sept multipléts, le reste des multipléts nuls reste en place.

Lignes de programme qui sont des nombres. Le HP-41 place un multiplét nul devant une chaîne de multipléts représentant un nombre, pour le cas où la ligne de programme précédente serait aussi un nombre. Ce multiplét nul joue le rôle d'un séparateur entre les deux lignes de programme, de manière qu'elles ne soient pas interprétées comme un seul et même nombre.

Compactage

Lorsque votre programme est terminé, les seuls multipliants nuls utiles sont ceux qui séparent les lignes successives représentant des nombres. Pour éliminer les autres multipliants nuls, exécutez `[GT0] [] []`. Lorsque la mémoire est compactée, les multipliants de tous les programmes se déplacent vers le haut en mémoire de programmation pour remplacer les multipliants nuls inutiles. (La mémoire du clavier personnel est également compactée dans les cas indiqués ci-dessous). La mémoire principale est compactée lorsque :

- Vous exécutez `[PACK]`.
- Vous exécutez `[GT0] [] []`.
- Vous effacez les programmes en exécutant `[CLP]` ou `[PCLPS]`.
- Le nombre de registres non affectés encore disponibles n'est pas suffisant pour permettre une opération nécessitant de nouveaux registres. Ces opérations sont : l'augmentation du nombre de registres affectés au stockage des données, l'introduction d'une ligne de programme, ou l'affectation d'une fonction à une touche.

Mémoire des alarmes

Lorsque vous armez la première alarme, deux registres non affectés servent à définir les limites de la mémoire des alarmes (ces deux registres viennent s'ajouter aux registres utilisés par l'alarme). Ces registres ne sont libérés que lorsque toutes les alarmes sont effacées. Le nombre de registres occupé par chaque alarme dépend du type de l'alarme.

- Chaque alarme utilise un registre pour l'heure et la date de son prochain déclenchement.
- Une alarme ayant une période de répétition nécessite un registre supplémentaire.
- Une alarme utilisant le registre alphabétique requiert un registre pour 7 caractères (* est un caractère ordinaire servant de repère pour le décompte). Par exemple, un message de 18 caractères nécessite deux registres pour les 14 premiers caractères et un registre pour les 4 derniers, et occupe au total trois registres.

Mémoire du clavier personnel

Lorsque vous affectez une fonction à une touche, l'information est stockée dans la mémoire du clavier personnel. L'affectation d'une fonction ou d'un label global résidant dans un module enfichable est également stockée dans la mémoire du clavier personnel. Cependant, lorsque vous affectez à une touche un label global figurant dans le catalogue 1, cette information *n'est pas* stockée dans la mémoire du clavier personnel, mais va dans la mémoire de programmation, avec ce label global.

Un registre peut contenir deux affectations. La première affectation nécessite un registre, et la seconde vient finir de remplir ce registre. Ainsi, chaque fois que vous réalisez une affectation « impaire » (qui porte le nombre total des affectations à un nombre impair), un nouveau registre vient s'ajouter à la mémoire du clavier personnel, et chaque fois que vous réalisez une affectation portant un numéro pair, le registre précédemment entamé se trouve rempli.

Une affectation est annulée lorsque vous affectez une nouvelle fonction à la même touche, ou lorsque vous annulez expressément une affectation par l'une des méthodes décrites à la page 168. Si les deux affectations contenues dans un registre ont été annulées, et si la mémoire principale est compactée, ce registre devient un registre non affecté.

Registres de stockage des données

Affectation

Modification de l'affectation. Deux fonctions permettent d'affecter des registres de mémoire principale au stockage des données ; l'une est destinée à être exécutée manuellement, et l'autre à être exécutée sous programme. Lorsque vous réduisez le nombre de registres de stockage, ce sont les registres ayant les numéros les plus élevés qui sont atteints.

- Vous modifiez manuellement le nombre de registres de stockage des données en exécutant `SIZE` puis en spécifiant le nombre de registres à affecter au stockage.
- Vous pouvez aussi réaliser cette modification sous programme en plaçant dans le registre X le nombre de registres à affecter au stockage des données, puis en exécutant `PSIZE`.

Vérification de l'affectation. Pour vérifier le nombre de registres qui sont affectés au stockage des données, exécutez `SIZE?`. Le registre X affiche alors le nombre total de ces registres.

Exemple. Le programme ci-dessous ne modifie l'affectation que si le nombre des registres de stockage est trop petit.

01 SIZE?	Affiche le nombre de registres actuellement affectés au stockage des données (registre X).
02 50	Nombre de registres de stockage requis par le programme. Le nombre actuel de ces registres (issu de la ligne 01) passe dans le registre Y.
03 X>Y?	Le nombre requis de registres de données est-il supérieur au nombre actuel ?
04 PSIZE	Si oui, modifier l'affectation actuelle pour obtenir l'affectation requise.

Registres au-dessus de R₉₉

Si vous affectez plus de 100 registres au stockage des données, vous ne pouvez accéder aux registres dont l'adresse est supérieure à 99 que par adressage indirect. Pour souligner cette différence importante, le présent manuel représente les adresses à trois chiffres entre parenthèses : R₍₁₂₀₎, par exemple. A moins que vous ne vouliez utiliser l'adressage indirect dans une boucle ou pour effectuer des opérations particulières en mémoire principale (arithmétique directe dans les registres par exemple), il est souvent plus facile d'utiliser un fichier de données en mémoire annexe.

Opérations sur les registres de stockage de données

Stockage et rappel

Les données des registres de stockages peuvent avoir trois sources/destinations : les registres de la pile opérationnelle, le registre alphabétique, et les fichiers de données en mémoire annexe. Les fonctions qui déplacent les données entre les registres de la pile ou le registre alphabétique et les registres de stockage en mémoire principale sont décrites dans le présent chapitre. Toutes les fonctions donnant accès aux fichiers de données en mémoire annexe sont décrites au chapitre 13, « Mémoire annexe ».

Spécification du registre en tant que paramètre. La plupart des fonctions ayant trait aux registres de données ne permettent l'accès qu'à un seul registre, dont l'adresse doit être spécifiée en tant que paramètre. Il y a plusieurs façons de spécifier un paramètre :

- De R_{00} à R_{99} , introduisez les deux chiffres de l'adresse.
- Par commodité, R_{01} à R_{10} peuvent être spécifiés avec une seule touche, faisant partie des deux rangées supérieures du clavier.
- Pour les registres de la pile ou le registre LAST X, appuyez sur \square , puis sur \square , \square , \square , \square ou \square .
- Pour tout registre devant faire l'objet d'un adressage indirect, appuyez sur la touche jaune, puis spécifiez l'adresse du registre indirect par l'une des trois méthodes ci-dessus.

Stockage. Pour copier les données du registre X dans un registre de stockage, appuyez sur \square puis spécifiez le registre de destination. Le registre X demeure inchangé ; les données précédemment dans le registre de stockage sont perdues.

Rappel. Pour copier dans le registre X les données d'un registre de stockage, appuyez sur \square puis spécifiez le registre source. Le contenu du registre source reste inchangé. Si les mouvements de montée dans la pile étaient inhibés, les données rappelées remplacent le contenu du registre X ; sinon il se produit un mouvement de montée des données dans la pile.

Stockage alphabétique. Pour copier dans un registre de données les six caractères les plus à gauche du registre alphabétique, appuyez sur \square puis spécifiez le registre de destination. Le contenu du registre alphabétique reste inchangé et les données précédemment dans le registre de destination sont perdues.

- Tout signe de ponctuation est décompté comme l'un des six caractères.
- Une chaîne de chiffres dans le registre alphabétique ne constitue pas un nombre. Si vous stockez des chiffres alphabétiques dans un registre, le contenu *semble* être un nombre, mais vous ne pouvez l'inclure dans des opérations numériques.
- Si vous voulez effectuer des opérations numériques sur les chiffres du registre alphabétique, vous devez utiliser \square pour interpréter ces chiffres comme un nombre et les placer dans le registre X (page 311).
- Copier les données du registre alphabétique dans le registre X à l'aide de la fonction \square n'est pas une opération équivalente à \square ; en effet, il ne se produit pas de montée dans la pile, et le contenu précédent du registre X est donc perdu.

Pour copier plus de six caractères dans un registre de données, vous devez modifier le contenu du registre alphabétique avant de répéter \square (sinon vous copiez à nouveau les mêmes caractères).

- Pour éliminer du registre les six caractères déjà copiés, exécutez \square (*Alpha Shift* = décalage alphabétique). Les six caractères les plus à gauche sont alors poussés hors du registre alphabétique.
- Pour faire passer les six caractères déjà copiés à l'extrémité droite du registre alphabétique, introduisez 6 dans le registre X et exécutez \square (*Alpha rotate* = permutation alphabétique).

Rappel alphabétique. Pour copier les données d'un registre de stockage dans le registre alphabétique, exécutez \square puis spécifiez le registre source. Le contenu du registre source demeure inchangé, les données sont annexées au contenu du registre alphabétique, et la saisie de caractères est validée. Si vous voulez que les données à copier constituent le début d'un nouveau message, exécutez \square avant de rappeler ces données.

Arithmétique directe dans les registres

Cette arithmétique vous permet de combiner un nombre se trouvant dans le registre X à un nombre se trouvant dans un registre de stockage de données sans rappeler dans la pile le nombre stocké.

- En exécutant `[STO][+]` *nn*, vous ajoutez le nombre du registre X au nombre du registre R_{nn} , puis stockez leur somme dans R_{nn} .
- En exécutant `[STO][-]` *nn*, vous soustrayez le nombre du registre X du nombre stocké dans R_{nn} , puis stockez le résultat dans R_{nn} .
- En exécutant `[STO][x]` *nn*, vous multipliez les nombres des registres X et R_{nn} , puis stockez le résultat dans R_{nn} .
- En exécutant `[STO][÷]` *nn*, vous divisez le nombre se trouvant dans le registre X par le nombre se trouvant dans R_{nn} , puis stockez le quotient dans R_{nn} .

Comme c'est le cas avec `[STO]`, le nombre initialement dans R_{nn} est perdu et le nombre se trouvant dans le registre X demeure inchangé. Cela vous permet de réutiliser une constante dans le registre X sans avoir à exécuter `[LAST]`.

Permutation

Vous remarquerez que `[STO]` et `[RCL]` dupliquent un nombre et perdent l'autre. Pour déplacer des nombres sans duplication ni perte de données, exécutez `[X<>]` et spécifiez le registre dont vous voulez permuter le contenu avec celui du registre X.

Opérations sur des blocs de registres

Vous pouvez déplacer ou permuter des blocs de registres en utilisant `[REGMOVE]` ou `[REGSWAP]`. Ces deux fonctions utilisent un code de commande *sss.dddmmm* dans le registre X pour définir les blocs source et destination :

- R_{sss} est le premier registre (avec l'adresse la plus petite) dans le bloc *source*.
- R_{ddd} est le premier registre (avec l'adresse la plus petite) dans le bloc *destination*.
- *mmm* représente le nombre de registres des blocs. Si vous ne spécifiez pas *mmm*, la valeur par défaut est *mmm* = 001.

Les blocs ne peuvent pas se chevaucher. Assurez-vous que $|sss - ddd| \geq mmm$.

Déplacement du contenu d'un bloc de registres. Pour copier le bloc source dans le bloc destination, placez le code de commande dans le registre X et exécutez `[REGMOVE]`. Comme pour `[STO]`, le contenu du bloc source reste inchangé, et le contenu précédent du bloc destination est perdu.

Permutation du contenu de deux blocs de registres. Pour permuter le contenu de deux blocs de registres, placez le code de commande dans le registre X et exécutez `[REGSWAP]`. Peu importe quel bloc est défini comme source et quel bloc comme destination.

Effacement des registres

Effacement d'un seul registre. Pour effacer un seul registre, stockez un zéro dans ce registre.

Effacement d'un bloc de registres. En commençant un programme, il est prudent d'effacer les registres de stockage de données qui doivent être utilisés par ce programme. Pour effacer un bloc de registres, exécutez `CLRGX` (*clear registers by X* = effacer registres définis dans X), en plaçant le code de commande dans le registre X. Ce nombre a la forme *ddd.fffii*, avec :

- R_{ddd} (*début*) est le premier registre (avec la plus petite adresse) à être effacé.
- R_{fff} (*fin*) est le dernier registre (avec la plus grande adresse) à être effacé.
- *ii* est l'*incrément*, si vous ne voulez un effacement que tous les *ii* registres. Si vous ne spécifiez pas *ii*, la valeur par défaut est $ii = 01$.

Par exemple, supposons que vous ayez stocké séquentiellement en mémoire une matrice de 4×4 , ligne par ligne. Vous pouvez effacer n'importe quelle ligne de la matrice en spécifiant la première et la dernière adresse dans la ligne, sans avoir à définir un incrément. Mais pour effacer une colonne (dont les valeurs ne sont pas séquentielles en mémoire), spécifiez d'abord l'adresse de la première et de la dernière valeur dans la colonne, puis choisissez $ii = 04$ pour effacer uniquement les registres correspondant à cette colonne.

R_{ddd} est effacé même si $ddd > fff$ ou $ddd + ii > fff$. On néglige le signe du code de commande et tous les chiffres décimaux.

Effacement de tous les registres. Pour effacer tous les registres de stockage des données, exécutez `CLRG`.

Mémoire annexe

Table des matières

Introduction	205
Fichiers de la mémoire annexe	205
Renseignements contenus dans l'en-tête	205
Spécification du fichier	206
Catalogue de la mémoire annexe	206
Vérification de la taille d'un fichier	208
Élimination d'un fichier	208
Opérations sur les fichiers de programme	208
Sauvegarde d'un programme	208
Accès à un programme	209
Redéfinition automatique des touches	211
Vérification de la taille d'un programme	211
Création de fichiers de données et de texte	211
Création de fichiers de données	211
Création de fichiers de texte	212
Modification de la taille des fichiers	213
Effacement des fichiers	213
Pointeurs des fichiers de données et de texte	213
Structure des fichiers de données	214
Structure des fichiers de texte	214
Opérations sur les pointeurs	215
Opérations sur les fichiers de données	216
Accès à tous les registres de données	217
Accès à un bloc de registres de données	218
Accès au registre X	220
Opérations sur les fichiers de texte	222
Vérification du nombre de multipléts disponibles	222
Opérations sur les enregistrements	222
Opérations sur les caractères	224
Recherche dans un fichier	226
Copie de données dans le registre alphabétique	226
Accès à des fichiers de stockage de masse	227

Introduction

La mémoire annexe est une extension de la mémoire principale, mais elle présente en outre certains avantages spécifiques. Toutes les informations contenues dans la mémoire annexe sont organisées en fichiers ; il existe trois types de fichiers, qui correspondent aux trois types d'informations possibles :

- Le fichier de programme, dans lequel vous pouvez stocker un programme se trouvant dans la mémoire principale. Mais vous devez retransférer ce programme en mémoire principale pour l'exécuter ou le modifier.
- Le fichier de données, dans lequel vous pouvez stocker le contenu d'un registre. Un fichier de données est un ensemble de registres. En une seule opération sur ce fichier de données, vous pouvez accéder à tous les registres de données de la mémoire principale, à un bloc de registres de données, ou au registre X. La possibilité d'accéder directement au registre X fait de la mémoire annexe non seulement une extension de la mémoire principale, mais également une solution de remplacement.
- Le fichier de texte, dans lequel vous pouvez stocker les données alphabétiques (également appelé fichier ASCII), ce fichier étant un ensemble de chaînes alphabétiques. La taille et la structure des fichiers de texte, combinées à la possibilité d'accès direct l'intermédiaire d'un éditeur de texte, confèrent à ces fichiers des possibilités alphabétiques qui surpassent de loin celles de la mémoire principale.

Comme chaque fichier requiert deux registres pour sa gestion par le HP-41CX, le nombre de registres dont vous disposez dans la mémoire annexe dépend du nombre de fichiers. Un fichier unique peut contenir jusqu'à 124 registres utilisables, deux fichiers jusqu'à 122 registres utilisables, etc. D'une manière générale, n fichiers offrent $126 - 2n$ registres utilisables.

Vous pouvez augmenter le nombre de registres de la mémoire annexe en ajoutant à votre HP-41CX un ou deux modules de mémoire annexe HP 82181A (ces modules sont décrits dans l'annexe E). Chaque module ajoute 238 registres. Le nombre maximal de registres de mémoire annexe est donc $124 + 238 + 238 = 600$, soit près de deux fois plus que dans la mémoire principale.

Fichiers de la mémoire annexe

Renseignements contenus dans l'en-tête

Chaque fichier commence par un *en-tête*, constitué de deux registres décrivant le fichier. Bien que vous ne puissiez jamais accéder directement aux en-têtes, vous devez connaître la quantité de mémoire qu'ils consomment et les renseignements qu'ils contiennent. Un en-tête contient :

Le nom du fichier. Lorsque vous créez un fichier, vous lui donnez un nom. Ce nom peut comprendre n'importe quel caractère à l'exception de la virgule, avec un maximum de sept caractères*. Vous introduisez le nom du fichier dans le registre alphabétique lorsque vous créez le fichier ou y faites référence.

Le type du fichier. Il peut s'agir d'un fichier de programme, de texte ou de données. Un fichier de programme est une copie d'un programme se trouvant en mémoire principale ; un fichier de données est un ensemble de registres de données ; et un fichier de texte est un ensemble de chaînes de caractères alphabétiques.

* Un nom de fichier ne peut pas être constitué de sept multipliants d'une valeur décimale de 255.

La taille du fichier. Il s'agit du nombre de registres de mémoire annexe requis par le fichier. Pour les fichiers de programme, l'en-tête contient également la taille du programme en multiplés.

Le pointeur. Les fichiers de données et de texte utilisent un nombre, appelé pointeur, pour indiquer à quel élément du fichier l'utilisateur va accéder. Les fichiers de programme ne comportent pas de pointeurs.

Spécification du fichier

Vous ne pouvez accéder qu'à un seul fichier à la fois. Le HP-41CX garde trace du dernier fichier auquel vous avez accédé; ce fichier devient le *fichier en cours*. Seules quelques fonctions de la mémoire annexe vous donnent la possibilité de spécifier le fichier auquel vous voulez accéder.

- Certaines fonctions *nécessitent* qu'un nom de fichier se trouve déjà dans le registre alphabétique. Par exemple les fonctions sur les fichiers de programme et les fonctions destructrices.
- Certaines fonctions *vous permettent* de spécifier un nom de fichier dans le registre alphabétique. Si le registre alphabétique est vide, ces fonctions s'exercent sur le fichier en cours.
- Certaines fonctions *n'ont pas besoin* d'un nom de fichier dans le registre alphabétique. Ces fonctions s'exercent uniquement sur le fichier en cours.

Catalogue de la mémoire annexe

Les fonctions **EMDIR** (*extended memory directory* = répertoire mémoire annexe) et **EMDIRX** (*extended memory directory by X* = répertoire fichier X) sont programmables et permettent d'accéder aux fichiers de la mémoire annexe. **EMDIR**, qui énumère tous les fichiers, est conçue pour être utilisée en interaction avec l'utilisateur; **EMDIRX**, qui donne des renseignements sur un fichier donné, est conçue pour les opérations automatiques sous programme.

Utilisation de **EMDIR.** Par commodité, vous pouvez exécuter **EMDIR** manuellement, comme un équivalent du catalogue 4. Si vous voulez l'affecter à une touche ou l'introduire dans une ligne de programme, vous devez utiliser le nom alphabétique de la fonction **EMDIR**. Lorsque vous exécutez **EMDIR** :

- Les fichiers sont énumérés dans l'ordre où ils ont été créés.
- Les nom, type (P = programme, D = données, A = ASCII ou texte), et nombre de registres de chaque fichier sont indiqués. Par exemple, les renseignements relatifs à un fichier ABC occupant 20 registres seraient affichés sous la forme suivante :

ABC	D020
-----	------

- Vous pouvez accélérer le défilement des fichiers en maintenant enfoncée n'importe quelle touche, sauf **R/S** et **ON**.
- Vous pouvez arrêter le défilement en appuyant sur **R/S** et le relancer en appuyant à nouveau sur cette même touche.
- Lorsque le défilement est arrêté, vous pouvez passer au fichier suivant en appuyant sur **SST** et au fichier précédent en appuyant sur **IST**. Une pression sur **SST** ou sur **IST** lorsque respectivement le dernier ou le premier fichier est affiché n'est suivie d'aucun effet.

Si vous laissez le catalogue défiler jusqu'au bout, la fonction `EMDIR` s'invalide automatiquement en fin de défilement. L'affichage vous indique alors le nombre de registres encore disponibles pour le prochain fichier à créer. Les deux registres requis pour le nouvel en-tête sont pris en compte, de sorte que vous pouvez réellement utiliser la totalité des registres indiqués pour stocker vos données. Ce nombre va dans le registre X, générant une montée dans la pile opérationnelle, sauf si les mouvements de montées sont inhibés. Vous pouvez aussi choisir d'invalider vous-même la fonction `EMDIR` à tout moment.

Lorsque le HP-41CX invalide automatiquement `EMDIR`, le fichier en cours reste le même qu'initialement. En revanche, vous pouvez faire de n'importe quel fichier le fichier en cours en interrompant le défilement lorsque le fichier voulu est affiché, puis en appuyant sur `▣`. Cela invalide `EMDIR`; dans ce cas, le nombre de registres encore disponibles *n'apparaît pas* dans le registre X.

Si la mémoire annexe ne comprend aucun fichier, le HP-41CX affiche le message `DIR EMPTY` (catalogue vide) et fait apparaître dans le registre X le nombre de registres encore disponibles, engendrant une montée dans la pile opérationnelle, sauf si les mouvements de montée sont inhibés.

Utilisation de `EMDIRX`. Pour déterminer le nom et le type du *n*ème fichier en mémoire annexe, placez *n* dans le registre X et exécutez `EMDIRX`. Par exemple, *n* = 1 pour le premier fichier créé (qui est aussi le premier fichier affiché par la fonction `EMDIR`).

Si le *n*ème fichier existe :

- Le nom du *n*ème fichier apparaît dans le registre alphabétique.
- Le type du fichier apparaît dans le registre X sous la forme d'un code à deux lettres*.
 - PR = fichier de programme.
 - DA = fichier de données.
 - AS = fichier ASCII (ou de texte).
- *n* va dans le registre LAST X.
- Le *n*ème fichier devient le fichier en cours.

Si le *n*ème fichier n'existe pas :

- Le registre alphabétique est effacé.
- Un zéro apparaît dans le registre X.
- *n* va dans le registre LAST X.
- Le fichier en cours reste le même.

Vous remarquerez que `EMDIRX` n'engendre *pas* de montée dans la pile opérationnelle. Le paramètre *n* est la valeur absolue de la partie entière du nombre se trouvant dans le registre X.

* Si le HP-41CX n'identifie pas le type du fichier, il affiche un entier positif.

Utilisation de `EMROOM` . Pour vérifier combien de registres sont encore disponibles pour le prochain fichier à créer, exécutez `EMROOM` (*extended memory room* = espace en mémoire annexe). Le nombre indiqué tient compte des deux registres nécessaires pour le nouvel en-tête; si vous prévoyez de créer plus d'un nouveau fichier, soustrayez deux registres pour chaque fichier nouveau supplémentaire. Ce nombre apparaît dans le registre X et engendre une montée dans la pile, sauf si les montées sont inhibées (ce nombre est le même que celui fourni par `EMDIR` lorsque cette fonction prend fin automatiquement).

Vérification de la taille d'un fichier

Pour vérifier le nombre de registres que comprend un fichier (y compris l'en-tête), introduisez le nom du fichier dans le registre alphabétique et exécutez `FLSIZE` (*file size* = taille fichier). Le résultat apparaît dans le registre X, générant une montée dans la pile opérationnelle (sauf si les montées sont inhibées). Vous pouvez vérifier la taille du fichier en cours en effaçant le registre alphabétique et en exécutant `FLSIZE` .

Élimination d'un fichier

Pour éliminer un fichier de la mémoire annexe — à la fois son en-tête et son contenu —, introduisez le nom de ce fichier dans le registre alphabétique et exécutez `PURFL` (*purge file* = élimination fichier). Tous les fichiers en amont dans la mémoire annexe « remontent » pour occuper l'espace laissé libre par le fichier éliminé.

Après que cette fonction ait été exécutée, il n'y a plus de fichier en cours; en conséquence, la fonction suivante opérant sur la mémoire annexe *doit nécessairement* spécifier un nom de fichier.

Opérations sur les fichiers de programme

Les fichiers de programme sont les plus simples de la mémoire annexe. Un fichier de programme comprend deux registres d'en-tête plus une copie exacte d'un programme de la mémoire principale.

Sauvegarde d'un programme

Vous pouvez créer un fichier qui soit une copie d'un programme en mémoire principale en commençant par spécifier dans le registre alphabétique le programme à copier et le nouveau nom du fichier, puis en exécutant `SAVEP` (*save program* = sauvegarde programme). Il y a trois façons de spécifier le programme et le nom du nouveau fichier :

- Identifiez le programme à copier à l'aide d'un des labels globaux de ce programme, faites-le suivre d'une virgule, puis du nom que vous donnez au nouveau fichier.
- Si vous voulez faire du label global le nom du nouveau fichier, vous pouvez n'introduire que le label global dans le registre alphabétique.
- Si le programme à copier est le programme en cours dans la mémoire principale, vous pouvez vous contenter d'introduire une virgule et le nom du nouveau fichier dans le registre alphabétique.

Registre alpha

`label, nom du fichier`

Registre alpha

`label`

Registre alpha

`, nom du fichier`

Si le registre alphabétique contient un label global, **SAVEP** copie le programme qui contient ce label et qui se trouve le plus près de la fin de la mémoire de programmation (c'est-à-dire le dernier énuméré dans le catalogue 1).

Le nom du fichier peut comprendre n'importe quels caractères, sauf des virgules, avec un maximum de sept caractères. Le huitième caractère et les suivants, ou la virgule et les caractères suivants, sont tous ignorés par le HP-41CX.

Remarque : Si le nom du nouveau fichier (quel que soit la manière dont il a été spécifié) est identique à celui d'un fichier du programme existant, ce dernier est écrasé.

Accès à un programme

Deux fonctions, **GETP** (*get program* = accès au programme) et **GETSUB** (*get subroutine* = accès au sous-programme), permettent de copier un programme de la mémoire annexe dans la mémoire principale. Elles placent toutes deux le programme copié en bas de la mémoire de programmation, mais diffèrent à d'autres égards :

- **GETP** *supprime* le programme se trouvant à la fin de la mémoire de programmation et (dans certains cas) fait de la première ligne du programme rappelé la ligne en cours.
- **GETSUB** *n'affecte pas* le programme se trouvant à la fin de la mémoire de programmation et ne modifie jamais la ligne en cours.

Exécution au clavier

- Si vous voulez remplacer le programme se trouvant à la fin de la mémoire de programmation, introduisez le nom du fichier de programme dans le registre alphabétique et exécutez **GETP**. La première ligne du programme rappelé devient la ligne en cours ; appuyez sur **R/S** pour lancer le programme.
- Si vous voulez exécuter le programme rappelé en appuyant sur **R/S** tout en conservant le programme se trouvant à la fin de la mémoire de programmation, appuyez sur **GTQ** **□** **□** avant d'exécuter **GETP** (c'est alors le programme vide créé par **GTQ** **□** **□** qui sera remplacé).
- Si vous voulez conserver le dernier programme, en particulier si vous rappelez plusieurs programmes d'un coup, introduisez le nom du fichier de programme dans le registre alphabétique et exécutez **GETSUB**. La ligne en cours reste la même qu'initialement.

Exécution par programme. `GETP` fonctionne de deux façons différentes, selon que le programme la contenant se trouve ou non en fin de mémoire de programmation; en revanche, `GETSUB` fonctionne de manière identique dans les deux cas.

Il y a donc trois façons de rappeler par programme un autre programme se trouvant en mémoire annexe :

- Si un programme exécute `GETSUB`, le programme rappelé se place en fin de mémoire de programmation, et l'exécution continue avec l'instruction succédant à `GETSUB`.
- Si un programme *ne se trouvant pas* en fin de mémoire de programmation exécute `GETP`, le programme rappelé remplace celui qui se trouve en fin de mémoire de programmation et l'exécution continue avec l'instruction succédant à `GETP`.
- Si un programme se trouvant à la fin de la mémoire de programmation exécute `GETP`, ce programme est remplacé par le programme rappelé et l'exécution se poursuit avec la première ligne du programme rappelé.

Exemples. Les trois possibilités ci-dessus donnent une grande souplesse pour rappeler un programme à partir d'un autre programme, et, souvent, vous aurez le choix entre plusieurs solutions. Dans les exemples ci-dessous, vous avez à choisir une solution plutôt qu'une autre. Supposons que `MAIN` soit un programme en mémoire principale (mais pas à la fin), et que `AA`, `BB` et `CC` soit des programmes en mémoire annexe, appelés par `MAIN` en tant que sous-programmes.

Les quatre premiers exemples illustrent les deux premières possibilités, avec `GETSUB` et `GETP` dans le programme `MAIN`. Pour être sûr que `MAIN` ne se trouve pas à la fin de la mémoire de programmation, vous pouvez appuyer sur `DFD` avant d'exécuter `MAIN` (afin de créer un programme vide en fin de mémoire), ou bien vous pouvez commencer `MAIN` en utilisant `GETSUB` pour rappeler un programme fictif ne contenant qu'un label global et une instruction `END` (le rappel d'un programme fictif vous permettrait aussi de terminer `MAIN` par `PCLPS`, en effaçant le programme fictif et tous les sous-programmes appelés par `MAIN`).

- Si `MAIN` appelle les sous-programmes `AA`, `BB` et `CC`, et s'il y a assez de place en mémoire principale pour ces trois sous-programmes : utilisez `GETSUB` trois fois au début de `MAIN` pour rappeler `AA`, `BB` et `CC`, et appelez chaque sous-programme individuellement à l'aide de `XEQ`.
- S'il n'y a pas assez de place pour les trois sous-programmes à la fois (et si vous ne savez pas à l'avance dans quel ordre `MAIN` va les appeler) : utilisez `GETSUB`, `XEQ`, et `PCLPS` pour appeler chaque sous-programme; ou bien placez un programme vide ou fictif à la fin de la mémoire de programmation puis appelez chaque sous-programme à l'aide de `GETP` et `XEQ`.
- Si `AA` contient un second label global `DD`, et si vous voulez que l'appel du sous-programme commence à `DD` : utilisez `GETSUB` pour rappeler `AA`, puis `XEQ` `DD` pour commencer le sous-programme à partir du label `DD`.

Les deux exemples ci-après illustrent la première et la troisième possibilité, avec utilisation de `GETSUB` et de `GETP` dans le sous-programme `AA`. Supposons que `MAIN` a rappelé `AA` à la fin de la mémoire de programmation.

- Si `AA` effectue un branchement sur `BB`, utilisez `GETP` dans `AA` pour rappeler `BB`. `AA` est alors automatiquement effacé et l'exécution est transférée sur `BB`.

- Si AA rappelle BB en tant que sous-programme, utilisez `[GETSMR]` et `[XEQ]` dans AA pour appeler BB. AA reste ainsi intact en mémoire de programmation, de sorte que l'exécution peut revenir à AA lorsque BB est fini.

Redéfinition automatique des touches

Comme indiqué au chapitre 10, un label global affecté à une touche du clavier personnel contient cette information (uniquement pour les programmes énumérés dans le catalogue 1). Lorsque vous copiez un programme dans la mémoire annexe à l'aide de `[SAVEP]`, toutes les affectations sont sauvegardées, ainsi que les labels globaux affectés. Si le clavier personnel est validé (c'est-à-dire si l'indicateur binaire 27 est armé) lorsque `[GETSUB]` ou `[GETP]` rappelle un programme se trouvant en mémoire annexe, toute affectation stockée avec ce programme sera également validée.

Vérification de la taille d'un programme

Vous pouvez vérifier combien un programme stocké dans un fichier contient de multipléts à l'aide des fonctions `[RCLPT]` ou `[RCLPTA]` (ces fonctions sont initialement utilisées avec les fichiers de données et de texte ; cette application aux fichiers de programme diffère de leur but initial). Ces deux fonctions font apparaître dans le registre X le nombre de multipléts du programme, générant une montée dans la pile opérationnelle, si ces mouvements ne sont pas inhibés.

- Si le fichier de programme n'est pas le fichier en cours, introduisez son nom dans le fichier alphabétique et exécutez `[RCLPTA]`.
- Si le fichier de programme est le fichier en cours, exécutez `[RCLPT]` ou effacez le registre alphabétique et exécutez `[RCLPTA]`.

Création de fichiers de données et de texte

Vous devez expressément affecter un espace en mémoire annexe à chaque fichier de texte ou de données. Avant de créer un fichier, vous pouvez vérifier combien de registres sont encore disponibles en exécutant `[EMDR]` (ou `[CATALOG, 4]` ou `[EMRODM]`). Le nombre apparaissant dans le registre X est le nombre maximum de registres que vous pouvez attribuer à un nouveau fichier.

Création de fichiers de données

Utilisez `[CRFLD]` (*create file/data* = créer fichier/données) pour créer un fichier de données.

1. Introduisez le nom du nouveau fichier dans le registre alphabétique. Il peut comprendre n'importe quel caractère, sauf des virgules, avec un maximum de sept caractères.
2. Introduisez dans le registre X le nombre de registres que vous voulez attribuer au fichier (sans compter les registres nécessaires à l'en-tête).
3. Exécutez `[CRFLD]`.

Le nouveau fichier est alors le fichier en cours, et son pointeur est positionné sur le premier registre.

Création de fichiers de texte

Pour créer des fichiers de texte, vous procédez comme pour les fichiers de données, mais vous devez d'abord traduire le nombre de registres les besoins en mémoire de votre fichier. Une estimation approximative est généralement suffisante car vous pouvez modifier ultérieurement la taille du fichier ; la formule indiquée ci-dessous, qui permet de calculer le nombre exact de registres requis, peut cependant être utile lorsque le fichier est créé par programme.

Le fichier de texte ci-dessous comprend trois enregistrements : ABC, ABCDEFGHI, et ABCDEFG. Au début de chaque enregistrement se trouve un multiplet supplémentaire (comme en crée la fonction `[XYOA]`) qui indique la longueur de cet enregistrement. Un enregistrement peut contenir jusqu'à 254 caractères.

3	A	B	C	9	A	B	
C	D	E	F	G	H	I	
7	A	B	C	D	E	F	
G	*						

Registres d'en-tête. Contiennent les nom, type et taille du fichier, et la valeur du pointeur.

Registres contenant les chaînes de caractères. Le nombre de ces registres constitue la taille du fichier.

Le « * » à la fin du fichier est un multiplet indiquant la fin du contenu (la valeur décimale de ce multiplet est 255 ; c'est pourquoi chaque enregistrement doit avoir une longueur inférieure à 255). Ce multiplet et ceux qui indiquent la longueur des enregistrements vous sont invisibles ; lorsque vous l'utilisez, le fichier ressemble au schéma de la page 215. Ce n'est que lorsque vous créez le fichier que vous devez prendre ces multiplets en considération.

Si vous voulez savoir exactement combien d'enregistrements et de caractères comprendra le fichier, vous pouvez procéder au calcul suivant :

1. Additionnez le nombre d'enregistrements et le nombre de caractères de tous les enregistrements.
2. Ajoutez 1 au résultat obtenu.
3. Divisez le résultat par 7. S'il y a un reste, arrondissez à l'entier supérieur.

Utilisez ensuite `[CRFLAS]` (*create file/ASCII* = créer fichier ASCII) pour créer le fichier.

4. Introduisez le nom du nouveau fichier dans le registre alphabétique. Ce nombre peut comprendre n'importe quel caractère sauf des virgules, et peut totaliser un maximum de sept caractères.
5. Introduisez dans le registre X le nombre de registres à attribuer au fichier.
6. Exécutez `[CRFLAS]`.

Le nouveau fichier est désormais le fichier en cours, et son pointeur se trouve sur le premier caractère du premier enregistrement.

Modification de la taille des fichiers

Vous pouvez modifier le nombre de registres attribués à un fichier de texte ou de données en exécutant `RESZFL` (*resize file* = modifier taille fichier) :

1. Assurez-vous que le fichier à modifier est bien le fichier en cours.
2. Introduisez dans le registre X le nombre de registres à attribuer au fichier.
3. Exécutez `RESZFL`.

La partie entière du nombre se trouvant dans le registre X devient la nouvelle taille du fichier en cours. La position du pointeur n'est pas modifiée.

Augmentation de la taille du fichier. Avant d'agrandir un fichier, vous pouvez vérifier combien de registres sont encore disponibles en exécutant `EMDIR` ou `EMROOM`. Vous pouvez généralement compter sur deux registres supplémentaires par rapport au nombre indiqué, car `EMDIR` et `EMROOM` réservent automatiquement deux registres pour un nouvel en-tête, qui n'est pas nécessaire dans ce cas. Toutefois, si le nombre affiché est zéro, cela peut signifier qu'il reste zéro, un, ou deux registres.

Réduction de la taille d'un fichier. Si vous réduisez de n registres la taille d'un registre, les n derniers registres du fichier sont perdus. Lorsque le registre X contient un nombre *positif*, `RESZFL` ne peut s'exécuter que si aucun registre utilisé n'est perdu. Un registre est considéré comme utilisé si :

- Il contient un nombre différent de zéro (fichiers de données).
- Il contient des caractères ou le multiplète de fin de fichier (fichiers de texte).

Si le nombre contenu dans le registre X est positif et si des registres utilisés vont être perdus, `RESZFL` entraîne l'apparition du message `FL SIZE ERR` (erreur de définition de la taille du fichier). Avec des fichiers de données, vous pouvez contourner cette protection en introduisant un nombre *négalif* dans le registre X. Les n derniers registres sont alors perdus, quel que soit leur contenu.

Effacement de fichiers

Vous pouvez effacer le contenu d'un fichier de données ou de texte sans éliminer l'en-tête en exécutant `CLFL` (*clear file* = effacement fichier) comme suit :

1. Introduisez le nom du fichier dans le registre alphabétique.
2. Exécutez `CLFL`.

Le fichier effacé devient le fichier en cours. S'il s'agit d'un fichier de données, tous les registres contiennent alors des zéros et le pointeur se positionne sur le premier registre. S'il s'agit d'un fichier de texte, il n'existe pas d'enregistrement et le pointeur se trouve sur le premier caractère du premier enregistrement à ajouter.

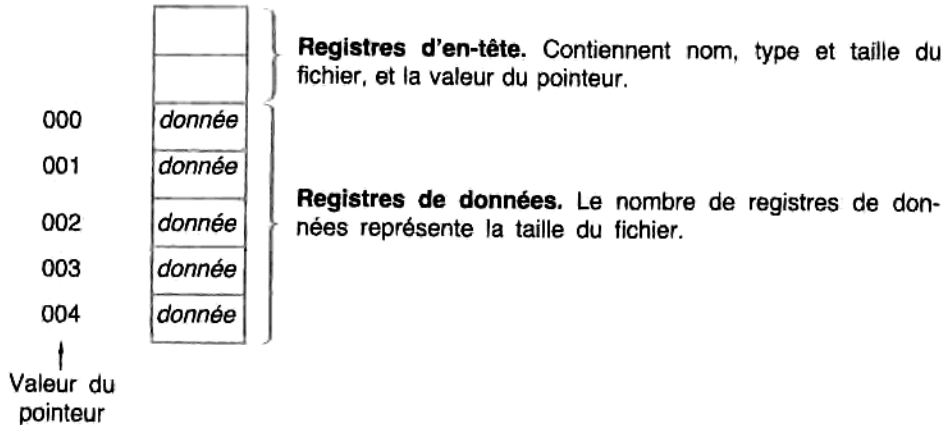
Pointeurs des fichiers de données et de texte

Les fichiers de données et de texte sont respectivement des ensembles de registres et de chaînes alphabétiques, avec des pointeurs vous permettant d'accéder à des éléments individuels dans ces ensembles.

Structure des fichiers de données

Un fichier de données comprend deux registres d'en-tête et un ou plusieurs registres de données. L'en-tête contient la valeur numérique de la position du pointeur, qui indique le prochain registre auquel vous allez accéder.

Le schéma ci-dessous montre un fichier de données de cinq registres. Vous remarquerez que les valeurs du pointeur commencent par 000 dans le premier registre et sont ensuite incrémentées d'une unité pour chaque registre suivant.



Structure des fichiers de texte

Un fichier de texte comprend deux registres d'en-tête et un ou plusieurs registres contenant vos enregistrements. Chaque enregistrement est une chaîne pouvant comprendre jusqu'à 254 caractères. Les enregistrements peuvent donc être de longueurs différentes, et le pointeur d'un fichier de texte comprend deux composantes (contenues dans l'en-tête):

Pointeur d'enregistrement. C'est la partie entière du pointeur. Sa valeur représente l'enregistrement en cours.

Pointeur de caractère. C'est la partie décimale du pointeur. Sa valeur représente le caractère en cours dans l'enregistrement en cours. Les fonctions s'appliquant sur l'ensemble d'un enregistrement ne prennent pas en compte cette partie du pointeur.

Un pointeur de fichier de texte se présente sous la forme *eee.ccc*, où *eee* est le pointeur d'enregistrement et *ccc* le pointeur de caractère. Dans le schéma ci-après, qui représente un fichier de texte de trois enregistrements, les valeurs du pointeur pour les trois A sont respectivement 000.000, 001.000, et 002.000. Les valeurs du pointeur pour H et I sont 001.007 et 001.008 (ce schéma montre comment interpréter les pointeurs de fichiers de texte. Le schéma de la page 212 représente le même fichier, tel qu'il serait réellement stocké dans les registres).

Valeur du pointeur
d'enregistrement

000	A	B	C						
001	A	B	C	D	E	F	G	H	I
002	A	B	C	D	E	F	F		
	↑	↑	↑	↑	↑	↑	↑	↑	↑
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8

} Valeur du pointeur de caractère

Opérations sur les pointeurs

Les pointeurs vous permettent d'accéder à des éléments individuels au sein des fichiers de données et de texte. La valeur numérique d'un pointeur représente le registre en cours, ou l'enregistrement et le caractère en cours; vous devez positionner le pointeur *avant* d'exécuter la fonction qui accède au registre, enregistrement ou caractère voulu.

Le plus souvent, vous définissez explicitement un pointeur lorsque vous accédez pour la première fois à un fichier, préparant ainsi les fonctions ultérieures s'adressant aux données indiquées. La plupart des fonctions utilisant des pointeurs s'accompagnent généralement d'une progression automatique du pointeur. Vous pouvez ainsi accéder à des éléments successifs en répétant simplement la fonction.

Parfois, la valeur du pointeur ne correspond à aucune information stockée. Par exemple, si une fonction accède au dernier registre d'un fichier de données, puis fait progresser le pointeur, la nouvelle valeur du pointeur ne correspond pas à un registre dans un fichier. Si vous répétez la fonction, le message **END OF FL** (fin de fichier) apparaîtrait.

Positionnement d'un pointeur. Pour positionner un pointeur, exécutez `SEEKPTA` (*seek pointer by Alfa* = rechercher pointeur via le registre alpha) ou `SEEKPT` (*seek pointer* = rechercher pointeur).

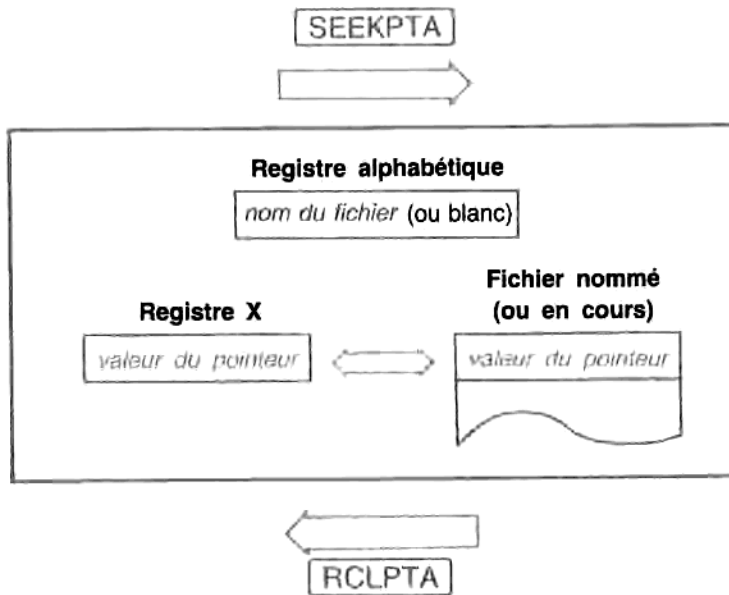
Si le fichier concerné n'est pas le fichier en cours, utilisez `SEEKPTA`.

1. Introduisez dans le registre alphabétique le nom d'un fichier de données ou de texte.
2. Introduisez dans le registre X la valeur du pointeur que vous recherchez.
3. Exécutez `SEEKPTA`.

Le fichier nommé dans le registre alphabétique devient le fichier en cours, et son pointeur interne est positionné sur la valeur se trouvant dans le registre X. Dans un fichier de données, la partie entière de ce nombre représente le pointeur de registre; dans un fichier de texte, elle représente le pointeur d'enregistrement, et les trois premiers chiffres de la partie décimale représentent le pointeur de caractère.

Si le fichier concerné est déjà le fichier en cours, vous pouvez soit:

- Introduire dans le registre X la valeur du pointeur, puis exécuter `SEEKPT`; soit
- Effacer le registre alphabétique, introduire dans le registre X la valeur du pointeur, puis exécuter `SEEKPTA`



Vérification de la valeur d'un pointeur. Vous pouvez vérifier la valeur en cours pour un pointeur à l'aide de `RCLPTA` (*recall pointer by Alpha* = rappel pointeur via Alpha) ou `RCLPT` (*recall pointer* = rappel pointeur). Si le fichier recherché n'est pas le fichier en cours, utilisez `RCLPTA` comme suit :

1. Introduisez le nom du fichier dans le registre alphabétique.
2. Exécutez `RCLPTA`.

Le fichier nommé dans le registre alphabétique devient le fichier en cours. La valeur en cours du pointeur apparaît alors dans le registre X, générant une montée dans la pile opérationnelle (sauf si les montées sont inhibées). Dans un fichier de données, le nombre représente le pointeur de registre. Dans un fichier de texte, le nombre se présente sous la forme *eee.ccc*, *eee* étant le pointeur d'enregistrement et *ccc* le pointeur de caractère.

Vous pouvez vérifier la valeur en cours du pointeur dans le fichier en cours en exécutant `RCLPT` ou en effaçant le registre alphabétique et en exécutant `RCLPTA`. La valeur du pointeur apparaît alors dans le registre X comme indiqué ci-dessus. `RCLPTA` et `RCLPT` peuvent également être utilisés, dans une moindre mesure, avec les fichiers de programme (voir page 211).

Opérations sur les fichiers de données

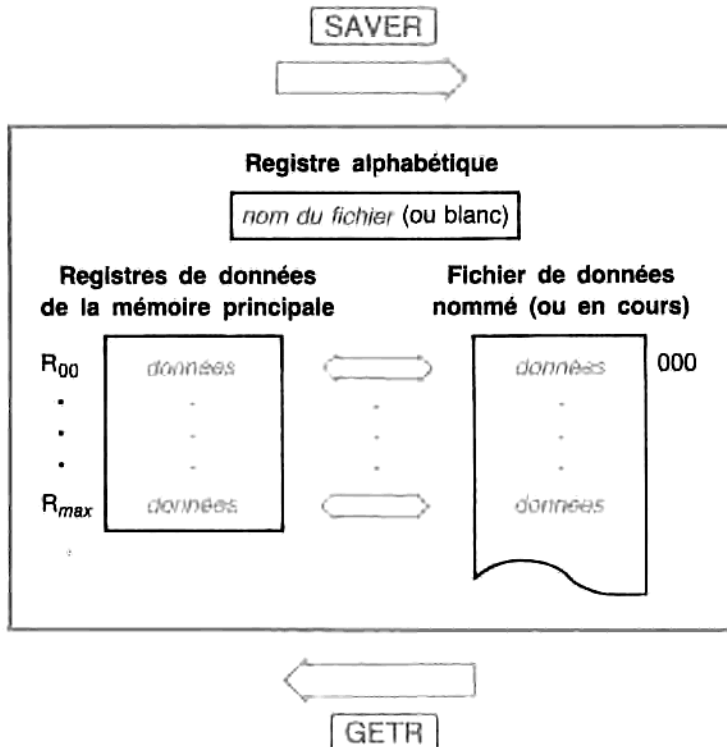
Les fichiers de données servent à la fois d'annexe et de solution de remplacement pour la mémoire principale. Pour les opérations comme l'adressage indirect et l'arithmétique directe dans les registres, vos données doivent se trouver dans la mémoire principale; mais il existe aussi des opérations sur la mémoire annexe, qui ne peuvent pas être reproduites en mémoire principale. Les fonctions qui permettent d'accéder aux registres de données sont décrites ci-dessous, suivies des fonctions qui permettent d'accéder au registre X.

Accès à tous les registres de données

Vous pouvez copier dans un fichier de données tous les registres de données de la mémoire principale avec **SAVER** (*save registers* = sauvegarder registres), et vous pouvez copier dans la mémoire principale tout ou partie d'un fichier de données avec **GETR** (*get registers* = accès registres). Dans les deux cas, les données sont transférées entre des registres ayant des adresses et des pointeurs correspondants : entre R_{00} dans la mémoire principale et le registre 000 dans le fichier de données, entre R_{01} et le registre 001, et ainsi de suite. La valeur du pointeur n'affecte pas ces fonctions, mais elles positionnent toutes deux le pointeur immédiatement après le dernier registre accédé.

Copie de tous les registres de données de la mémoire principale. Pour copier chaque registre de données de la mémoire principale dans le registre correspondant du fichier de données, introduisez le nom du fichier dans le registre alphabétique puis exécutez **SAVER**. Si le fichier recherché est déjà le fichier en cours, vous pouvez simplement effacer le registre alphabétique et exécuter **SAVER**.

S'il y a moins de registres de destination (dans le fichier de données) que de registres sources (en mémoire principale), aucun registre n'est copié et un message d'erreur **END OF FL** (fin de fichier) apparaît. En effet, la fonction **SAVER** suppose que les données de *tous* les registres de la mémoire principale soient sauvegardés.



Copie de données dans tous les registres de données de la mémoire principale. Pour copier les registres d'un fichier de données dans les registres de données correspondants de la mémoire principale, introduisez le nom du fichier dans le registre alphabétique puis exécutez `GETR`. Si le fichier recherché est le fichier en cours, vous pouvez simplement effacer le registre alphabétique, puis exécuter `GETR`.

En commençant par le registre 000 dans le fichier et R_{00} en mémoire principale, `GETR` transfère les données entre registres correspondants jusqu'à ce qu'il n'y ait plus de registres d'un type ou de l'autre (à la différence de `SAVER`, qui entraîne l'apparition d'un message d'erreur s'il n'y a pas un accès à tous les registres de données de la mémoire principale).

Accès à un bloc de registres de données

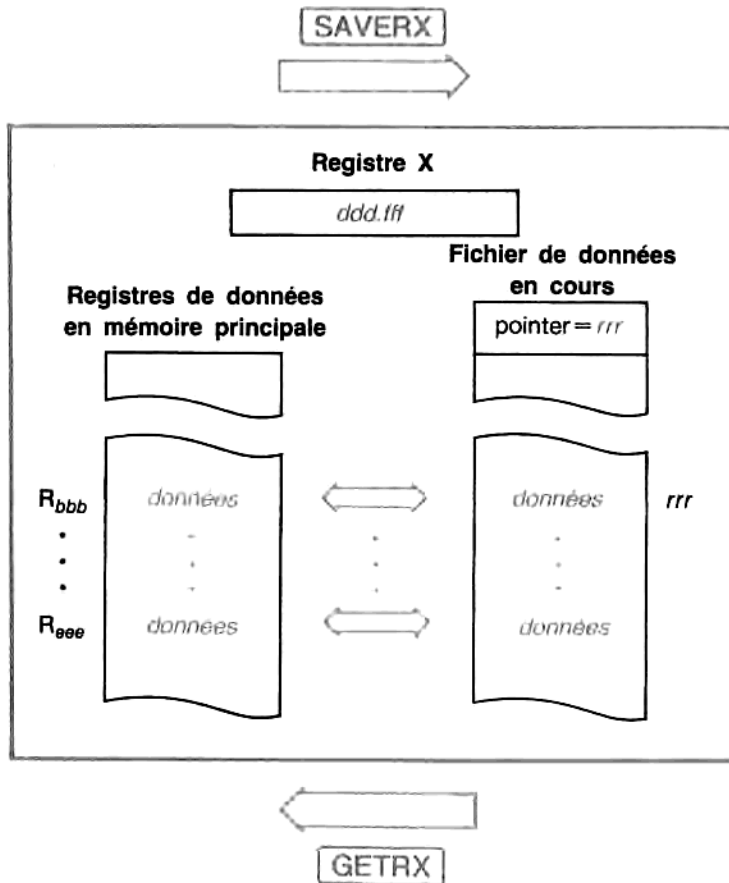
Vous pouvez transférer des données entre un bloc de registres de données en mémoire principale et un bloc de même taille dans un fichier de données grâce à la fonction `SAVERX` (*save registers by X = sauvegarder registres via X*) ou `GETRX` (*get registers by X = accès registres via X*). Le fichier de données doit être le fichier en cours. Les deux blocs de registres sont définis comme suit :

- Le bloc de registres en mémoire principale est défini par un code dans le registre X. Ce code a la forme *ddd.fff*, R_{ddd} étant le registre au début du bloc et R_{ff} celui de la fin du bloc.
- Le bloc de registres du fichier de données est défini par la valeur du pointeur et par le code se trouvant dans le registre X. Le bloc commence par le registre courant et comprend les *fff* — *ddd* registres suivants.

Le transfert de données s'effectue entre R_{ddd} en mémoire principale et le registre courant dans le fichier, entre R_{ddd+1} et le registre suivant dans le fichier de données, et ainsi de suite. `SAVERX` et `GETRX` positionnent toutes deux le pointeur de fichier de données immédiatement après le dernier registre accédé. Si le fichier comprend moins de registres (du registre en cours à la fin du fichier) que le bloc en mémoire principale, aucun registre n'est copié et un message d'erreur **END OF FL** (fin de fichier) apparaît.

Copie de données d'un bloc de registres en mémoire principale. Utilisez `SAVERX` pour copier le contenu d'un bloc de registres de la mémoire principale dans le bloc de registres d'un fichier de données :

1. Assurez-vous que le fichier de destination est le fichier en cours et que son pointeur est positionné sur le premier registre devant recevoir des données.
2. Introduisez le code *ddd.fff* dans le registre X.
3. Exécutez `SAVERX`.



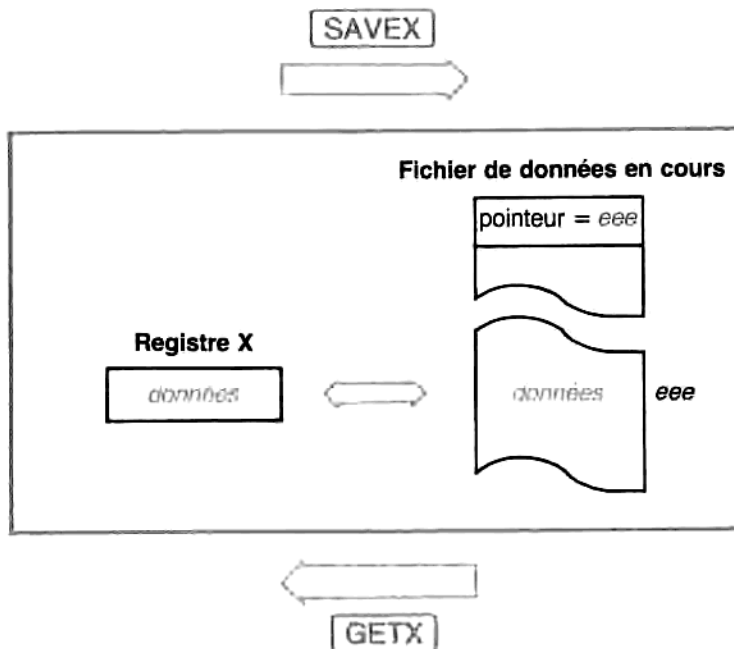
Accès au registre X

Vous pouvez transférer des données entre le registre X et un fichier de données à l'aide de `SAVEX` (*save X* = sauvegarder X) et de `GETX` (*get X* = accès à X). Le fichier de données doit être le fichier en cours, et le registre doit être le registre en cours. Les deux fonctions font progresser le pointeur du fichier de données immédiatement après le registre accédé.

Stockage du contenu du registre X. Utilisez `SAVEX` pour copier le contenu du registre X dans le registre d'un fichier de données :

1. Assurez-vous que le fichier destination est bien le fichier en cours et que le registre destination est le registre en cours.
2. Exécutez `SAVEX`.

Le registre X reste inchangé et le contenu précédent du registre destination est perdu (comme avec `STO`).



Rappel de données dans le registre X. Utilisez `GETX` pour copier le contenu du registre d'un fichier de données dans le registre X :

1. Assurez-vous que le fichier source est le fichier en cours et que le registre source est le registre en cours.
2. Exécutez `GETX`.

Le registre source reste inchangé ; la pile opérationnelle enregistre une montée et le contenu du registre T est perdu, sauf si les montées dans la pile sont inhibées (comme avec **RCL**).

Exemple. Supposons qu'un programme doive sauvegarder le contenu de la pile opérationnelle avant d'utiliser cette dernière pour armer une alarme, puis qu'il doive ensuite rappeler dans la pile son contenu précédent. Cette opération s'effectue en trois étapes, décrites ci-dessous comme des sous-programmes distincts :

- 01 TPILE** Introduit le nom du fichier dans le registre alphabétique.
- 02 4** Introduit le nombre de registres dans le registre X.
- 03 CRFLD** Crée un fichier de données nommé PILE et contenant quatre registres.

Le second sous-programme copie dans le fichier les nombres se trouvant dans la pile opérationnelle ; cette opération s'effectue juste avant d'introduire les paramètres d'alarme dans la pile.

- 01 TPILE** Introduit le nom du fichier dans le registre alphabétique.
- 02 STOL** Sauvegarde le contenu du registre X dans le registre LASTX.
- 03 CLX** Introduit des zéros dans le registre X (sans montée dans la pile).
- 04 SEEKPTA** Fait de PILE le fichier en cours et du registre 000 le registre en cours.
- 05 X< > L** Ramène dans le registre X le contenu sauvegardé dans LASTX (sans montée dans la pile).
- 06 R↑** Les opérations ci-contre tour à tour déplacent le contenu de la pile vers le haut et copient le contenu du registre X dans un registre du fichier PILE. Le nombre *t* (initialement dans le registre T) est copié le premier, puis *z*, *y* et *x*. Le pointeur avance automatiquement, introduisant *t* dans 000, *z* dans 001, *y* dans 002 et *x* dans 003. Cet ordre simplifie le troisième sous-programme, qui ramène ces valeurs dans la pile.
- 07 SAVEX**
- 08 R↑**
- 09 SAVEX**
- 10 R↑**
- 11 SAVEX**
- 12 R↑**
- 13 SAVEX**

Le troisième sous-programme recopie les nombres dans la pile ; il est exécuté après que l'alarme ait été armée.

- 01 TPILE** Introduit le nom du fichier dans le registre alphabétique.
- 02 CLX** Introduit des zéros dans le registre X.
- 03 SEEKPTA** Fait de PILE le fichier en cours et du registre 000 le registre en cours.

04 GETX	Rappelle <i>t</i> dans le registre X, génère une montée dans la pile, et fait avancer le pointeur.
05 GETX	Rappelle <i>z</i> dans le registre X, génère une montée dans la pile, et fait avancer le pointeur.
06 GETX	Rappelle <i>y</i> dans le registre X, génère une montée dans la pile, et fait avancer le pointeur.
07 GETX	Rappelle <i>x</i> dans le registre X, génère une montée dans la pile, et fait avancer le pointeur.

Opérations sur les fichiers de texte

La plupart des opérations présentées ci-après réalisent un transfert de données entre un fichier de texte et le registre alphabétique. N'importe quelle information contenue dans le registre alphabétique — caractères issus du clavier ou multipléts créés par `[XTOA]` — peut être transférée vers et à partir d'un fichier de texte grâce à ces opérations.

Pour les opérations simples, il est plus facile d'utiliser l'éditeur de texte (`[ED]`), décrit au chapitre suivant. Si vous voulez voir un fichier de texte ou le modifier en utilisant des caractères standard, `[ED]` vous permet de voir ce que vous faites au fur et à mesure que vous le faites. En revanche, tout programme manipulant des fichiers de texte *automatiquement* doit utiliser les opérations décrites ci-après.

Vérification du nombre de multipléts disponibles

Avant d'ajouter des données à un fichier de texte, vous pouvez vérifier combien de multipléts sont encore disponibles, grâce à `[ASROOM]` (*ASCII room* = espace pour données ASCII). Le fichier de texte concerné doit être le fichier en cours. `[ASROOM]` fait apparaître dans le registre X le nombre de multipléts disponibles, générant une montée dans la pile (sauf si les mouvements de montée sont inhibés). N'oubliez pas que tout nouvel enregistrement nécessite un multiplétt supplémentaire, pour les besoins de gestion. S'il ne reste plus assez d'espace, reportez-vous à `[RESZPL]`, page 213.

Opérations sur les enregistrements

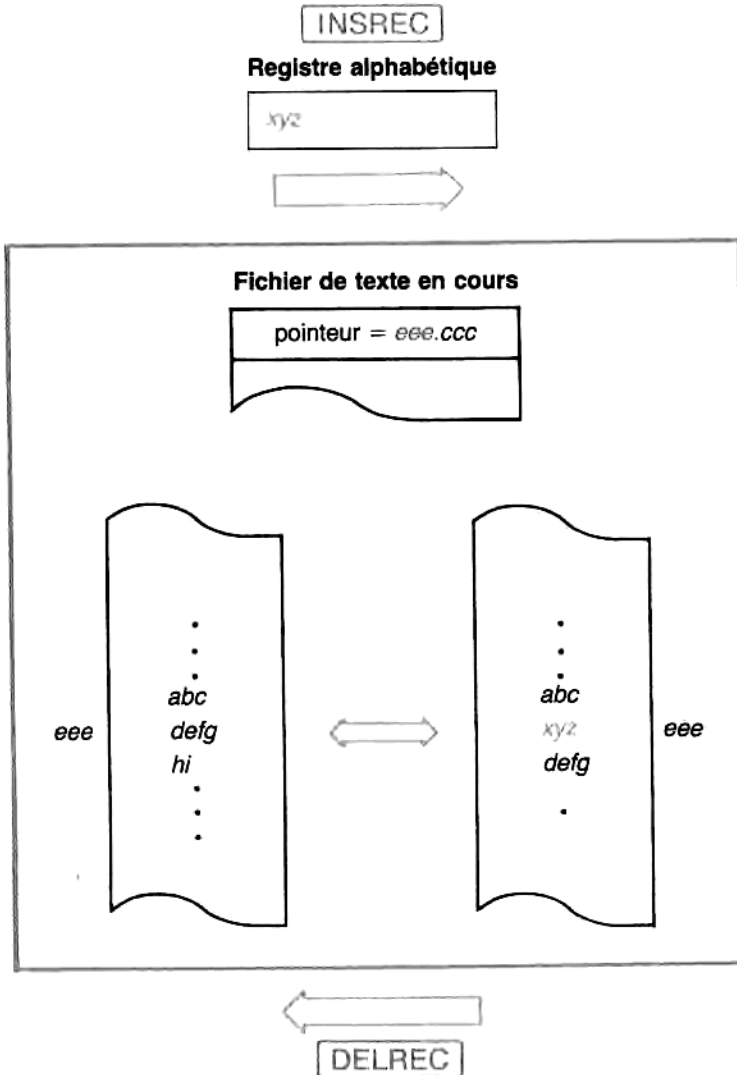
Vous pouvez annexer, insérer, ou supprimer un enregistrement dans un fichier de texte à l'aide de `[APPREC]` (*append record* = annexer enregistrement), `[INSREC]` (*insert record* = insérer enregistrement) ou `[DELREC]` (*delete record* = supprimer enregistrement). Le fichier concerné doit être le fichier en cours. `[APPREC]` et `[INSREC]` font du contenu du registre alphabétique un nouvel enregistrement dans le fichier et positionnent le pointeur immédiatement après le dernier caractère du nouvel enregistrement.

Annexion d'un enregistrement. Utilisez `[APPREC]` pour annexer un nouvel enregistrement :

1. Assurez-vous que le fichier de texte concerné est bien le fichier en cours.
2. Introduisez les données destinées au nouvel enregistrement dans le registre alphabétique.
3. Exécutez `[APPREC]`.

Insertion d'un enregistrement. Utilisez **INSREC** pour insérer un nouvel enregistrement :

1. Assurez-vous que le fichier de texte concerné est bien le fichier en cours et que le pointeur d'enregistrement se trouve à l'emplacement choisi pour le nouvel enregistrement (la valeur du pointeur de caractère n'a pas d'importance).
2. Introduisez les données du nouvel enregistrement dans le registre alphabétique.
3. Exécutez **INSREC**



Suppression d'un enregistrement. Utilisez `[DELREC]` pour supprimer un enregistrement :

1. Assurez-vous que le fichier de texte et l'enregistrement concernés sont bien le fichier et l'enregistrement en cours (la valeur du pointeur de caractère est sans importance).
2. Exécutez `[DELREC]`.

La valeur du pointeur d'enregistrement reste inchangée, mais l'enregistrement en cours est alors celui qui, précédemment, suivait l'enregistrement supprimé. Le pointeur de caractère prend la valeur 000.

Opérations sur les caractères

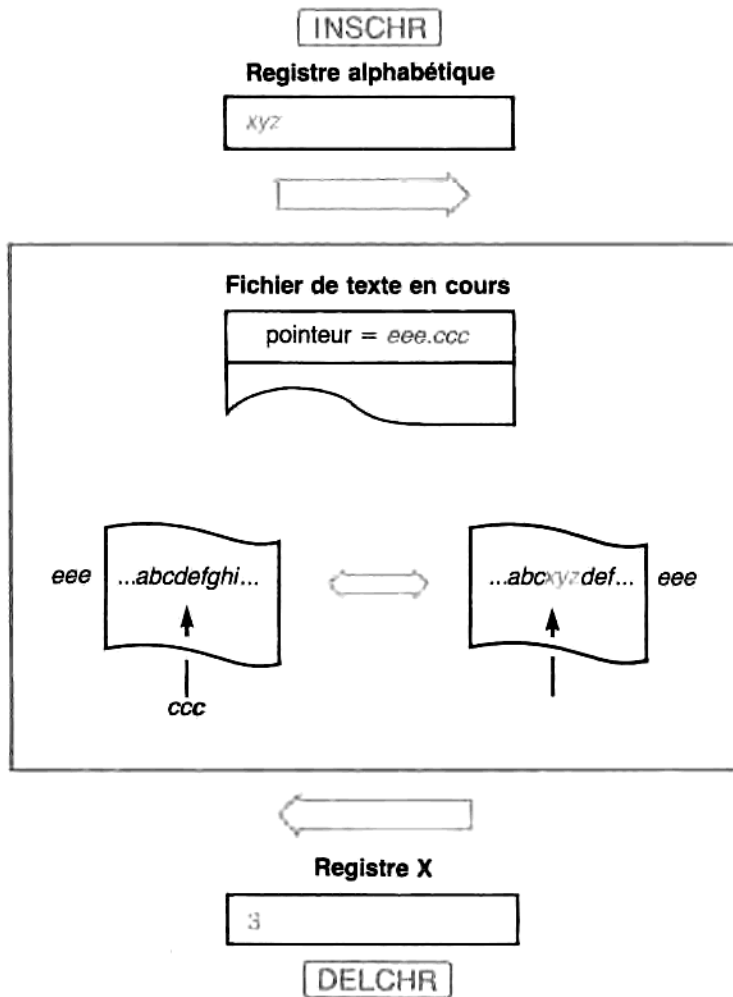
Vous pouvez annexer, insérer, ou supprimer des caractères dans un enregistrement à l'aide de `[APPCHR]` (*append characters* = annexer caractères), `[INSCHR]` (*insert characters* = insérer caractères) ou `[DELCHR]` (*delete characters* = supprimer caractères). L'enregistrement concerné doit être l'enregistrement en cours dans le fichier en cours. `[APPCHR]` et `[INSCHR]` ajoutent le contenu du registre alphabétique à l'enregistrement et positionnent le pointeur après le dernier caractère ajouté. Cela vous permet de répéter ces fonctions immédiatement après l'introduction des nouvelles données dans le registre alphabétique.

Annexion de caractères. Utilisez `[APPCHR]` pour annexer des caractères à un enregistrement :

1. Le fichier de texte et l'enregistrement doivent être le fichier et l'enregistrement en cours (la valeur du pointeur de caractère est sans importance).
2. Introduisez les caractères à annexer dans le registre alphabétique.
3. Exécutez `[APPCHR]`.

Insertion de caractères. Utilisez `[INSCHR]` pour insérer des caractères dans un enregistrement :

1. Le fichier de texte et l'enregistrement doivent être le fichier et l'enregistrement en cours, et le pointeur de caractère doit se trouver à l'endroit de l'insertion.
2. Introduisez dans le registre alphabétique les caractères à insérer.
3. Exécutez `[INSCHR]`.



Suppression de caractères. Utilisez `DELCHR` pour supprimer des caractères dans un enregistrement :

1. Assurez-vous que le fichier de texte et l'enregistrement concernés sont le fichier et l'enregistrement en cours, et que le premier caractère à supprimer est le caractère en cours.
2. Introduisez dans le registre X le nombre de caractères à supprimer.
3. Exécutez `DELCHR`.

La valeur du pointeur de caractère reste inchangée, mais le caractère en cours est celui qui, précédemment, suivait le dernier caractère supprimé.

Recherche dans un fichier

Dans un fichier de texte, vous pouvez rechercher une chaîne de caractères à l'aide de `POSFL` (*position infile* = position dans le fichier), en procédant comme suit :

1. Assurez-vous que le fichier dans lequel doit s'effectuer la recherche est bien le fichier en cours.
2. Placez le pointeur à l'endroit où vous voulez que la recherche commence.
3. Introduisez dans le registre alphabétique la chaîne recherchée.
4. Exécutez `POSFL`.

Si la chaîne recherchée est trouvée :

- Le pointeur se place sur le premier caractère de cette chaîne.
- La valeur de ce pointeur apparaît dans le registre X.

Si la chaîne recherchée n'est pas trouvée :

- Le pointeur de texte reste inchangé.
- Une valeur - 1 apparaît dans le registre X.

Dans les deux cas, le contenu du registre alphabétique n'est pas affecté, et il se produit une montée dans la pile, sauf si les montées sont inhibées. Si la chaîne recherchée a été trouvée, vous pouvez rappeler cette partie du fichier dans le registre alphabétique à l'aide des fonctions ci-dessous.

Copie de données dans le registre alphabétique

`GETREC` (*get recors* = accès enregistrement) et `ARCLREC` (*Alpha recall record* = accès enregistrement avec rappel reg. alpha) copient dans le registre alphabétique les caractères d'un enregistrement. La seule différence est que la première fonction efface préalablement le registre alphabétique.

- `GETREC` efface le registre alphabétique avant d'y copier les caractères.
- `ARCLREC` annexe les caractères copiés au contenu existant du registre alphabétique.

Par ailleurs, les deux fonctions s'exécutent de la même façon.

- Toutes deux commencent à partir du caractère en cours dans l'enregistrement en cours du fichier en cours.
- Toutes deux copient des caractères successifs jusqu'à ce que le dernier caractère de l'enregistrement soit copié ou que le registre alphabétique soit plein (cela signifie que `ARCLREC` n'a aucun résultat si le registre alphabétique est déjà plein).
- Toutes deux désarment l'indicateur binaire 17 si le dernier caractère de l'enregistrement est copié ou arment cet indicateur dans le cas contraire (pour certains périphériques HP-IL, cet indicateur binaire détermine si des sorties successives sont considérées comme des lignes indépendantes ou non).
- Toutes deux positionnent le pointeur de caractère juste après le dernier caractère copié.

Pour exécuter `GETREC` ou `ARCLREC`, assurez-vous que le fichier concerné est le fichier en cours et que le pointeur se trouve à l'endroit voulu, puis exécutez la fonction. Après avoir traité la donnée rappelée dans le registre alphabétique, vous pouvez tester l'indicateur binaire 17 pour vérifier s'il y a encore des caractères à copier dans cet enregistrement.

Accès à des fichiers de stockage de masse

Vous pouvez réaliser des copies permanentes de fichiers de texte avec un dispositif de stockage de masse commandé par un module HP-IL HP 82160A. `SAVEAS` (*save ASCII* = sauvegarder données ASCII) copie dans un dispositif de stockage de masse un fichier de texte se trouvant en mémoire annexe, et `GETAS` (*get ASCII* = accès données ASCII) copie en mémoire annexe le fichier se trouvant dans la mémoire de masse. Les règles ci-dessous s'appliquent aussi bien à la sauvegarde qu'au rappel des fichiers :

- Il faut qu'il existe un fichier destination avant que le fichier source ne soit copié. Pour créer le fichier destination sur le dispositif de stockage de masse, utilisez `CREATE` (fonction dans le module HP-IL) pour créer un fichier de données. Il deviendra un fichier de texte lorsque vous exécuterez `SAVEAS`.
- Si le fichier destination est plus petit que le fichier source, le plus grand nombre de données possible sera copié avant qu'un message d'erreur **END OF FL** (fin de fichier) n'arrête l'exécution du programme. Vous pouvez armer l'indicateur binaire 25 (ne pas tenir compte de la situation d'erreur) si vous voulez une copie incomplète du fichier source.
- Spécifiez les fichiers source et destination dans le registre alphabétique, en les séparant par une virgule.

Registre alphabétique

<i>nom du fichier source, nom du fichier destination</i>
--

Si les deux fichiers ont le même nom, il vous suffit d'introduire ce nom dans le registre alphabétique.

L'éditeur de texte

Table des matières

Introduction	228
L'affichage de l'éditeur de texte	229
La fenêtre	229
Témoins	230
Opérations dans l'éditeur de texte (ED)	230
Validation et invalidation de l'éditeur de texte	230
Conditions par défaut	230
Le clavier	230
Le clavier numérique	232
Les touches de commande des caractères	232
Les touches de commande des enregistrements	232
Erreurs	233
Utilisation de ED dans un programme	233

Introduction

Le présent chapitre décrit la façon d'utiliser l'éditeur de texte, qui permet de travailler avec les fichiers de texte (ou fichiers ASCII). Les explications fournies supposent que vous soyez déjà familiarisé avec la création, l'annulation, et les besoins en mémoire des fichiers de texte (chapitre 13).

Le positionnement du pointeur d'enregistrement/caractère faisait partie de la description des opérations sur les fichiers de texte (comme APPREC, APPCHR, et INSCHR), au chapitre 13. En effet, ces opérations utilisent comme intermédiaire le registre alphabétique, qui permet de transférer jusqu'à 24 caractères entre vous et le fichier de texte. *Mais elles ne vous permettent pas de lire le fichier lui-même.*

L'éditeur de texte, lui, vous permet de visualiser directement les fichiers de texte. Il permet d'introduire, de modifier, et de visualiser directement le contenu des fichiers de texte. Il simplifie ainsi considérablement les opérations d'écriture et de modification de ces fichiers. Il existe cependant deux restrictions : ses fonctions ne peuvent pas être programmées alors que des fonctions comme `[DELCHR]` (*delete character* = suppression caractère) peuvent l'être, et vous ne pouvez pas l'utiliser pour introduire des caractères ne figurant pas sur le clavier alphabétique (alors que vous le pouvez avec des fonctions comme `[APPCHR]` (*Append character* = annexion caractère) ou `[INSCHR]` (*Insert character* = insertion caractère)*. Le fonctionnement complet de l'éditeur de texte est décrit au chapitre 8 « Stockage de texte, données et programmes dans des fichiers ». En conséquence, le présent chapitre est structuré de manière à faciliter son utilisation comme référence, avec des paragraphes très courts. Il y a également une entrée pour `[ED]` dans le tableau de fonctions page 426 (« Fonctions de la mémoire annexe »).

L'affichage de l'éditeur de texte

La fenêtre

Lorsque vous exécutez `[ED]`, l'affichage devient une fenêtre de 12 caractères dans un enregistrement faisant partie du fichier de texte spécifié. Le curseur (soulignement « _ ») apparaît en alternance avec un caractère ou un espace vide. La position du curseur et l'enregistrement affiché dépendent de la valeur du pointeur d'enregistrement/caractère.

Le numéro de l'enregistrement. Par commodité, un numéro d'enregistrement à deux ou trois chiffres apparaît à l'extrémité gauche de l'affichage de l'enregistrement. Ce numéro ne fait pas partie de l'enregistrement lui-même.

L'indicateur d'enregistrement vide ('). Un enregistrement vide est signalé par un numéro d'enregistrement suivi d'un T en exposant, '**. Ce symbole apparaît dans un enregistrement qui vient d'être créé, ainsi que dans un enregistrement qui perd son dernier caractère.

Ponctuation. Les signes de ponctuation (« . », « , » ou « : ») apparaissent à droite des emplacements des caractères normaux. Un signe de ponctuation ne compte pas comme l'un des 12 caractères disponibles, sauf si une ligne comprend deux signes de ponctuation ou plus (les signes de ponctuation adjacents sont séparés par la largeur d'un caractère). Comme le curseur constitue un caractère à part entière (soulignement), il ne peut pas occuper le même emplacement qu'un signe de ponctuation ; de sorte que, lorsque le pointeur d'enregistrement/caractère se trouve sur un signe de ponctuation, ce dernier clignote, sans que le curseur n'apparaisse.

Le curseur. Sauf lorsque l'affichage montre le début ou la fin d'un enregistrement, le curseur et le caractère en cours sont à peu près centrés au milieu de la fenêtre d'affichage. Vous pouvez déplacer le curseur vers la droite jusqu'à un caractère au-delà de l'extrémité de l'enregistrement. C'est sa position lorsque vous ajoutez des caractères à la fin d'un enregistrement.

*`[ED]`, qui valide l'éditeur de texte, est programmable, mais aucune de ses fonctions ne l'est. Cf. « Utilisation de `[ED]` dans un programme », page 233.

**Pour introduire un caractère non standard (comme `@`) dans un fichier de texte, il faut d'abord l'introduire dans le registre alphabétique avec `[XTOA]` (*X to A* = X vers A), puis dans le fichier de texte avec `[APPCHR]` ou `[INSCHR]`.

***L'indicateur d'enregistrement vide est en fait un caractère fictif : il est identique à un caractère, mais il ne peut pas être effacé lorsqu'il signale un enregistrement vide (et qui ne contient donc que cet indicateur). Le `^` utilise un octet de mémoire (comme un caractère) si bien qu'un enregistrement vide ne l'est pas *au sens strict* du terme. (La fonction `[ASROOM]` comptera deux octets par enregistrement vide : un pour la prise en charge, l'autre pour l'indicateur d'enregistrement vide.)

Témoins

Pour l'éditeur de texte, les témoins apparaissant sur l'affichage sont redéfinis :

- **1** indique que le mode Insertion est validé (page 232).
- **ALPHA** indique que le clavier alphabétique normal (avec les touches de commande de l'éditeur) est validé ; l'absence de ce témoin signifie que le clavier numérique est validé (page 232).
- **SHIFT** et **BAT** ont leur signification habituelle.
- Tous les autres témoins sont invalidés, indépendamment de leur état au moment où l'éditeur a été validé. Lorsque l'éditeur est invalidé, ils reviennent à l'état dans lequel ils se trouvaient avant l'utilisation de l'éditeur.

Opérations dans l'éditeur de texte (**ED**)

Validation et invalidation de l'éditeur de texte

Pour accéder à un fichier de texte avec l'éditeur de texte :

1. S'il n'existe pas déjà, créez un fichier à l'aide de **CRFLAS** , en spécifiant son nom et sa taille (voir page 212).
2. Une fois le nom du fichier dans le registre alphabétique, exécutez **ED** (éditeur). Si le registre alphabétique est vide, vous accédez au fichier en cours.
3. Sur l'affichage, le curseur apparaît sur le caractère en cours dans l'enregistrement en cours (déterminé par le pointeur d'enregistrement/caractère). Un nouveau fichier commence par un enregistrement vide (numéroté 000).

Pour invalider l'éditeur de texte, appuyez sur la touche **ON** (**EXIT** sur le clavier de l'éditeur). L'éditeur de texte est *automatiquement* invalidé si vous essayez d'introduire plus de 254 caractères (la limite) dans un enregistrement, ou après quelques minutes d'inutilisation. Lorsque l'éditeur de texte est invalidé, tous les témoins redéfinis par **ED** reviennent à leur état antérieur.

Conditions par défaut

Chaque fois que vous validez manuellement l'éditeur de texte, le clavier alphabétique est validé (en remplacement du clavier numérique), avec affichage du témoin **ALPHA**, et le mode remplacement est validé (les nouveaux caractères « écrasent » les caractères existants). Si la fonction **ED** est exécutée à partir d'un programme, l'indicateur binaire Alpha (48) reste dans le même état qu'avant l'exécution de **ED** afin que le programme puisse vérifier quel clavier est validé.

Le clavier

Vous trouverez page 231 une illustration annotée du clavier de l'éditeur de texte. Ce clavier est aussi schématiquement représenté au dos de votre HP-41CX et dans l'Aide-mémoire. Ce clavier comprend deux types de touches : les touches des caractères alphabétiques (y compris les chiffres), et les touches de commande, qui donnent accès aux fonctions de modification.

Le clavier de l'éditeur de texte

Indicateur d'enregistrement vide

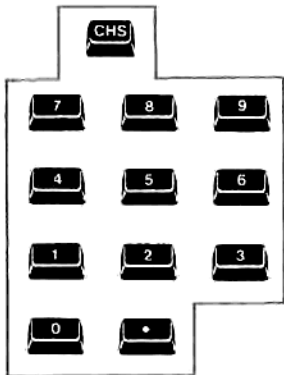
Numéro d'enregistrement
(*eee* dans le pointeur *eee.ccc*)

Le curseur indique le caractère
(*ccc* dans le pointeur *eee.ccc*)

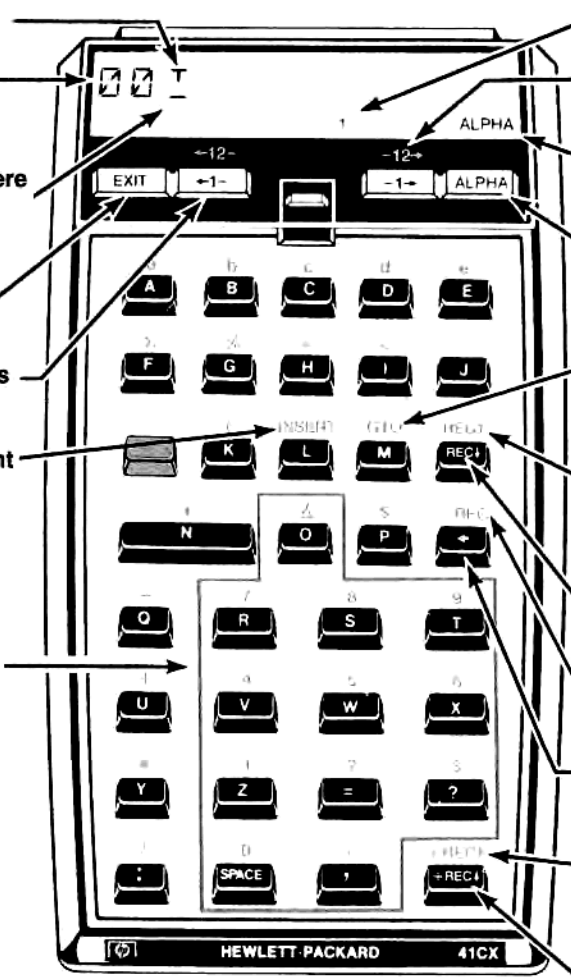
Invalidation de l'éditeur de texte

Déplacement du curseur vers la gauche

Mode insertion/remplacement
(fonctionne en bascule)



Clavier numérique
(Témoin **ALPHA** non affiché)



Indicateur du mode insertion

Déplacement du curseur vers la droite

Témoin du clavier alphabétique

Clavier alphabétique/numérique
(fonctionnement en bascule)

Aller à l'enregistrement *nnn*

Aller à l'enregistrement précédent

Aller à l'enregistrement suivant

Effacer enregistrement

Effacer caractère

Insérer nouvel enregistrement avant enregistrement en cours

Insérer nouvel enregistrement après enregistrement en cours

Le jeu de caractère du clavier de l'éditeur de texte correspond au jeu de caractères alphabétiques, y compris les caractères secondaires et les chiffres. C'est le jeu de caractères par défaut lorsque vous exécutez **ED** manuellement.

Le clavier numérique

En utilisant la touche **ALPHA** pour faire disparaître le témoin **ALPHA**, vous validez le clavier numérique, qui est représenté dans l'encart de la page précédente. Ce clavier rend la saisie numérique plus pratique : en effet, les chiffres, le point **.** et la fonction **CHS** deviennent alors des fonctions primaires.* *Tous les autres caractères deviennent inopérants.* Lorsque vous invalidez l'éditeur de texte, le témoin **ALPHA** (indicateur binaire 48) est ramené à son état antérieur.

Les touches de commande des caractères

Les légendes de la figure page 231 identifient toutes les touches de commande.

Commande du curseur. **←1** (**USER**), **1→** (**PRGM**), **←10** (**USER**), et **10→** (**PRGM**). Le curseur représente le pointeur d'enregistrement/caractère. Vous ne pouvez pas déplacer le curseur au-delà du début de l'enregistrement, ni au-delà d'une position de caractère après la fin de l'enregistrement.

La séquence **REC←** **REC←** ou **REC→** **REC→** permet de déplacer instantanément le curseur au début d'un long enregistrement.

Insertion de caractère. **INSERT** (**LBL**) permet de passer du mode remplacement au mode insertion de caractère. Le mode par défaut est le mode remplacement, sans qu'aucun témoin ne soit affiché. En mode insertion, le témoin 1 apparaît.

En mode remplacement (mode par défaut), tout caractère introduit « écrase » (remplace) le caractère indiqué par le curseur. En mode insertion, tout caractère introduit vient s'insérer en avant du curseur.

Suppression de caractère. **↵** supprime le caractère indiqué par le curseur ; les caractères situés à sa droite se décalent vers la gauche pour occuper l'espace libéré. Si le curseur se trouve au-delà du dernier caractère de l'enregistrement, une pression sur **↵** supprime le dernier caractère. S'il ne restait qu'un caractère dans l'enregistrement, l'indicateur d'enregistrement vide, **␣**, apparaît.

Les touches de commande des enregistrements

Insertion d'un enregistrement. **←REC↵** (**VM-W**) et **→REC↵** (**RS**) insèrent un enregistrement vide respectivement avant et après l'enregistrement en cours. L'indicateur d'enregistrement vide, **␣**, apparaît (**←REC↵** et **→REC↵** mettent également fin au mode insertion s'il était validé).

*La touche **.** correspond à « » ou à « » selon la convention choisie pour l'affichage du séparateur décimal (indicateur binaire 28).

Suppression d'enregistrement. **[REC]** (**[CL]**) supprime l'enregistrement en cours. Le curseur (pointeur d'enregistrement/caractère) se déplace sur le premier caractère de l'enregistrement suivant, sauf si l'enregistrement supprimé était le dernier, auquel cas le curseur se déplace sur le premier caractère de l'enregistrement précédent (qui est désormais le dernier).

Déplacement parmi les enregistrements. **[REC+]** (**[SST]**) et **[REC-]** (**[BET]**) déplacent le curseur (pointeur) sur le premier caractère de l'enregistrement suivant ou précédent, respectivement. Le curseur n'ira pas au-delà du premier ni du dernier enregistrement (l'affichage clignote si vous essayez).

[GTO] nnn déplace le curseur sur le premier caractère de l'enregistrement indiqué, qui doit être spécifié par trois chiffres. Tout comme la fonction **[GTO]** normale, c'est une fonction paramétrable qui requiert l'introduction d'un paramètre à trois chiffres.

Pour localiser rapidement le dernier enregistrement, spécifiez un nombre très élevé pour **[GTO]**, 999 par exemple.

Erreurs

NO ROOM (plus d'espace) — accompagné d'un bip : se produit lorsqu'il n'y a plus d'espace mémoire dans le fichier pour permettre l'ajout de caractères ou d'enregistrements. C'est un simple message d'avertissement, qui dure environ 1 seconde. Il n'est suivi d'aucun autre effet. Si vous avez besoin de plus d'espace dans le fichier, affectez-lui des registres supplémentaires à l'aide de **[RESZPL]**, cf. page 213.

REC TOO LONG (enregistrement trop long) : se produit si vous essayez de dépasser la longueur maximale d'un enregistrement (254 caractères). Cela entraîne automatiquement une invalidation de l'éditeur de texte (il vous suffit de ré-exécuter **[ED]** pour vous retrouver au même emplacement qu'avant l'erreur).

FL NOT FOUND (fichier non trouvé) : se produit si vous exécutez **[ED]** alors que le contenu du registre alphabétique ne correspond pas au nom du fichier existant.

NAME ERR (nom erroné) : se produit si vous exécutez **[ED]** alors que le registre alphabétique contient un nom interdit, comme « , » ou sept multiplats d'une valeur décimale de 255 (#####).

FL TYPE ERR (type de fichier erroné) : se produit si vous exécutez **[ED]** alors que le registre alphabétique ne contient pas un fichier de texte.

Utilisation de **[ED]** dans un programme

Comme indiqué au début du présent chapitre, **[ED]** est programmable, bien que ses opérations de commande ne le soient pas. Par conséquent, pour utiliser **[ED]** dans un programme, il faut qu'un utilisateur soit présent pour introduire ou modifier le texte nécessaire, au moment où le programme exécute **[ED]**.

01 'SECRETS
02 ED

Tout ce que vous avez besoin de faire est d'introduire le nom du fichier voulu dans le registre alphabétique et d'exécuter [↵]. L'exécution du programme s'arrête automatiquement, et l'affichage de l'éditeur de texte apparaît (si vous ne donnez pas de réponse dans les quelques minutes qui suivent, l'éditeur de texte sera automatiquement invalidé et l'exécution du programme reprendra).

01 AON
02 'SECRETS
03 0
04 SEEKPTA
05 ED
06 AOFF

Toutefois, il vaut mieux s'assurer également que le clavier alphabétique est validé (| $\overline{A-Z}$ |), s'il doit l'être, et que le curseur (ou pointeur d'enregistrement/caractère) se trouve à l'endroit voulu. Les lignes 06 et 07 placent le pointeur d'enregistrement/caractère sur l'enregistrement 000, caractère 000 (les opérations sur le pointeur sont décrites au chapitre 13).

Un programme qui exécute |0| s'interrompt pour attendre une entrée. Lorsque le texte est introduit ou la modification terminée, appuyez sur [EXIT] pour relancer le programme, et *non pas* sur [RS]. N'oubliez pas que [RS] signifie [+REC ←] pour l'éditeur de texte.

Pour faire automatiquement une modification programmée dans un fichier de texte, vous devez utiliser des fonctions programmables comme |← + + +| et |← → ←| (voir chapitre 13).

Les fonctions d'horloge

Table des matières

Mise à la date et mise à l'heure	237
Mise à l'heure par <code>SETIME</code>	237
Remise à l'heure par <code>T+X</code>	238
Compensation de dérive par <code>CORRECT</code>	238
Affichage de l'heure	238
Format de l'affichage	238
Période (12 ou 24 heures)	239
Heure seule ou heure et date	239
Opérations sur les valeurs horaires	240
Rappel de l'heure actuelle par <code>TIME</code>	240
Adjonction d'un temps au registre ALPHA par <code>ATIME</code> (12 h)	240
Adjonction d'un temps au registre ALPHA par <code>ATIME24</code> (24 h)	241
Opérations sur la date	242
Formats de la date (<code>MDY</code> et <code>DMY</code>)	242
Mise à la date par <code>SETDATE</code>	242
Rappel de la date actuelle par <code>DATE</code>	242
Adjonction d'une date au registre ALPHA par <code>ADATE</code>	242
Calculs de dates	243
Dates valides	243
Addition de jours à une date par <code>DATE+</code>	243
Nombre de jours entre deux dates par <code>DDAYS</code>	244
Jour de la semaine par <code>DOW</code>	244
Limites et erreurs	244
Limites	245
Erreurs	245

Le HP-41 contient une horloge à quartz et un ensemble de fonctions associées. L'horloge fonctionne en permanence, que l'ordinateur soit en marche ou arrêté. Le présent chapitre décrit les fonctions d'horloge en détail, tandis que le chapitre 6 les a présentées plus simplement, et avec des exemples. Les fonctions d'horloge sont résumées au tableau de fonctions intitulé « fonctions temporelles » et plusieurs servent au chapitre 22 « programmes liés au temps ».

Le fonctionnement de l'horloge est indépendant du reste des fonctions de l'ordinateur et de la mémoire permanente. En conséquence, l'effacement de la mémoire permanente *n'affecte pas* l'horloge ni le chronomètre, mais *annule* toutes les alarmes et le format de la date. L'annexe F décrit en détail les effets de l'effacement de la mémoire, de l'interruption et de la baisse de l'alimentation.

L'annexe B contient des renseignements importants sur les alarmes multiples et périmées. L'annexe F concerne la précision de l'horloge et la consommation d'énergie.

Mise à la date et mise à l'heure

1. La fonction **SETIME** sert à la première mise à l'heure, et **SETDATE** à la première mise à la date.
2. La fonction **T+X** sert à la mise à l'heure : elle modifie les données introduites par **SETIME** (erreur ou changement d'heure légale, et éventuellement de date).
3. La fonction **CORRECT** permet de compenser l'éventuelle dérive constatée au bout d'une semaine au moins.

Si la mise à la date ou à l'heure ou leurs modifications ou corrections déclenchent des alarmes, l'ordinateur émet un signal sonore double, et les alarmes concernées sont périmées (cf. chapitre 16).

Mise à l'heure par **SETIME**

La fonction **SETIME** met l'horloge à l'heure avec la précision d'un centième de seconde. Le tableau ci-contre indique le format de l'heure (période de 12 ou 24 heures).

Procédure :

1. Introduisez dans le registre X l'heure selon le format **±HH.MMSSss**.
2. Exécutez la fonction **SETIME**.

La mise à l'heure indiquée dans le registre X est effectuée au moment où vous relâchez la dernière touche de la séquence d'exécution de la fonction **SETIME**. La précision maximale d'une opération manuelle étant de 0,1 seconde, vous pouvez parfaire la précision grâce à la fonction **T+X** décrite ci-après.

Mise à l'heure

Valeur numérique	Heure
0	Minuit
1	1 (a.m.)
2	2 (a.m.)
⋮	⋮
11	11 (a.m.)
12	Midi
13 ou -1	13 = 1 p.m.
14 ou -2	14 = 2 p.m.
⋮	⋮
23 ou -11	23 = 11 p.m.

Remise à l'heure par **T+X**

La fonction **T+X** incrémente ou décrémente l'heure actuelle de la valeur introduite dans le registre X. Elle sert à modifier l'heure actuelle par suite d'un changement d'heure ou de l'inévitable imprécision de la mise à l'heure manuelle. Elle ne sert pas à compenser l'éventuelle dérive de l'horloge, ce qui est le rôle de la fonction **CORRECT**, décrite ci-après.

Procédure :

1. Introduisez dans le registre X la modification (positive ou négative) à ajouter algébriquement à l'heure actuelle dans le format **±HHH.MMSSss**.
2. Exécutez la fonction **T+X**.

Si cette modification fait passer la date à un autre jour, l'affichage de la date se trouve modifié.

Compensation de dérive par **CORRECT**

La fonction **CORRECT** effectuée à la fois une remise à l'heure et un réajustement du facteur de correction (calculé d'après des exécutions précédentes de plusieurs fonctions, dont **SETIME** et **CORRECT**), dont l'objet est de compenser l'éventuelle dérive de l'horloge due aux variations de tension, de température et de fabrication. La fonction **CORRECT** est utilisée avec profit à des intervalles supérieurs à une semaine. L'annexe F donne plus de détails sur la fonction **CORRECT**.

Procédure :




1. Introduisez dans le registre X l'heure exacte selon le format **HH.MMSSss**.
2. Exécutez la fonction **CORRECT**. La mise à l'heure indiquée dans le registre X est effectuée au moment où vous relâchez la dernière touche de la séquence d'exécution de la fonction **CORRECT**.

Affichage de l'heure





- Actionnez **ON** ou exécutez **CLOCK** pour afficher l'heure actuelle. Seule la fonction **CLOCK** est programmable.
- L'action sur les touches **ON** éteint momentanément le HP-41 avant d'afficher l'heure. Ceci remet à zéro les indicateurs binaires 12 à 20 (cf. chapitre 12).
- L'action sur **+** efface l'affichage de l'heure et l'envoie dans le registre X. L'action sur la plupart des autres touches (dont **ON**) annule l'affichage de l'heure et exécute la fonction correspondante.




Le type d'affichage dépend du format choisi (voir ci-dessous).



Remarque : L'affichage de l'heure consomme une énergie électrique supérieure à sa valeur normale. Veuillez consulter à l'annexe G les paragraphes « Consommation » et « Baisse de tension ».

Les indicateurs binaires 12 à 20 (décrits en détail au chapitre 19) sont réinitialisés à chaque remise en marche du HP-41. L'action sur  les réinitialise aussi puisqu'elle éteint momentanément le HP-41 avant d'afficher l'heure. Toutefois,  n'éteint pas le HP-41, et ne réinitialise donc pas les indicateurs binaires 12 à 20; pour plus de commodité, vous pouvez affecter la fonction  à une touche.





Format de l'affichage



L'affichage peut montrer soit heures:minutes:secondes, soit heures:minutes-mois/jour, sur une période de 12 ou 24 heures. Le choix opéré (touches , ,  ou ) est maintenu même si la mémoire permanente est effacée; il n'est pas maintenu en cas d'interruption prolongée de l'alimentation. L'affichage montre par défaut heures:minutes:secondes sur une période de 12 heures, avec l'indication AM (matin) ou PM (soir); vous pouvez opérer des choix différents selon les méthodes décrites ci-après.

Période (12 ou 24 heures). La fonction  fixe la période de l'affichage à 24 heures. La fonction  fixe la période de l'affichage à 12 heures. A noter que ces fonctions ne concernent que l'affichage de l'heure, et non le format des entrées ou sorties de valeurs horaires utilisées dans les calculs (registre X; cf. page 237 le paragraphe « Mise à l'heure par  »).

Heure seule ou heure et date. La fonction  détermine l'affichage de l'heure seule (avec minutes et secondes). La fonction  détermine l'affichage de l'heure (avec minutes) et de la date (dont le format est au choix jour-mois-année ou mois-jour-année; cf. « Formats de la date », page 242). Exemple : affichages possibles de 3 h 15 de l'après-midi du 21 janvier :

Formats d'affichage

Format		
	3:15:00 PM	3:15 PM 01/21
	15:15:00	15:15 01/21

Remarque : L'affichage par , qui ne change que chaque minute, consomme moins d'énergie que l'affichage par , qui change chaque seconde.

Opérations sur les valeurs horaires

TIME place l'heure actuelle dans le registre X. **ATIME** et **ATIME24** convertissent au format d'affichage un nombre contenu dans le registre X et le placent dans le registre ALPHA.

Rappel de l'heure actuelle par **TIME**

La fonction **TIME** place le nombre représentant l'heure actuelle dans le registre X et décale la pile vers le haut (à moins que le décalage soit inhibé). Le nombre est en format 24 heures, avec centièmes de seconde (**HH.MMSSss**). Il comprend 6 chiffres après le point séparant les heures des minutes; vous devez donc actionner **FOx** 6 pour le voir en entier.

Si la fonction **TIME** est exécutée au clavier, l'affichage montre l'heure actuelle dans le format effectif (12 ou 24 heures), et non le registre X. L'action sur **↔** commute l'affichage sur le registre X.

Exemple : affichage de 17 h 05 min 24 s.

Valeurs horaires

Format	TIME (Affichage moment)	↔ (Registre X)
CLK12	5:05:24 PM	17:052400
CLK24	17:05:24	17:052400

Adjonction d'un temps au registre ALPHA par **ATIME** (12 h)

La fonction **ATIME** (*Alpha time* = temps Alpha) sert à ajouter le nombre figurant dans le registre X au contenu du registre ALPHA, dans le format d'affichage en cours : **CLK12** ou **CLK24**. Cette fonction est utile aux programmes utilisant l'heure.

ATIME et **CLK24** (ou **ATIME24**, décrite au paragraphe suivant) sont des fonctions de période 24 heures, utiles pour traiter les temps écoulés. Dans ces cas, il n'y aura ni **AM** ni **PM**, ni altération des nombres compris entre 13 et 23.

La partie entière de x détermine le formatage du nombre pour son adjonction dans le registre ALPHA :

- **ATIME** accepte les nombres dans l'intervalle $-100 < x < 100$. Le signe moins n'est pas pris en compte, sauf de -1 à -11 , auquel cas il signifie « après-midi » (cf. page 237).
- Si $|x| \leq 23$, le nombre apparaît selon le format effectif **CLK12** ou **CLK24**.
- $|x| \geq 24$ permet d'indiquer une durée. N : **AM** ni **PM** ne sont utilisés. Par exemple, 30:12:00 signifie 30 heures 12 minutes.

Exemple (**FIX** 6) :

Adjonction d'un temps dans le registre ALPHA

Valeur dans registre X	Ajoute au registre ALPHA	
	en CLK12	en CLK24
- 11	11:00:00.00 PM	23:00:00.00
- 15.25	3:25:00.00 PM	15:25:00.00
30.125633	30:12:56.33	30:12:56.33

- Le nombre ajouté au registre ALPHA est tronqué (et non arrondi) selon le format effectif de l'affichage :

FIX , SCI , ENG	Format du nombre ajouté	Ex. : 10.0637 (10:06:37 AM en CLK12)
0	<i>HH</i>	10 AM
1 ou 2	<i>HH:MM</i>	10:06 AM
3 ou 4	<i>HH:MMSS</i>	10:06:37 AM
5 ou plus	<i>HH:MMSSss</i>	10:06:37.00 AM

Si la chaîne contenue dans le registre ALPHA est trop longue pour permettre l'adjonction du nombre représentant le temps, la chaîne est décalée à gauche (avec perte des caractères) d'autant de positions qu'il est nécessaire pour placer à droite le nombre en entier.

Si $-100 < x < 100$, l'exécution de **ATIME** provoque une erreur (message **DATA ERROR**).

Adjonction d'un temps au registre ALPHA par **ATIME24** (24 h)

La fonction **ATIME24** est semblable à **ATIME**, sauf que le nombre est exprimé en format 24 heures ou en durée. Le format effectif de l'heure n'a aucun effet sur ce nombre.

Opérations sur la date

Formats de la date (MDY et DMY)

Le format pris par défaut est MDY (*Month-Day-Year* : mois-jour-année). La fonction DMY active le format mois-jour-année. Ce format est le seul concernant l'heure ou la date à être remis à sa valeur par défaut lors de l'effacement de la mémoire permanente.

Le tableau suivant indique comment le HP-41 interprète les nombres représentant les dates que vous avez introduites. Il indique également les différents formats de sortie de la date.

Options de formats de la date

Format	Format d'entrée-sortie du registre X	Sortie de la date sur l'imprimante et sur l'affichage	Indicateur 31
<input type="checkbox"/> MDY mois-jour-année	MM.JJAAAA	MM/JJ ou MM/JJ/AA	Effacé
<input type="checkbox"/> DMY jour-mois-année	JJ.MMAAAA	JJ.MM ou JJ.MM.AA	Armé

Vous pouvez omettre les zéros de tête et de queue des nombres que vous introduisez. Par exemple, le nombre représentant le 6 mai 1990 peut être introduit sous la forme 5.06199 ; les zéros de tête et de queue sont rétablis dans l'affichage de la date.

Mise à la date par SETDATE

Procédure :

- Entrez la date dans le registre X selon le format validé (voir ci-dessus) : **MM.DDYyyy** ou **DD.MMyyyy**. Vous pouvez omettre les zéros de tête et de queue.
- Exécutez SETDATE.

Le HP-41 accepte toute date comprise entre le 1^{er} janvier 1900 et le 31 décembre 2199.

Si le format que vous avez utilisé ne correspond pas au format validé, il n'y aura pas de message d'erreur sauf si votre date est vraiment illégale. Si la mise à la date périmée certaines alarmes, deux signaux sonores retentissent (voir « Alarmes périmées » au chapitre 16).

Rappel de la date actuelle par DATE

La fonction DATE place le nombre représentant la date actuelle dans le registre X et décale la pile vers le haut (à moins que le décalage soit inhibé).

- Le nombre est, selon le format effectif MDY ou DMY, sous la forme **MM.JJAAAA** ou **JJ.MMAAAA**.

- Si la fonction **DATE** est exécutée au clavier, l'affichage montre la date actuelle et le jour de la semaine (JOUR) abrégé en trois lettres sous la forme **MM/JJ/AA JOU** ou **JJ.MM.AA JOU**. A noter que les années 2000 à 2199 sont affichées sur 4 chiffres et non 2 : **MM/JJ/AAAA:JO** ou **JJ.MM.AAAA JOU**.

Appuyez sur **←** pour renvoyer le registre X à l'affichage.

Adjonction d'une date au registre ALPHA par **ADATE**

La fonction **ADATE** ajoute le nombre stocké dans le registre X au contenu du registre ALPHA.

- Le nombre doit être, selon le format effectif, de la forme **MM.JJAAAA** ou **JJ.MMAAAA**. L'intervalle valide est tel que $|x| < 100$; il permet donc de représenter des durées de 99,999999 mois ou jours.
- Le nombre x est converti au format effectif, soit **MM/JJ/AAAA** ou **JJ.MM.AAAA**. Le signe moins n'est pas pris en compte. Les zéros de tête et de queue restent.
- Le nombre ajouté au registre ALPHA est tronqué (et non arrondi) selon le format effectif de l'affichage :

Ajout d'une date au registre ALPHA

FX , ECI , ENG	Format du nombre ajouté	Exemple : 6.11196 (11 juin 1960 en MDY)
0	<i>MM</i> ou <i>JJ</i>	06
1 ou 2	<i>MM/JJ</i> ou <i>JJ.MM</i>	06/11
3 ou 4	<i>MM/JJ/AA</i> ou <i>JJ.MM.AA</i>	06/11/60
5 ou plus	<i>MM/JJ/AAAA</i> ou <i>JJ.MM.AAAA</i>	06/11/1960

Calculs de dates

- **DATE+** ajoute ou soustrait un nombre de jours d'une date et détermine la date résultante.
- **DDAYS** calcule le nombre de jours compris entre deux dates.
- **DOW** détermine le jour de la semaine d'une date donnée.

Dates valides

- L'intervalle de validité des dates pour ces calculs s'étend du 15 octobre 1582 (origine du calendrier grégorien) au 10 septembre 4320.
- Le nombre (**MM.JJAAA** ou **JJ.MMAAA**) doit être positif. S'il y a des chiffres après **AAAA**, ils doivent être des zéros.

Une donnée invalide provoque l'affichage du message d'erreur **DATA ERROR** ou **DATA ERROR Y**. Un résultat invalide à une opération par **DATE+** provoque le message **OUT OF RANGE** (hors limites).

Addition de jours à une date par `DATE+`

La fonction `DATE+` calcule une date à partir d'une date connue, placée dans le registre Y, et du nombre de jours à ajouter ou soustraire, placé dans le registre X.

Procédure :

1. Introduisez la date connue (**MM.JJAAAA** ou **JJ.MMAAAA**).
2. Introduisez le nombre de jours, positif s'il faut l'ajouter, négatif s'il faut le retrancher. Seule sa partie entière est prise en compte.
3. Exécutez la fonction `DATE+`. La pile se décale vers le bas et place le résultat, la date calculée, dans le registre X sous la forme **MM.JJAAA** ou **JJ.MMAAAA**. Le contenu précédent du registre X est sauvegardé dans le registre LAST X.

Nombre de jours entre deux dates par `DDAYS`

La fonction `DDAYS` calcule le nombre de jours qui séparent deux dates se trouvant dans les registres X et Y; elle calcule $x-y$.

Procédure :

1. Introduisez la première date dans le registre Y et la deuxième date dans le registre X sous la forme **MM.JJAAAA** ou **JJ.MMAAAA**. Si la première date est postérieure à la seconde ($y > x$), le résultat est négatif.
2. Exécutez la fonction `DDAYS`. La pile se décale vers le bas et place le résultat dans le registre X. La deuxième date est sauvegardée dans le registre LAST X.

Jour de la semaine par `DOW`

La fonction `DOW` (*Day Of the Week* = jour de la semaine) calcule le jour de la semaine correspondant à la date se trouvant dans le registre X sous la forme **MM.JJAAAA** ou **JJ.MMAAAA**.

Procédure :

1. Introduisez la date.
2. Exécutez la fonction `DOW`.
3. Le résultat est un nombre de 0 (dimanche) à 6 (samedi), contenu dans le registre X. La date est sauvegardée dans le registre LAST X.

Si la fonction est exécutée au clavier, le jour de la semaine apparaît sous forme de trois lettres :

SUN (*sunday*) = dimanche
MON (*monday*) = lundi
TUE (*tuesday*) = mardi
WED (*wednesday*) = mercredi
THU (*thursday*) = jeudi
FRI (*friday*) = vendredi
SAT (*saturday*) = samedi

L'action sur commute le registre X sur l'affichage pour vous donner le numéro du jour de la semaine.

Limites et erreurs

Limites

Toutes les fonctions d'horloge (sauf **ON**) sont programmables.

Heure : minutes (**MM**) et secondes (**SS**), de 00 à 59. Centièmes de seconde (**ss**) de 00 à 99.

Date : du 1^{er} janvier 1900 (valeur prise par défaut) au 31 décembre 2199.

Calculs sur les dates : du 15 octobre 1582 (origine du calendrier grégorien) au 10 septembre 4320.

Affichage : le format 6 est nécessaire pour visualiser l'heure jusqu'au centième de seconde.

Erreurs

Le message **DATA ERROR** (erreur de donnée) apparaît si la donnée n'est pas dans le format approprié ou dans les limites autorisées pour les fonctions **ADATE**, **ATIME**, **ATIME24**, **CORRECT**, **DOW**, **SETAF**, **SETDATE**, **SETIME**, **SETSW**, **T+X**.

Le message **OUT OF RANGE** (hors limites) apparaît si le résultat d'un calcul est en dehors des limites autorisées (fonctions **T+X** et **DATE+**).

Alarmes

Table des matières

Types d'alarmes	247
Alarme-message	247
Alarme de commande avec interruption (↑↑)	248
Alarme de commande sans interruption (↑)	248
Armement des alarmes par <code>XYZALM</code>	250
Paramètres de <code>XYZALM</code>	250
Exemples	251
Rappel en pile des paramètres d'alarme par <code>RCLALM</code>	252
Activation et annulation d'alarmes-messages	253
Le cycle d'activation	253
Procédure d'annulation	254
Alarmes simultanées	255
Catalogue d'alarmes (<code>ALMCAT</code> , <code>CATALOG 5</code>)	255
Effacement d'alarmes de la mémoire	258
Effacement d'une alarme-message en action	258
Effacement d'alarmes de commande avec répétition	259
Alarmes périmées	259
Création d'alarmes périmées	260
Alarmes périmées inactivées	260
Témoin automatique d'alarmes périmées	260
Activation d'alarmes périmées	260
Programmes d'armement d'alarmes	262
Armement d'une alarme par un programme (SETALM)	262
Armement d'une alarme par rapport au temps courant (ALMREL)	263

Les alarmes du HP-41CX ont des fonctions variées. Selon les paramètres que vous fournissez, l'alarme va du simple rappel (signal sonore plus message) au déclenchement à une heure déterminée de l'exécution d'un programme. En aucun cas, une alarme n'interrompt une fonction en cours d'exécution, mais attend la fin de son exécution (y compris une fonction exécutée par un programme) avant de se déclencher*.

Le HP-41 dispose de trois types d'alarmes, dont les armements sont semblables. Une fois armées, les alarmes sont stockées en mémoire principale, avec les programmes et les affectations des touches. Le chapitre 6, « Fonctions d'horloge », constitue une introduction illustrée d'exemples. Le tableau de fonctions les résume sous le même titre.

Un programme dont l'exécution est déclenchée par une alarme peut modifier l'état des indicateurs binaires ou registres de la même manière que tout autre programme. En conséquence, veillez à ce qu'un tel programme restitue les valeurs des données qui peuvent servir dans d'autres calculs.

Types d'alarmes

Le diagramme de la page 249 décrit les trois types d'alarmes. L'alarme-message effectue un simple rappel. L'alarme de commande sert à déclencher l'exécution de programmes (avec ou sans interruption).

Alarme-message

L'alarme-message est utile pour rappeler un rendez-vous. Elle émet une série de tonalités et affiche avec clignotement le message qui a été introduit dans le registre ALPHA au moment de son armement, de 24 caractères de longueur maximale ; à défaut de message spécifique, le HP-41 affiche l'heure et la date.

L'alarme-message se déclenche dès qu'elle échoit, que l'ordinateur soit en fonctionnement, arrêté, ou exécutant un programme. Toutefois, aucune alarme n'interrompt une fonction en cours d'exécution.

L'alarme-message doit être activée, faute de quoi elle devient périmée (gardée en mémoire en tant qu'alarme inactive). Les alarmes périmées sont décrites plus loin dans ce chapitre et à l'annexe B. Le message d'une telle alarme remplace tout affichage en cours, mais n'a aucun effet sur le fonctionnement ou l'état de l'ordinateur, y compris les registres de la pile, de stockage de données, et ALPHA.

* Y compris les fonctions longues, telles que l'affichage de catalogues, les catalogues 4 à 6 à tout instant, l'éditeur de texte **ED** et le chronomètre **SW**. Les alarmes ne se déclenchent pas pendant ces opérations, mais attendent leur achèvement pour le faire.

Alarme de commande avec interruption (††)

Une alarme de commande avec interruption déclenche l'exécution d'un programme ou d'une fonction programmable du catalogue 2, repéré lors de son armement par †† dans le *label global* ou le *nom de la fonction du catalogue 2* dans le registre ALPHA*. Elle exécute un programme en commençant à un label global ou, si aucun label ou nom ne suit les deux flèches, à la ligne courante du programme. A l'échéance, le programme ou la fonction spécifiée est exécuté, *que le HP-41 soit en marche ou arrêté*, dès l'achèvement de toute opération en cours.

Si l'alarme de commande du programme B arrive à échéance alors que le programme A (qui peut être lui-même exécuté par une autre alarme de commande) est en cours d'exécution, l'exécution du programme A est *temporairement* suspendue, tandis que le programme B (ou la fonction du catalogue 2 à laquelle se réfère l'alarme de commande avec interruption) commence à s'exécuter**. Le HP-41 exécute le programme B comme un sous-programme du programme A, en employant un niveau de sous-programme (cf. « Branchements » au chapitre 20).

Une alarme de commande n'a pas besoin d'être annulée ; simplement, elle agit puis s'efface ou est réarmée.

Alarme de commande sans interruption (†)

Une alarme de commande sans interruption déclenche l'exécution d'un programme ou d'une fonction programmable du catalogue 2, repéré lors de son armement par † dans le label global ou le nom de la fonction du catalogue 2 dans le registre ALPHA (*). Elle n'exécute un programme que si le HP-41CX est arrêté ou bien affiche l'heure.

- Si un programme est en cours d'exécution quand l'alarme arrive à échéance, l'alarme fait émettre une double tonalité, et devient périmée.
- Si le HP-41 est sous tension, mais n'exécute aucun programme ni n'affiche l'heure, l'alarme de commande sans interruption devient une alarme-message et n'exécute pas le programme ou fonction spécifié ; comme une alarme-message, elle devient périmée si elle n'est pas annulée.

L'alarme de commande sans interruption est utile si vous voulez qu'un programme soit exécuté à tout moment, excepté si vous effectuez des calculs ou exécutez un programme. L'alarme de commande sans interruption qui commande effectivement un programme n'a pas besoin d'être activée ; l'alarme de commande sans interruption qui devient une alarme-message a besoin d'être annulée.

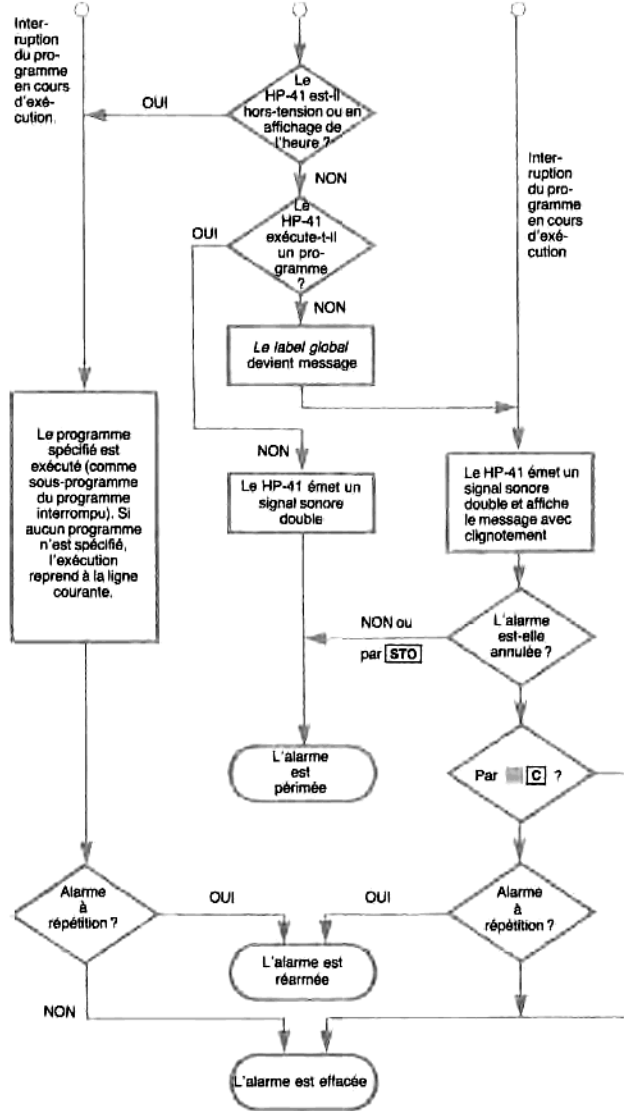
* Veuillez vous reporter à l'annexe I « Catalogue des nouvelles fonctions » page 399 pour les fonctions du catalogue 2.

** L'interruption du programme a lieu à la suite de l'exécution de toute fonction dès que l'alarme arrive à échéance. En outre, l'alarme n'interrompt rien tant que le déplacement de la pile n'est pas autorisé.

Organigramme des alarmes

Le type d'alarme est déterminé par la présence dans le registre ALPHA de 0, 1, ou 2 flèches (↑) lors de l'exécution de XYZALM. Ce tableau indique ce que fait chaque type d'alarme à échéance.

Alarme de commande avec interruption	Alarme de commande sans interruption	Alarme-message
↑↑ label global	↑ label global	message



Quelques remarques :

Une alarme arrivée à échéance est soit **effacée** en mémoire, soit **réarmée** en fonction de son intervalle de répétition, soit **périmée**.

Une alarme de commande avec interruption sera toujours **effacée** (ou **réarmée** si elle est à répétition). Elle ne sera **jamais** périmée.

Une alarme à répétition ne sera **réarmée** que si elle sert à déclencher l'exécution d'un programme ou si elle est annulée. Toute alarme **périmée** ne sera pas **réarmée**.

Si un programme a été interrompu pour une alarme, il reprend son exécution à ce stade.

Armement des alarmes par `XYZALM`

La fonction programmable unique `XYZALM` (alarme X, Y, Z) sert à armer les trois types d'alarmes au moyen des données présentes dans les registres X, Y, Z et ALPHA. Les données temporelles peuvent être spécifiées avec la précision de 0,1 seconde.

Procédure :

1. Chargez les paramètres voulus (cf. ci-dessous) dans les registres X, Y, Z et ALPHA. *Le contenu du registre ALPHA détermine le type d'alarme.*
2. Exécutez la fonction `XYZALM`

Toute donnée illicite en entrée provoque un message d'erreur spécifiant le registre dont il s'agit (**DATA ERROR X**, etc.). Le HP-41 émet une double tonalité dans le cas où l'alarme est armée à une date passée, ou si des alarmes périmées résident en mémoire.

Paramètres de `XYZALM`

T			Alpha	message ou rien	Alarme-message
Z	période de répétition	(HHHH.MMSSs ou 0)	Alpha	↑↑ label global	Alarme de commande avec interrupton
Y	date	(MM.JJAAAA ou JJ.MMAAAA ou 0)	Alpha	↑ label global	Alarme de commande sans interruption
X	heure	(HH.MMSSs)			

Registre Z : période de répétition de l'alarme. $1 \text{ seconde} \leq \text{intervalle} < 10\,000 \text{ heures}$. L'alarme à répétition (intervalle 0) détermine sa prochaine échéance à partir de la première échéance et de l'intervalle de répétition (cf. page 253 le paragraphe « Activation et annulation d'alarmes-messages »). *Pour que l'alarme ne se répète pas, fixez sa période à zéro dans le registre Z.*

Remarque : Il peut être difficile d'annuler une alarme dont la période de répétition est très brève (moins de 10 secondes). Consultez à ce sujet « Effacement d'alarmes de la mémoire », page 258.

Registre Y : date de l'alarme. Intervalle des dates valides : du 1^{er} janvier 1900 au 31 décembre 2199. *Pour la date du jour, fixez Y à zéro.*

Registre X : heure de l'alarme. De zéro à $\pm 23\ 59\ 59$ (0,1 seconde avant minuit). Les nombres négatifs de -1 à -11 signifient « après-midi » (cf. le tableau « mise à l'heure », page 237).

Registre ALPHA :

- 1. Alarme-message.** Chaîne de 24 caractères alphabétiques maximum. En cas de chaîne nulle le HP-41 affiche la date et l'heure à la place.
- 2. Alarme de commande avec interruption.** [] suivi d'un *label global* ou d'un *nom d'une fonction du catalogue 2* ou de rien*. Le label et le nom de fonction sont limités à 6 caractères, bien que les labels puissent avoir 7 caractères dans les programmes ; les caractères en excédent ne sont pas pris en compte, même s'ils apparaissent sur le listage du catalogue**.

L'exécution du programme peut débiter ailleurs qu'à un label global : il suffit de ne taper que deux flèches dans le registre ALPHA. L'exécution commence alors à la ligne courante du programme courant, à échéance. Ceci est utile pour faire reprendre l'exécution à l'endroit où elle s'est arrêtée, sans recourir à l'instruction [OFF].

- 3. Alarme de commande sans interruption.** [] suivi d'un *label global* ou d'un *nom d'une fonction du catalogue 2*. Le label et le nom de fonction sont limités à 7 caractères.

Comme pour l'alarme de commande avec l'interruption, l'exécution du programme peut débiter ailleurs qu'à un label global : il suffit de ne taper qu'une flèche dans le registre ALPHA. L'exécution commence alors à la ligne courante du programme courant, à échéance.

Exemples

T	
Z	0
Y	0
X	1.01

vide
Alpha

0
8.31199
-10

message

0.15
0
11

message

L'exécution de [XYZALM] arme une alarme non répétitive à 1 h 01. Affichage de la date et de l'heure.

Alarme non répétitive à 10 h 00 du soir le 31 août 1990.

Alarme avec répétition toutes les 15 minutes, commençant à 11 h 00 du matin le jour même.

* La touche [] se trouve au-dessus de la touche [N].

** Créez un bref programme d'appel pour un programme dont le nom a 7 lettres ; et armez l'alarme pour commander ce programme d'appel.

T	
Z	0
Y	0
X	1.02

↑↑ TEST
Alpha

Alarme de commande, avec interruption, non répétitive, à 1 h 02, ce jour. appel et exécution du programme TEST, qui interrompt momentanément le programme en cours d'exécution, le cas échéant.

	0
	0
	14.02

↑ TEST

Alarme de commande sans interruption, non répétitive, à 14 h 02, ce jour. Appel et exécution du programme TEST *uniquement si* le HP-41 est arrêté ou bien affiche l'heure.

Rappel en pile des paramètres d'alarme par `RCLALM`

La fonction `RCLALM` (*recall alarm*) rappelle dans les registres X, Y, Z et ALPHA les paramètres de l'alarme spécifiée; un programme peut les y examiner et modifier; vous pouvez aussi les stocker dans d'autres registres ou dans une mémoire de masse. Vous pouvez également effectuer une recopie sélective des alarmes à garder, et effacer toutes les autres par `CLRALMS` (cf. page 258).

L'alarme est spécifiée par la valeur absolue du nombre figurant dans le registre X, qui est son numéro d'ordre dans le catalogue (cf. « Catalogue d'alarmes », page 255). Le numéro doit être compris entre 1 et 253. Un numéro supérieur au nombre des alarmes provoque le message **NO SUCH ALM**, *cette alarme n'existe pas*.

T	
Z	période de répétition
Y	date
X	heure

HH.MMSSss

chaîne de caractères
Alpha

Message ou
↑↑ label global ou
↑ label global

Le format de sortie est le même que le format d'entrée de `XYZALM`, sauf que l'heure est toujours en période de 24 heures **HH.MMSSss**. La disposition jour-mois est celle du format effectif. Le contenu de chaque registre redevient celui qui existait au moment de l'exécution de la fonction `XYZALM` pour cette alarme.

L'exécution de `RCLALM` sauvegarde le numéro d'alarme dans le registre LAST X et décale dans T le contenu du registre Y.

Activation et annulation d'alarmes-messages

Les alarmes-messages, y compris les alarmes de commande sans interruption qui deviennent des alarmes-messages, suivent un cycle d'activation caractéristique. Elles doivent être annulées (description ci-dessous) au cours du cycle d'activation, faute de quoi elles deviennent périmées.

- L'annulation d'une alarme sans répétition l'arrête et l'efface de la mémoire.
- L'annulation d'une alarme avec répétition l'arrête et la réarme pour sa prochaine échéance, définie par l'échéance initiale et non par l'heure de l'annulation.

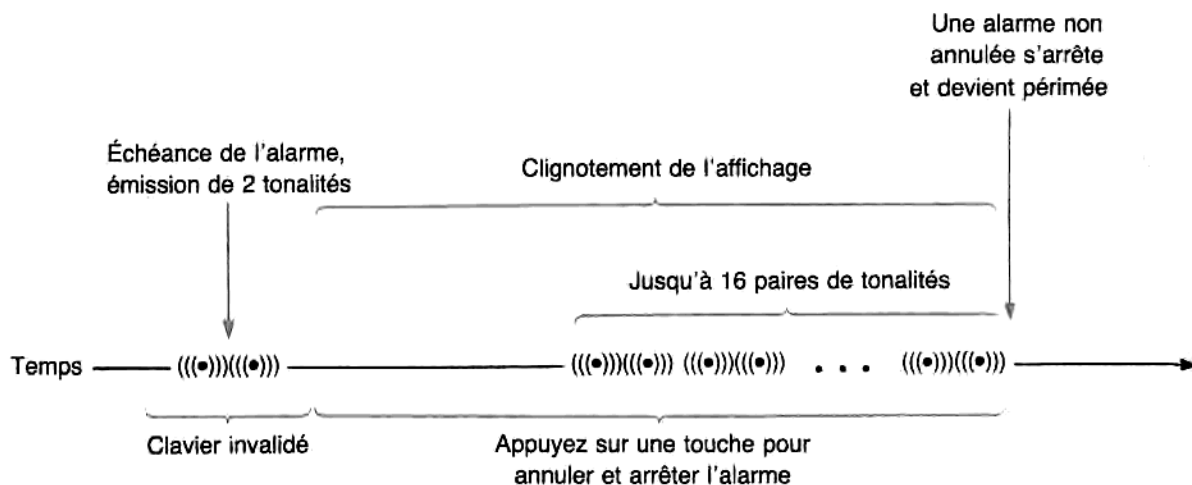
Les alarmes annulées (périmées) restent en mémoire; elles sont automatiquement réactivées quand le HP-41CX est mis hors tension, et émettent une double tonalité dès qu'il est remis sous tension (cf. « Alarmes périmées »).

Les alarmes qui commandent effectivement l'exécution d'un programme s'annulent d'elles-mêmes : elles exécutent simplement le programme ou la fonction du catalogue 2 indiqué, puis s'effacent automatiquement de la mémoire, ou bien se réarment si elles sont à répétition.

Le cycle d'activation

Lorsqu'une alarme-message arrive à échéance, le cycle suivant se déroule :

1. Le HP-41 émet une double tonalité et affiche les 12 premiers caractères de la chaîne alphanumérique ou l'heure et la date si la chaîne est nulle. Une alarme de commande sans interruption devenue alarme-message provoque l'affichage d'une **flèche** et du **label global** ou du **nom de la fonction**. Cette phase dure environ 1 seconde et rend le clavier inopérant.
2. L'affichage commence à clignoter. *En commençant à l'étape 3, vous pouvez annuler l'alarme en appuyant sur presque n'importe quelle touche (cf. paragraphe suivant).*
3. Une fois que l'affichage a clignoté 5 fois, la double tonalité de l'alarme retentit à nouveau jusqu'à 16 fois.
4. Si l'alarme n'est pas annulée à cet instant, elle devient périmée. Si c'est une alarme à répétition, elle n'est pas remise à zéro.



Lorsqu'une alarme-message est annulée, le cycle d'activation figuré ci-dessus s'arrête. Si le message comporte plus de 12 caractères, l'affichage suivant dépend de la méthode d'annulation (cf. ci-après).

Procédure d'annulation

Pendant que l'affichage clignote et avant la longue série de tonalités, vous pouvez annuler l'alarme par l'une des méthodes suivantes. Utilisez les mêmes procédures pour annuler les alarmes périmées automatiquement réactivées.

Résultats de l'annulation de l'alarme

Touche pressée		
← ou ON	STO	Toute autre touche
Arrêt de l'alarme. Effacement immédiat de l'affichage. Effacement de l'alarme**.	Arrêt de l'alarme. L'affichage persiste 3 secondes après le relâchement de la touche*. Une nouvelle pression de la touche prolonge l'affichage de 3 s. Alarme conservée comme alarme périmée.	Arrêt de l'alarme. L'affichage persiste 3 secondes après le relâchement de la touche*. Une nouvelle pression de la touche prolonge l'affichage de 3 s. Effacement de l'alarme**.
* S'il y a plus de 12 caractères, les 12 premiers sont affichés lorsque la touche est maintenue enfoncée, et les suivants durant 3 secondes. ** Si s'agit d'une alarme répétitive, elle est réarmée et non effacée.		

Attendez toujours que le message d'alarme soit effacé avant d'exécuter une autre fonction ; sinon, vous retardez son effacement.

Alarmes répétitives. La nouvelle échéance d'une alarme répétitive est calculée en ajoutant la période de répétition à l'heure de l'alarme, et non à l'heure de l'annulation.

Si vous annulez une alarme répétitive par **[STO]**, celle-ci n'est pas réarmée, mais reste en tant qu'alarme périmée. La période de répétition est cependant maintenue dans le catalogue d'alarme. Si une alarme répétitive périmée est réarmée, elle ne peut se réarmer à une heure passée.

Pour effacer de la mémoire une alarme-message répétitive pendant son cycle d'activation, appuyez sur **[C]**.

Retard d'activation. Si une alarme arrive à échéance alors qu'une précédente alarme-message est active, elle est retardée jusqu'à ce que la précédente alarme soit annulée ou ait terminé son cycle d'activation.

Alarmes simultanées

Si plusieurs alarmes sont armées pour la même heure, elles s'activent dans l'ordre dans lequel elles ont été armées. Toutefois, une alarme de commande interrompt l'exécution d'un programme déclenchée par une autre alarme de commande (avec ou sans interruption). Une alarme-message suspend temporairement l'exécution d'un programme déclenchée par une alarme de commande précédente, mais attend la fin du cycle d'activation de toute autre alarme.

Si plus d'une alarme simultanée ou chevauchante est une alarme de commande, les alarmes successives interrompent l'exécution de programmes déclenchée par les alarmes précédentes, et exécutent leurs propres programmes avant que les programmes précédents puissent s'exécuter. Une alarme de commande sans interruption n'interrompt pas un programme ni n'attend la fin de son exécution ; elle émet simplement une double tonalité et devient périmée. Les alarmes simultanées s'activent dans le même ordre que les alarmes périmées, comme l'explique l'annexe B.

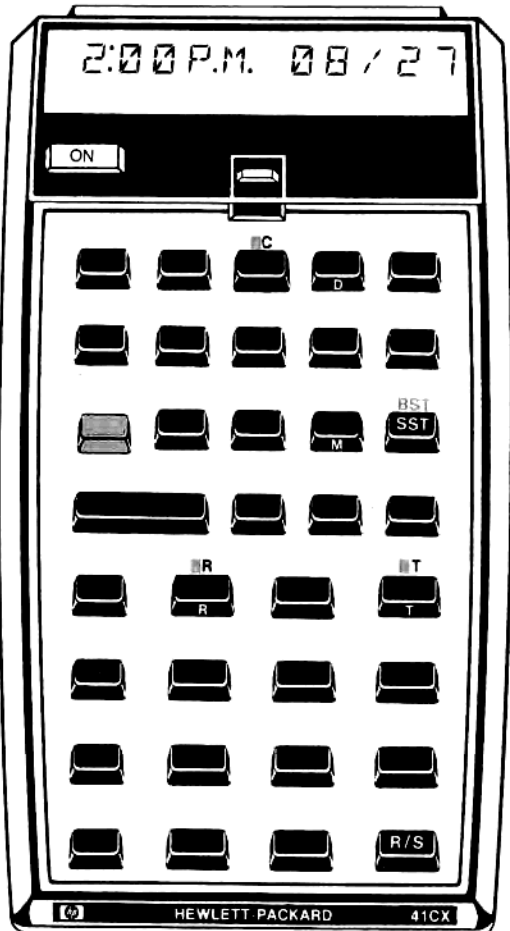
Catalogue d'alarmes (**[ALMCAT]**, **[CATALOG] 5**)

Le catalogue d'alarmes et le clavier associé (celui qui est redéfini pour des opérations spécifiques du catalogue d'alarmes) sont validés soit en exécutant **[ALMCAT]** (catalogue d'alarmes), soit en appuyant sur **[CATALOG] 5**. Seule la fonction **[ALMCAT]** est programmable ; **[CATALOG] 5** est plus rapide à exécuter manuellement.

Caractéristiques du catalogue d'alarmes :

- Fourniture de la liste de toutes les alarmes mémorisées par le HP-41, y compris les alarmes périmées.
- Listage des alarmes par ordre croissant d'heure d'activation ; la place d'une alarme répétitive est actualisée à chaque réarmement.
- Chaque listage d'alarmes indique l'heure et la date de l'alarme et une chaîne alphabétique (message ou label ↑↑ ou ↑).
- Après listage de la dernière alarme, sortie du catalogue d'alarmes et affichage du registre X.

Touches actives sur le clavier du catalogue d'alarmes



- Son exécution valide le déplacement de la pile.
- Comme pour d'autres catalogues, son listage peut être arrêté et relancé par **[R/S]** *. Le listage étant arrêté, vous pouvez progresser en avant ou en arrière au moyen de **[SST]** ou **[BST]**; l'affichage clignote quand vous atteignez le début ou la fin du catalogue.
- Le listage étant arrêté, le clavier est celui du catalogue d'alarmes.

Les touches suivantes opèrent certaines fonctions lorsque le listage du catalogue d'alarmes est arrêté. Les lettres-symboles ne sont pas des lettres du clavier alphabétique; ne passez donc pas en mode ALPHA.

Clavier du catalogue d'alarmes

Touche(s)	Fonction
[T]	Heure d'activation de l'alarme affichée.
[■] [T]	Heure courante.
[D]	Date d'activation de l'alarme affichée.
[R]	Période de répétition de l'alarme affichée.
[■] [R]	Réarmement de l'alarme affichée à sa prochaine échéance définie par sa période de répétition.
[M]	Message ou label ↑↑ ou ↑ (s'ils existent) de l'alarme affichée.
[■] [C]	Effacement de la mémoire de l'alarme affichée.
[←]	Sortie du catalogue d'alarmes; affichage du registre X.
[ON]	Arrêt du HP-41; sortie du catalogue d'alarmes.

Si le HP-41 est arrêté durant le mode catalogue d'alarmes et que vous n'appuyez sur aucune touche pendant environ 2 minutes, il sort automatiquement du mode catalogue d'alarmes.



Bien que la touche **[ALM/CAT]** soit programmable, les opérations au clavier du catalogue des alarmes ne le sont pas. Si **[ALM/CAT]** est exécutée à partir d'un programme, celui-ci reprend quand le système sort du catalogue des alarmes.

* Le chapitre 9 décrit les opérations communes à tous les catalogues. L'appui sur toute touche autre que **[R/S]** et **[ON]** accélère le listage du catalogue.

Effacement d'alarmes de la mémoire



Les alarmes de commande s'effacent automatiquement d'elles-mêmes après action. Les alarmes-messages (y compris les alarmes de commande sans interruption devenues alarmes-messages) sont effacées lors de leur annulation. De plus, le HP-41 dispose de fonctions générales qui permettent d'effacer une ou plusieurs alarmes non échues ou périmées. Enfin, il est assez difficile d'effacer une alarme de commande répétitive de période très courte.






Effacement d'une alarme-message en action




Si une alarme est en train d'agir, le fait de l'annuler l'arrête et l'efface de la mémoire, sauf s'il s'agit d'une alarme répétitive, laquelle est arrêtée et effacée par  .


Effacement


Il y a plusieurs façons d'effacer une alarme non échue ou périmée (c'est-à-dire qui n'agit pas à l'instant présent), ou répétitive.


La plus simple façon est d'utiliser les touches   décrites page précédente; la fonction de ces touches n'est *pas programmable*.


Les fonctions suivantes sont programmables — ,  et .  et  sont particulièrement utiles si vous voulez qu'un programme arme une alarme répétitive, et l'efface ultérieurement.

Effacement de toutes les alarmes par ,  (*clear alarms*) efface toutes les alarmes de la mémoire. Elle sert parfois conjointement avec  (page 252) pour effacer toutes les alarmes et ainsi libérer une partie de la mémoire affectée aux alarmes, *après* avoir rappelé les alarmes individuellement et sauvegardé leurs paramètres.

Effacement d'une alarme par sa chaîne alphabétique.  (*clear alarm by Alpha*) efface la première alarme dont la chaîne alphabétique (message ou label ↑ ou †) correspond à celle placée dans le registre Alpha.

S'il n'existe qu'une seule alarme possédant une chaîne alphabétique déterminée, l'utilisation de  est la plus sûre pour effacer une alarme. S'il y en a plusieurs, seule la première du catalogue est effacée.

Si  est exécutée avec un registre Alpha vide, elle efface la première alarme qui ne comporte aucune chaîne alphabétique.

Si  ne trouve aucune alarme correspondant à la chaîne spécifiée dans le registre Alpha, elle affiche le message **NO SUCH ALM**.

Effacement d'une alarme par son numéro d'ordre. `[CLALMX]` (clear alarm by X) efface l'alarme dont le numéro d'ordre au catalogue est spécifié dans le registre X.

L'effacement d'alarmes de mêmes chaînes alphabétiques ne pose aucun problème. Par ailleurs, souvenez-vous que le numéro d'ordre au catalogue change chaque fois qu'une alarme est armée, réarmée, active ou effacée.

L'argument de `[CLALMX]` est la valeur absolue de la partie entière de x ; par exemple, si $x = 5.1$, `[CLALMX]` efface la cinquième alarme du catalogue. Si $x = 0$ ou $x > 999$, `[CLALMX]` provoque l'affichage du message **DATA ERROR** (erreur de donnée).

Si `[CLALMX]` ne trouve aucune alarme correspondant au nombre spécifié dans le registre X, elle affiche le message **NO SUCH ALM**.

Effacement d'alarmes de commande avec répétition

Une alarme répétitive peut normalement être effacée par les méthodes décrites précédemment. Toutefois, si sa période est brève (environ 10 s; le minimum est 1 s), son effacement peut présenter une difficulté, l'exécution de la manœuvre devant être plus brève que sa période. Deux méthodes sont disponibles :

- Affectation de `[ALMCAT]` à une touche du clavier utilisateur. La manœuvre est réduite à l'appui sur une seule touche. Le mode catalogue d'alarmes n'est pas interrompu par une alarme. ■ `[C]` permet d'effacer l'alarme voulue.
- Utilisation de la prise en compte enchaînée de deux touches pour `[CATALOG] 5` ou une autre fonction d'effacement. En appuyant sur la seconde touche avant de relâcher la première, le clavier du HP-41 reste en fonctionnement sans interruption. L'alarme ne se déclenche pas puisqu'elle attend une interruption entre fonctions ou une pression de touches, et ne peut donc pas se périmer.

Dans un programme, employez `[CLALMA]` ou `[CLALMX]` pour effacer une alarme répétitive.

Alarmes périmées

Une alarme périmée est une alarme présente en mémoire dont l'échéance est antérieure à l'heure actuelle. L'annexe B décrit les règles d'activation automatique d'alarmes périmées multiples.

Création d'alarmes périmées

Une alarme devient périmée normalement si :

- Une alarme-message échoit sans avoir été annulée, ou est annulée par **[STO]**.
- Une alarme de commande sans interruption (†) échoit lors de l'exécution d'un programme.
- Une alarme de commande sans interruption (†) échoit en tant qu'alarme-message (le HP-41 n'étant pas arrêté), et n'est pas ensuite annulée.

A noter qu'une alarme de commande avec interruption ne peut normalement pas se périmiser puisqu'elle s'active toujours à échéance, puis s'efface ou se réarme automatiquement. Elle peut toutefois se périmiser si son heure est passée.

Alarmes périmées inactivées

Une alarme périmée est inactivée si :

- Elle est dépassée par suite d'un changement d'horaire.
- Elle est initialement armée à une heure passée.

Ceci constitue le seul cas d'alarme de commande avec interruption qui peut se périmiser.

Témoin automatique d'alarmes périmées

Le HP-41 émet une double tonalité à chaque mise en marche quand existe au moins une alarme périmée, à titre de témoin ; car les alarmes ne sont pas activées, elles n'ont aucun effet, et ne peuvent être annulées.

Le témoin automatique retentit également si vous effectuez un changement d'heure ou exécutez **[XVZALM]** alors que des alarmes dépassées existent ou se créent.

Activation d'alarmes périmées

Chaque fois que vous arrêtez le HP-41 ou appuyez sur **[ON]** (et non **[CLOCK]**), le HP-41 active automatiquement les alarmes périmées*. L'activation automatique commence avec l'alarme la plus ancienne :

- Les alarmes-messages périmées se déclenchent.
- Les alarmes de commande avec interruption périmées exécutent le programme prévu**.

* Après exécution de **[CLOCK]**, la pression de **[ON]** n'active aucune alarme périmée restante.

** L'activation automatique d'une alarme de commande (avec ou sans interruption) commence par arrêter momentanément le HP-41. C'est pourquoi aucune autre alarme périmée, sauf les alarmes dépassées, ne s'active ensuite, jusqu'à ce que vous appuyez sur **[ON]** ou sur **[ON]** de nouveau (cf. annexe B).

- Une alarme de commande sans interruption périmée s'active automatiquement en tant qu'alarme de commande avec interruption (s'il n'y a pas d'alarmes de commande avec interruption périmées précédentes)*. De cette façon, si la fin d'un programme arrête automatiquement (par l'instruction **[OFF]**), toute alarme de commande sans interruption qui arrive à échéance lors de l'exécution du programme s'exécute automatiquement au moment où le programme arrête le HP-41.

L'annexe B décrit l'activation automatique d'alarmes périmées multiples.

Effacement/réarmement automatique d'alarmes périmées automatiquement activées. Quand une alarme-message périmée est automatiquement activée, son annulation l'efface de la mémoire ou la réarme si elle est à répétition. (**[ON]** arrête mais n'efface ou ne réarme pas une alarme périmée en cours d'activation. Voyez le paragraphe ci-dessous à ce sujet.)

Une alarme de commande qui est automatiquement activée est aussi automatiquement effacée ou réarmée, si elle est à répétition, aux échéances *futures* .

Après annulation d'alarmes-messages périmées, le HP-41 termine la fonction qui a déclenché l'activation automatique; il s'arrête ou affiche l'heure.

Arrêt de l'activation automatique d'alarmes périmées. Si vous appuyez sur **[ON]** pendant le cycle d'activation automatique d'une alarme-message, le cycle complet est arrêté. L'alarme n'est pas annulée. Les alarmes périmées non annulées et non activées restent périmées**.

Ceci est un moyen d'arrêter ce qui pourrait être une longue série d'activations et de reprendre le contrôle du HP-41 tout en sauvegardant les alarmes. Ce moyen est aussi pratique si vous avez par inadvertance avancé l'heure, et que les alarmes futures se périment.

Si vous appuyez sur **[ON]** pendant l'activation d'une alarme de commande, le HP-41 s'arrête, donc stoppe l'exécution de tout programme commandée par une alarme. Ceci déclenche l'activation de toutes les alarmes périmées restantes. L'alarme de commande est effacée ou réarmée, puisqu'elle était activée.

Activation d'une alarme de commande périmée sans interruption. Vous pouvez activer une seule alarme de commande périmée (la plus ancienne en mémoire) en exécutant **[ALMINOW]** (*alarm now*; alarme maintenant).

Lors de l'exécution d'un programme, une alarme de commande sans interruption peut arriver à échéance et donc se périmer. Vous pouvez rechercher et activer une telle alarme (manuellement ou par programme) en exécutant **[ALMINOW]**, pourvu qu'il n'y ait pas en mémoire d'autre alarme de commande périmée. Exécutée par un programme, **[ALMINOW]** opère comme un sous-programme.

* L'activation automatique d'une alarme de commande (avec ou sans interruption) commence par arrêter momentanément le HP-41. C'est pourquoi aucune autre alarme périmée, sauf les alarmes dépassées, ne s'active ensuite, jusqu'à ce que vous appuyiez sur **[ON]** ou sur **[ON]** de nouveau (cf. annexe B).

** L'arrêt du cycle d'activation automatique par **[ON]** provoque aussi l'arrêt de l'activation de toute autre alarme qui arrive à échéance durant ce cycle.

Programmes d'armement d'alarmes

Pour utiliser les programmes suivants, introduisez-les en mémoire manuellement ou au moyen de la version en code à barres présentée à l'annexe J.

Armement d'une alarme par un programme (SETALM)

Le programme SETALM permet d'armer une alarme particulière.

Entrées sur demande par message : heure de l'alarme, message, date, intervalle de répétition éventuel.

Résultats : armement d'une alarme-message ou de commande, avec ou sans interruption.

Procédure :

1. Exécutez SETALM ().
2. En réponse à la question **TIME ?** (heure?), tapez l'heure d'armement dans le format **HH.MMSS**. En l'absence de réponse, le programme se termine. Appuyez sur .
3. En réponse à la question **MESSAGE ?**, tapez le message pour une alarme-message, (label global) pour une alarme de commande avec interruption, (label global) pour une alarme de commande sans interruption. Si pour une alarme-message vous ne tapez aucun message afin que le programme n'affiche que l'heure, appuyez simplement sur .
4. En réponse à la question **DATE ?**, tapez la date d'armement de l'alarme dans le format **MM.JJAAAA** ou **JJ.MMAAAA**. Si vous voulez la date du jour, appuyez simplement sur .
5. En réponse à la question **RESET ?** (répétition?), tapez la période de répétition dans le format **HHHH.MMSS**. Si vous ne voulez pas de répétition, appuyez simplement sur .

L'exécution de ce programme modifie le contenu de la pile et du registre Alpha, et met à 0 l'indicateur binaire (abrégé « i.b. ») 22.

Listage de SETALM.

01 ♦ LBL "SETALM"

02 CF 22

Mise à 0 de l'i.b. 22 (entrée numérique)

03 "TIME?"

Arrêt et demande de l'heure. Si vous ne tapez pas d'heure (i.b. 22 testé à 0), le

04 PROMPT

programme s'arrête. Quand vous avez entré l'heure, appuyez sur pour re-

05 FC?C 22

lancer le programme.

*06 RTN

07 "DATE?"	Stockage dans T du message DATE? et dans Y du message RESET?* .
08 ASTO T	
09 "RESET?"	
10 ASTO Y	
11 "MESSAGE?"	Affichage de MESSAGE? à partir du registre Alpha et arrêt de la saisie alphabétique. Effacement du registre Alpha; maintien de l'affichage de MESSAGE? . Validation du clavier alphabétique. Arrêt. Si vous ne tapez aucun message, le HP-41 affiche l'heure et la date puisque la ligne 13 efface le registre ALPHA.
12 AVIEW	
13 CLA	
14 AON	
15 STOP	
16 AOFF	
17 VIEW T	Affichage de DATE? Arrêt et attente d'entrée*.
18 STOP	
19 FC?C 22	Si vous ne tapez pas de date (pas d'entrée numérique, donc i.b. 22 à 0). La valeur prise par défaut est 0 (date du jour).
20 0	
21 VIEW Z	Affichage de RESET? (venant du registre Y) et arrêt pour saisie de la période de répétition*.
22 STOP	
23 FC?C 22	Si vous ne tapez pas de période de répétition, la valeur prise par défaut est 0.
24 0	
25 X<>Z	Remplacement dans X de l'heure (qui a été déplacée dans Z) et armement de l'alarme*.
26 XYZALM	
27 END	

Armement d'une alarme par rapport courant (ALMREL)

Le programme ALMREL illustre plusieurs techniques de programmation, y compris d'importantes manipulations de la pile; son objet est d'armer une alarme relative à l'heure réelle, et non en spécifiant une heure absolue.

Entrées (sur demande par message) :

1. Délai dans le format **HHHH.MMSS** (ce qui peut représenter plus d'un jour).
2. Chaîne alphabétique de l'alarme (message ou †† ou † et label).

Résultats : alarme-message ou de commande, avec ou sans interruption.

* Les fonctions **LISTO**, **AVIEW** et **X<>** sont des fonctions paramétrées, lors de leur exécution, deux indicateurs de saisie apparaissent : par exemple **X<>---**. Appuyez sur **□Z** pour entrer Z comme paramètre.

Procédure :

1. Exécutez ALMREL.
2. En réponse à la question + **HH.MMSS ?**, tapez le délai le format **HHHH.MMSS**. Appuyez sur **[RS]**.
3. En réponse à la question **MESSAGE ?**, tapez la chaîne alphabétique (éventuellement nulle) de l'alarme puis appuyez sur **[RS]**.

Dès que vous appuyez sur **[RS]** après la question **MESSAGE ?**, le HP-41CX calcule l'heure absolue d'échéance de l'alarme. Le délai est compris entre 0000,0003 (3 secondes) et 9999,595999 (plus de 416 jours). ALMREL rejette une entrée non numérique ou négative du délai (l'indicateur binaire 22 est à 1 après saisie numérique).

L'exécution de ce programme modifie le contenu de la pile et du registre Alpha, et désarme l'indicateur binaire (abrégé « i.b. ») 22.

Listage de ALMREL.**01 ♦ LBL "ALMREL"****02 CF 22**

Désarmement de l'i.b. 22 (entrée numérique).

03 "+ HH.MMSS?"Arrêt et demande du délai. **[RS]** pour redémarrer le programme.**04 PROMPT****05 FC?C 22**

Test d'entrée non numérique. Si i.b. 22 à 0 (entrée non numérique), reprise au début.

06 RTN**07 X<0?**

Test d'entrée négative.

08 RTN

Si nombre négatif, reprise au début.

09 "MESSAGE?"

Arrêt après demande de chaîne alphabétique. Effacement de registre Alpha.

10 AVIEWValidation du clavier alphabétique (AON), puis invalidation (AOFF) au relancement du programme. Si vous voulez un message, tapez-le et appuyez sur **[RS]**; sinon, appuyez simplement sur **[RS]**.**11 CLA****12 AON****13 STOP****14 AOFF****15 TIME**

Rappel de l'heure courante. Addition du délai à l'heure courante.

16 HMS+**17 ENTER †**

Mise du résultat (nouvelle heure) dans X, Y et Z.

18 ENTER †**19**

Calcul du nombre de jours entiers compris dans le délai.

20 /**21 INT****22 DATE**

Rappel de la date courante. Délai en jours dans X; date courante dans Y. Calcul de la nouvelle date de l'alarme (date courante plus délai en jours).

23 X<>Y**24 DATE+**

25 LASTX	Rappel du délai en jours et conversion en heures. Soustraction de ce nombre du
26 24	délai initial en heures dans Z ; résultat : délai en heures après délai en jours,
27 *	c'est-à-dire nouvelle échéance de l'alarme.
28 ST- Z	
29 CLX	Mise de 0 dans X, et recopie dans T. Après décalage, représente dans Z la période
30 STO T	de répétition (nulle). La nouvelle heure passe de Z dans Y ; la nouvelle date passe
31 RDN	de Y dans X.
32 X<>Y	Date dans Y : heure dans X. La pile est prête à armer l'alarme.
33 XYZALM	
34 END	

Fonctionnement du chronomètre

Table des matières

Clavier du chronomètre ([SW])	266
Affichage du chronomètre	267
Fonctionnement du clavier du chronomètre	268
Clavier du chronomètre	268
Les pointeurs de registres	270
Registre courant	270
Modification des pointeurs de registres	270
Prises de temps intermédiaires	271
Affichage lors de la prise de temps intermédiaires	271
Limite d'un pointeur de registre	271
Rappel de temps intermédiaires par [RCL]	271
Examen de différences de temps intermédiaires par [ΔSPLIT]	272
Fonctions programmables du chronomètre	273
Déclenchement et arrêt du chronomètre ([RUNSW] , [STOPSW])	273
Chargement et rappel de l'heure courante par [SETSW] et [RCLSW]	273
Spécification des pointeurs ([SWPT])	274
Chronomètre en décompteur	274
Impression de temps intermédiaires	275
Exemple : programme de chronomètre (SPLITS)	275

La fonction *chronomètre*, **[SW]** ou **[SWPT]**, constitue à elle seule un mode de fonctionnement complet. Elle redéfinit le clavier du HP-41. Les circuits du chronomètre sont distincts de ceux de l'horloge. Le chronomètre peut fonctionner même s'il n'est pas affiché, y compris si le HP-41 est arrêté; il peut continuer à fonctionner librement pendant l'exécution d'autres fonctions ou programmes.

Le chapitre 6 constitue une introduction au chronomètre, illustrée d'exemples. Le tableau de fonctions temporelles résume ses fonctions.

Clavier du chronomètre (**[SW]**)

Le HP-41CX est livré avec une grille qui identifie les fonctions du clavier du chronomètre (cf. page 269, et aide-mémoire). Toute touche non identifiée par cette grille est inactive en mode chronomètre.

- L'exécution de **[SW]** ou **[SWPT]** valide l'affichage du chronomètre et redéfinit les fonctions des touches du clavier. **[SW]** remet à zéro les pointeurs de registres et permet l'affichage de temps intermédiaires ; **[SWPT]** place les pointeurs où vous le désirez (cf. page 274).
- Une pression sur **[EXIT]** invalide le clavier du chronomètre, mais *n'arrête pas automatiquement un chronométrage en cours*. L'affichage se commute sur le registre X. Cette opération n'est pas programmable.
- **[ON]** arrête le HP-41 et invalide le clavier du chronomètre, même si un chronométrage est en cours. **[ON]** *n'arrête pas le chronomètre*.

[SW] et **[SWPT]** sont programmables. Aucune des fonctions du clavier du chronomètre n'est programmable. Ce chapitre décrit quelques fonctions programmables du chronomètre.

Affichage du chronomètre

Quand le clavier du chronomètre est validé, l'affichage a le format suivant :

HH:MM:SS.ss } Rnn

Temps écoulé (avec centièmes de seconde,
qui n'apparaissent qu'à l'arrêt)

Adresse du registre où sera stocké
le prochain temps intermédiaire

Remarque : L'affichage du chronomètre consomme autant d'énergie électrique que l'exécution d'un programme (cf. à l'annexe G « Consommation »). *Si le clavier du chronomètre est actif, le HP-41 ne peut se mettre automatiquement hors tension.*

L'affichage du temps écoulé ne s'efface pas si vous validez et invalidez le clavier du chronomètre, de même qu'un chronométrage en cours n'est pas arrêté si vous invalidez le clavier. Toutefois, la validation du clavier du chronomètre par **[SW]** remet le pointeur de l'affichage à **}-R00**.

Une pression sur **[CLEAR]** remet le chronomètre arrêté à 00:00:00.00. La période du chronomètre est de 100 heures : il repasse automatiquement à 00:00:00.00 après 99:59:59.99. Aucune fonction du chronomètre n'efface les temps intermédiaires stockés, lesquels *remplacent* les nombres se trouvant auparavant dans les registres. Pour remettre un registre à zéro, stockez-y la durée zéro, ou bien en dehors du mode chronomètre, utilisez **[CLRGRK]** pour effacer plus d'un registre.

Une pression sur toute touche non définie sur le clavier du chronomètre arrête l'affichage sans arrêter le chronomètre ; ceci montre des temps écoulés au dixième de seconde. Seul l'arrêt du chronomètre permet l'affichage complet de 8 chiffres.

Fonctionnement du clavier du chronomètre

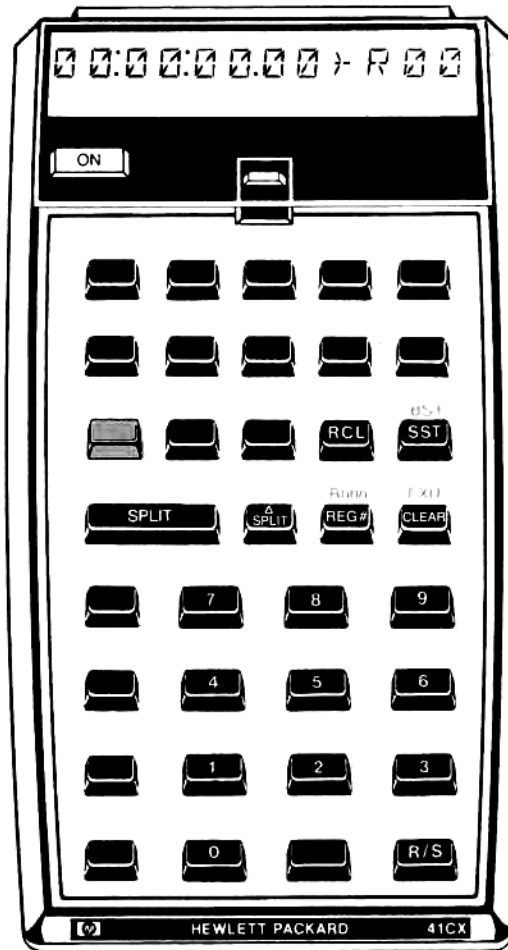
Le clavier du chronomètre est validé par l'exécution de **[SW]** ou **[SWPT]**. De plus, l'exécution de **[SW]** remet à zéro les pointeurs de stockage (**▷**) et de rappel (**◁**), et prend par défaut le mode de stockage normal de temps intermédiaires (**▷ R00**).

Clavier du chronomètre

Touche	Fonction
[R/S]	<i>(Run/Stop)</i> . Déclenche et arrête le chronomètre ; ne remet pas à zéro le chronomètre ou les pointeurs de registres ; annule le mode rappel.
[CLEAR]	Remet à 00:00:00.00 le chronomètre arrêté. Ne modifie pas le pointeur de registre. En mode rappel, [CLEAR] termine ce code mais ne remet pas à zéro le chronomètre.
[EXIT]	Invalide le clavier du chronomètre ; <i>n'arrête pas un chronométrage en cours</i> .
[SPLIT]	Mémorise le temps intermédiaire dans le registre de stockage ▷ Rnn et incrémente le pointeur ; annule le mode rappel.
[RCL]	Touche-bascule entre mode <i>rappel</i> et mode <i>stockage</i> . Le mode <i>rappel</i> affiche le temps intermédiaire stocké dans le registre ◁ Rnn . Le mode <i>rappel</i> est terminé par [R/S] , [SPLIT] et [CLEAR] .
[ΔSPLIT]	Touche-bascule entre le mode différence de temps intermédiaires (D) et le mode temps intermédiaires (R). Affiche la différence entre le temps intermédiaire le plus récent et le précédent. L'affichage change à chaque prise de temps intermédiaire par [SPLIT] . Sert aussi en mode <i>rappel</i> . Ce mode se termine avec la désactivation du clavier.
Numériques*	Spécifient le numéro du registre en y déplaçant le pointeur. Assurez-vous de spécifier 2 ou 3 chiffres, selon le cas.
[SST] *	<i>(Single step)</i> . Incrémente le pointeur de registre sans prendre de temps intermédiaire.
[BST] *	<i>(Back step)</i> . Décrémente le pointeur de registre sans prendre de temps intermédiaire.
[REG#]	<i>Numéro de registre</i> . Supprime/rappelle l'affichage du pointeur de registre ; fonctionne en bascule.
[Rnnn]	Passes l'affichage du pointeur de registre de 2 à 3 chiffres, afin d'afficher les registres 100 à 319 ; fonctionne en bascule.
toute autre	Gèle l'affichage du chronomètre tant qu'elle est enfoncée, sans arrêter le fonctionnement du chronomètre.

* Ces fonctions changent le pointeur de registre, que le chronomètre fonctionne ou soit arrêté.

Clavier du chronomètre



Les pointeurs de registres

Les pointeurs de registres, affichés sous la forme Rnn et Dnn , donnent l'adresse nn du registre courant. R (*regular*) désigne un temps intermédiaire. D désigne la différence entre deux temps intermédiaires. \vdash indique que le temps intermédiaire va être stocké. \doteq indique que le temps intermédiaire (ou la différence) va être affiché.

Prise de temps intermédiaires	
$\vdash Rnn$	Prise de temps intermédiaire
$\vdash Dnn$	Prise de temps intermédiaire ; affichage de la différence
Rappel de temps intermédiaires	
$\doteq Rnn$	Rappel de temps intermédiaire
$\doteq Dnn$	Rappel de différences de temps intermédiaires

Registre courant. $[SW]$ remet toujours à 00 les pointeurs de registres, même si le temps n'est pas remis à zéro. Le registre courant, d'adresse nn , a les fonctions suivantes :

- Dans $\vdash Rnn$ (stockage de temps intermédiaire), nn désigne l'adresse du registre où $[SPLIT]$ va stocker le prochain temps intermédiaire.
- Dans $\vdash Dnn$ (stockage de différences de temps intermédiaires), nn désigne l'adresse du registre où $[SPLIT]$ va stocker le prochain temps intermédiaire et celle du second de deux registres successifs dont la différence des contenus sera calculée et affichée.
- Dans $\doteq Dnn$ (rappel de différences de temps intermédiaires), nn désigne l'adresse du second de deux registres successifs dont la différence de contenus est affichée.

Modification des pointeurs de registres. Le pointeur de registre de stockage (\vdash) s'incrémente automatiquement à chaque prise de temps intermédiaire. Vous pouvez aussi le modifier manuellement au moyen de $[SST]$, $[MET]$, et des touches numériques. Le pointeur de registre de rappel (\doteq) n'est modifiable que manuellement.

Remarque : Les pointeurs des registres de stockage (\vdash) et de rappel (\doteq) sont traités indépendamment (mais sont tous deux remis à 00 par $[SW]$). Les modes R et D sont deux modes différents d'affichage, utilisant le même pointeur de registre.

Pointeur de registre à 3 chiffres ($\{Rnn\}$). Le pointeur de registre passe automatiquement de 2 à 3 chiffres lorsqu'il s'incrémente de 99 à 100, ou manuellement au moyen de $[R000]$. Si vous passez en format 2 chiffres et que le pointeur a une adresse à 3 chiffres, le chiffre des centaines n'est pas affiché. L'affichage d'adresses à 3 chiffres supprime de l'affichage (mais pas de la mémoire) le dernier chiffre à droite du chronomètre.

Suppression de l'affichage du pointeur de registre par $[REG\#]$. La pression sur $[REG\#]$ supprime (et redonne) l'affichage du pointeur de registre, mais ne modifie pas son état. Ceci permet d'observer le dernier chiffre à droite du chronomètre ou un temps négatif dans le cas de l'affichage d'adresses de registres à 3 chiffres.

Prises de temps intermédiaires

Les temps intermédiaires sont mémorisés automatiquement dans des *registres de stockage* consécutifs (les mêmes que ceux où vous stockez toutes sortes de données). Les temps intermédiaires consécutifs sont *cumulés*, à moins que vous ne remettiez le chronomètre à zéro par **[CLEAR]**. Vous pouvez mémoriser autant de temps intermédiaires qu'il y a de registres disponibles. Les temps intermédiaires remplacent dans les registres les données s'y trouvant précédemment; ils peuvent aussi être lus ou modifiés hors mode chronomètre.

Les temps intermédiaires sont stockés sous la forme **HH.MMSSss**, et affichés sous la forme **HH:MM:SS.ss** ou **HH:MM:SS.s**. Inversement, si un registre contient une donnée qui n'est pas un temps et qu'elle est rappelée en mode chronomètre, le HP-41 tente de l'afficher sous la forme **HH:MM:SS.ss**. Si **HH > 99**, le message d'erreur **ERROR : Rnn** apparaît (les groupes **MM** et **SS** supérieurs à 59 ne sont pas cause d'erreur). Pour effacer **ERROR : Rnn**, déplacez le pointeur à un registre contenant un temps intermédiaire lisible. L'appui sur **[CLEAR]** termine le mode rappel.

Le chapitre 6, page 76, présente un diagramme du fonctionnement du chronomètre.

Affichage lors de la prise de temps intermédiaires.

Lorsqu'en mode **R** vous maintenez enfoncée la touche **[SPLIT]**, l'affichage montre le temps intermédiaire courant et le registre dans lequel il est mémorisé; le chronomètre n'est plus affiché mais continue à fonctionner. Lorsque vous relâchez la touche **[SPLIT]**, l'adresse du registre s'incrémente, et l'affichage du chronomètre reprend.

En mode **D**, le fonctionnement de **[SPLIT]** est le même. L'affichage montre la différence entre le temps intermédiaire et le précédent. Le registre dont l'adresse est affichée mémorise le temps intermédiaire qui vient d'être pris (cf. « Examen différences de temps intermédiaires », page suivante).

Limite d'un pointeur de registre

Si le pointeur courant de registre (**Rnn** ou **Dnn**) est déplacé automatiquement ou manuellement (sauf par **[DET]**) au dernier registre disponible ou au-delà, le HP-41 le signale en émettant un bip, que le chronomètre soit en fonctionnement ou non.

De plus, en mode rappel, le déplacement du pointeur à un registre qui n'existe pas invalide le clavier du chronomètre et provoque l'affichage du message **NONEXISTENT**. Le même message apparaît en mode stockage (→) si l'exécution de **[SPLIT]** adresse un registre inexistant.

Rappel de temps intermédiaires par **RCL**

La pression sur **RCL** bascule le chronomètre et l'affichage entre le mode rappel (=) et le mode stockage (+). En mode rappel, l'affichage montre le contenu du registre indiqué, et un chronométrage en cours n'est pas arrêté. Pour examiner les temps intermédiaires stockés dans d'autres registres, déplacez le pointeur de registre au moyen de **SST** , **DSF** ou d'une touche numérique.

Une nouvelle pression sur **RCL**, de même que la pression sur **SPLIT**, **R/S** ou **CLEAR**, réaffiche le chronomètre. Les pointeurs de registres pour stockage de temps intermédiaires et rappel de temps intermédiaires étant traités séparément, lorsque l'affichage repasse du mode stockage (+) au mode rappel (=) ou vice versa, il reprend à l'adresse du pointeur (*nn* ou *mm*) à laquelle il était resté. Ceci vous permet, par exemple, de prendre plusieurs temps intermédiaires, de passer en mode rappel (que le chronomètre soit en marche ou arrêté) pour examiner les valeurs qui viennent d'être stockées, de revenir au mode stockage et de reprendre des temps intermédiaires en commençant par le registre qui suit immédiatement celui dans lequel le dernier temps a été mémorisé.

Examen de différences de temps intermédiaires par **ΔSPLIT**

La pression sur **ΔSPLIT** valide le mode **D**, différence de temps intermédiaires, à l'affichage.

- La valeur affichée est la *différence* entre le temps intermédiaire courant (stocké dans **Dnn**) et celui stocké dans le registre précédent.
- Si le pointeur est sur **D00**, l'affichage montre le temps intermédiaire courant.

Le mode **D** permet l'affichage de la différence entre deux temps intermédiaires sans arrêter le chronomètre. Il est utilisable lors des modes stockage (+ **Dnn**) ou rappel (= **Dnn**).

Vous pouvez comparer rapidement la différence entre deux mesures liées en utilisant le mode **D** par **SPLIT** ou **RCL**. Ce mode vous permet de prendre des temps intermédiaires sans chevauchement, mais trop rapprochés pour vous laisser le temps de remettre le chronomètre à zéro. Vous pouvez par exemple mémoriser à la fois les temps de début et de fin d'une série d'événements rapprochés mais non chevauchants. En rappelant les différences des temps intermédiaires, vous obtenez les durées des événements.

Stockage et rappel. Les différences de temps intermédiaires ne sont ni stockées ni rappelées; elles sont calculées. La pression sur **SPLIT** effectue une prise de temps intermédiaire même si le mode **D** est validé (+ **Dnn**); mais l'affichage, « gelé » au moment où la touche **SPLIT** est enfoncée, *montre* le temps intermédiaire. Le rappel de temps intermédiaires en mode **D** (= **Dnn**) *ne rappelle pas* réellement des différences de temps intermédiaires déjà stockées, mais *affiche* le résultat de calculs effectués à partir de temps intermédiaires stockés.

En conséquence, le mode **D** n'altère pas le fonctionnement de **SPLIT** ou de **RCL**; il indique simplement qu'un affichage statique représente une différence et non un temps intermédiaire. L'affichage d'un chronométrage en cours n'est pas affecté.

Différences négatives et erreurs. Si le second de deux temps intermédiaires successifs est *inférieur ou égal* au premier :

- En mode stockage (\rightarrow **Dnn**), l'affichage montre le contenu du registre courant.
- En mode rappel (\rightarrow **Dnn**), l'affichage montre le message d'erreur **ERROR = Dnn**.

Remarque : Pour effacer le message d'erreur **ERROR = Dnn** ou **ERROR = Run**, déplacez le pointeur de registre. La pression sur **CLEAR** termine le mode rappel sans sortir de la situation d'erreur.

Le message **ERROR = Dnn** apparaît aussi lorsqu'un des deux registres impliqués en modes rappel et Δ Split contient une donnée sous une autre forme **HH.MMSSss** (cf. page 271).

Fonctions programmables du chronomètre

Le HP-41 dispose de six fonctions programmables du chronomètre : deux valident le clavier Chronomètre (**SW** et **SWPT**) et quatre opèrent sur le chronomètre quand son clavier n'est pas validé (**RUNSW**, **STOPSW**, **SETSW** et **RCLSW**) ; elles servent à l'intérieur d'un programme et ne sont pas exécutables au clavier du chronomètre.

Tous les temps transitant par le registre X doivent être sous la forme \pm **HH.MMSSss**.

Déclenchement et arrêt du chronomètre (**RUNSW**, **STOPSW**)

RUNSW déclenche le chronomètre.

STOPSW arrête le chronomètre.

Chargement et rappel de l'heure courante par **SETSW** et **RCLSW**

SETSW charge dans le chronomètre l'heure contenue dans le registre X sous la forme \pm **HH.MMSSss**.

Tout nombre relatif dont la valeur absolue est supérieure à 99.595999 est illicite et provoque un message d'erreur (**DATA ERROR**). Tout chiffre situé au-delà de la position ss est ignoré.

SETSW n'arrête pas un chronométrage en cours, mais le remet à la valeur spécifiée.

RCLSW rappelle dans le registre X l'indication courante du chronomètre sous la forme **HH.MMSSss** et décale la pile vers le haut (à moins que le décalage ne soit inhibé).

Spécification des pointeurs (**SWPT**)

La fonction **SWPT** (*stopwatch and pointer*) agit comme **SW**, avec en outre le positionnement du pointeur, ce qui permet à un programme de déplacer le pointeur de manière qu'il évite l'écrasement de données par des temps intermédiaires.

Procédure :

- A la validation du clavier du chronomètre, la valeur *sss.rrr* placée dans le registre X fixe les valeurs des pointeurs des registres de stockage (\rightarrow **Flm**, \rightarrow **Dnn**) et de rappel (\rightarrow **Rnn**, \rightarrow **Orr**).
- A l'invalidation du clavier du chronomètre, les valeurs du pointeur en cours sont renvoyées au registre X sous la forme *sss.rrr*. La valeur *sss.rrr* introduite est sauvegardée dans le registre LAST X.

La valeur de *sss* peut servir à déterminer combien de temps intermédiaires ont été pris, *en supposant qu'ils ont été stockés dans des registres consécutifs* et que le pointeur de registre de stockage de temps intermédiaires n'a pas été déplacé manuellement.

Valeur à placer dans le registre X



Une valeur *positive ou nulle* de *sss.rrr* valide le mode prise de temps intermédiaires (**#**). Une valeur *négative* de *sss.rrr* valide le mode différences de temps intermédiaires (**D**). Pour valider le mode prise de temps intermédiaire et mettre les deux pointeurs à zéro, spécifiez un nombre compris entre — 0.001 et zéro. Si le clavier est invalidé, que les deux pointeurs sont à zéro, et que le mode différence de temps intermédiaires est actif, le registre X reçoit la valeur — 0.0000001.

Dans le cas peu probable où, lors de l'invalidation du clavier, un pointeur est indéterminé (par exemple, son adresse est incomplète), la valeur de pointeur (*sss* ou *rrr*) placée dans le registre X est zéro. Ceci peut se produire lorsque le clavier du chronomètre est invalidé alors que l'adresse du pointeur de registre est incomplète.

Chronomètre en décompteur

Si le chronomètre est défini par **SETSW** à *un nombre négatif*, il déclenche une alarme à son passage à 00:00:00.00.

- Si le clavier et l'affichage du chronomètre ne sont pas actifs, l'alarme est semblable à une alarme-message, et affiche **TIMER ALARM**. Vous pouvez l'arrêter en appuyant sur une touche quelconque. Son annulation n'est pas nécessaire puisqu'elle n'est pas stockée en mémoire et ne peut être périmée.
- Si le clavier du chronomètre est actif, l'alarme ne fait qu'émettre une double tonalité; l'affichage du chronomètre n'est pas interrompu.

Le chronomètre ne s'arrête automatiquement, dans aucun des deux cas ; après avoir décompté, il commence à compter dès son passage par zéro. **[RS]** (au clavier) et **[STOPSW]** (pas au clavier) permettent l'arrêt du chronomètre.

Impression de temps intermédiaires

Si vous disposez d'une imprimante HP 82143A ou d'une imprimante HP-IL HP 82162A et désirez imprimer des valeurs stockées de temps intermédiaires, cela est possible grâce aux fonctions **[ATIME24]** (cf. chapitre 15) et **[PRA]** (cf. le manuel de votre imprimante). L'impression n'est possible que lorsque le clavier du chronomètre est actif.

Les différences de temps intermédiaires étant calculées et non stockées, vous pouvez les calculer de la manière suivante grâce à la fonction **[HMS--]** (qui n'existe pas sur le clavier du chronomètre) :

1. Rappelez le dernier temps intermédiaire de son registre de stockage.
2. Rappelez le précédent temps intermédiaire de son registre de stockage.
3. Exécutez la fonction **[HMS--]**.
4. Effacez le registre ALPHA (alphabétique), au besoin.
5. Exécutez **[ATIME24]**.
6. Exécutez **[PRA]**.

Pour imprimer une suite de valeurs, sautez les étapes 1 à 3 et rappelez le temps intermédiaire voulu.

Exemple : Programme de chronomètre (SPLITS)

Ce programme prépare le chronomètre à prendre des temps intermédiaires, valide le clavier du chronomètre et, lorsque le clavier est désactivé, sort sur l'imprimante connectée un groupe déterminé de temps intermédiaires mémorisés, sous la forme **HH:MM:SS.ss**. La valeur rappelée de chaque registre doit être inférieure à 100 ; sinon, un message d'erreur apparaît.

Commencez par stocker en mémoire de programmation la suite d'instructions listée page suivante. L'annexe J contient une version en code-barres de ce programme ; vous pouvez donc le saisir très rapidement si vous disposez du crayon-lecteur HP 82153A.

Entrée (quand le HP-41 la sollicite) :

1. Le numéro du premier registre contenant un temps intermédiaire à imprimer.
2. Le numéro du dernier registre contenant un temps intermédiaire à imprimer.

Résultat : Une liste ordonnée des registres et de leur contenu, sous le format spécifié.

Procédure :

1. Exécutez `[SPLITS]`.
2. Quand l'affichage le sollicite, prenez autant de temps intermédiaires que vous voulez. Vous pouvez les stocker à partir de n'importe quel registre, mais dans des registres successifs.
3. Appuyez sur `[R]` quand la prise de temps intermédiaires est terminée.
4. En réponse au message `FIRST REG?`, tapez le numéro du premier registre dont vous voulez imprimer le contenu. Si vous voulez partir de R_{00} , ne tapez rien. Appuyez sur `[R/S]`.
5. En réponse au message `LAST REG?`, tapez le numéro du dernier registre dont vous voulez imprimer le contenu. Appuyez sur `[R/S]`.

Listage du programme

01 ♦ LBL "SPLITS"	
02 STOPSW	Arrêt du chronomètre.
03 0	
04 SETSW	Remise du chronomètre à 00:00:00.00.
05 SW	Validation du clavier. Une fois pris tous les temps intermédiaires et le clavier invalidé, le programme continue.
06 "FIRST REG?"	À défaut de nombre, le registre est R_{00} (ligne 03).
07 PROMPT	
08 "LAST REG?"	
09 PROMPT	
10 ADV	Avance du papier de l'imprimante.
11 RCLFLAG	
12 X<>Z	Sauvegarde de l'état des indicateurs binaires dans le registre Z.
13 ♦ LBL 00	
14 FIX 0	Élimination de l'éventuelle partie décimale du numéro de registre.
15 CF 29	En conjonction avec <code>[FIX] 0</code> , suppression du séparateur décimal.
16 "R"	Début de la formation de la chaîne de sortie (R pour registre).
17 100	Lignes 17 à 23 : formattage de la sortie. Si le numéro de registre est inférieur à 100, insertion d'un espace entre R et le numéro. Si le numéro de registre est inférieur à 10, ajout d'un zéro en chiffre de dizaine. Ceci permet d'aligner à l'impression les numéros de registres et leur contenu.
18 X<=Y?	
19 GTO 01	
20 "†"	
21 SQRT	
22 X>Y?	
23 "† 0"	

24+LBL 01	Recopie dans X du numéro du premier registre.
25 RND	Adjonction dans le registre ALPHA du numéro de registre figurant dans X.
26 ARCL X	Rappel de la valeur contenue dans le registre dont l'adresse est dans X.
27 "┌="	
28 RCL IND X	
29 FIX 6	
30 ATIME24	Adjonction de cette valeur dans le registre ALPHA.
31 PRA	Impression de la chaîne contenue dans le registre ALPHA.
32 RND	Incrémentation du numéro de registre et poursuite si ce numéro est inférieur ou égal à celui du dernier registre.
33 1	
34 +	
35 X < = Y?	
36 GTO 00	
37 RCL Z	Restitution de l'état initial des indicateurs binaires, formats d'affichage compris.
38 STOFLAG	
39 CLX	Effacement du registre X.
40 END	

Exemple de sortie : Les temps intermédiaires ont été stockés dans les registres R₀₀ à R₁₀.

```

                                XEQ "SPLITS"
FIRST REG?                      RUN
                                LAST REG?
                                10.0000    RUN

R 00 =00:00:07.62
R 01 =00:00:09.35
R 02 =00:00:10.20
R 03 =00:00:12.24
R 04 =00:00:13.99
R 05 =00:00:07.06
R 06 =00:00:08.90
R 07 =00:00:10.54
R 08 =00:00:12.76
R 09 =00:00:13.77
R 10 =00:00:17.89

```


Bases de programmation

Table des matières

Chargement d'un programme	280
Saisie au clavier	280
Copie d'un programme résidant en ROM	281
Extension de la mémoire programme	281
Exécution d'un programme	282
Lignes de programme	282
Opérations non programmables	283
Déplacement du pointeur en mémoire programme	283
Utilisation de GTO ▢	283
Utilisation du catalogue 1	284
Incréméntation et décrémentation du pointeur (SST , BST)	284
Autres méthodes	285
Mise au point d'un programme	285
Suppression d'instructions	285
Insertion d'instructions	286
Effacement de programmes	286
Utilisation de CLP	287
Utilisation de PCLPS	287

Chargement d'un programme

Saisie au clavier

1. Appuyez sur **PRGM** pour passer en mode programme.
2. Appuyez sur **GTO** **▢** **▢** pour placer le pointeur de programme en début de mémoire.
3. Appuyez sur **LBL** , suivi d'un label global.
4. Tapez les instructions au moyen des claviers normal, personnel et alphabétique comme vous le faites en mode exécution.
5. Appuyez sur **GTO** **▢** **▢** pour terminer le programme (facultatif).

La pression sur **[GTO]** **[.]** **[.]** a les effets suivants :

- La mémoire principale est compactée, afin de rendre tous les registres possibles disponibles pour le prochain programme, la redéfinition du clavier ou des alarmes.
- Une instruction **[END]** est insérée pour terminer le programme précédent, créant un programme nul au début de la mémoire (consistant en l'instruction **.END.**). L'une des raisons d'appuyer sur **[LVEB]** **[.]** **[.]** après le chargement d'un programme est de donner au programme sa propre instruction **[END]**, afin que le catalogue 1 affiche le nombre de multiplètes qu'il occupe.
- Le pointeur de programme est placé à ce programme nul; l'afficheur montre **00 REG nnn**, où *nnn* indique le nombre de registres disponibles pour un nouveau programme. Les instructions que vous tapez sont stockées au début de la mémoire.

Le nombre de registres disponibles apparaît aussi avec l'instruction **.END.** permanente. Si la dernière ligne de programme est affichée, l'appui sur **[SST]** fait passer l'affichage à **.END. REG nnnn**. Pour ajouter des instructions, il vous suffit de les taper. Pour revoir votre programme :

- Appuyez sur **[SST]** pour examiner la première ligne.
- Appuyez sur **[BST]** pour examiner la dernière ligne tapée.

Copie d'un programme résidant en ROM

Si vous désirez modifier un programme résidant dans une mémoire morte (ROM) telle qu'un module d'application, vous devez d'abord le copier en mémoire programme en exécutant **[COPY]** et en spécifiant n'importe quel label global du programme concerné. La copie est stockée en début de mémoire programme.

Extension de la mémoire programme

Si la mémoire programme n'est pas assez vaste pour accueillir une instruction que vous ajoutez ou un programme que vous copiez, l'affichage indique **PACKING** (compactage en cours) puis **TRY AGAIN** (essayez de nouveau). Si vous essayez effectivement de nouveau et que le message **TRY AGAIN** réapparaît, effectuez une ou plusieurs des opérations suivantes afin d'étendre la mémoire programme disponible :

- Examinez au moyen de **[SIZE?]** combien de registres sont affectés au stockage de données et diminuez ce nombre par **[SIZE]** ou **[PSIZE]**.
- Effacez les programmes terminés par **[CLP]** ou **[PCLPS]**. Vous pouvez auparavant les sauvegarder en mémoire annexe par **[SAVEP]**.
- Annulez une ou plusieurs alarmes.
- Annulez les réaffectations de touches du clavier personnel autres que les labels globaux du catalogue 1; exécutez **[PACK]** ou **[DTG]** **[.]** **[.]**.

Exécution d'un programme

Pour exécuter un programme, assurez-vous que le HP-41 est en mode exécution, puis effectuez l'une des opérations suivantes :

- Appuyez sur **[XEQ]** et spécifiez le label global du programme où vous désirez que l'exécution commence.
- Affectez un label global à une touche. Le clavier personnel étant actif, appuyez sur cette touche. L'exécution commence à ce label.
- Placez le pointeur au début du programme et appuyez sur **[R/S]** . L'exécution commence à la ligne courante.
- Placez le pointeur au début du programme et appuyez sur **[SST]**. Seule la ligne courante est exécutée; le pointeur est incrémenté. L'exécution d'une seule ligne est utile lors de la mise au point de programme. L'examen du résultat de chaque ligne permet d'isoler une éventuelle erreur.
- Placez le pointeur au début du programme en armant l'indicateur binaire 11 et arrêtez le HP-41. A la prochaine mise en marche, il exécute automatiquement le programme en commençant à la ligne courante.
- Armez une alarme pour exécuter le programme à une heure déterminée.

Le témoin **PRGM** apparaît au bas de l'affichage quand un programme est en cours d'exécution. L'indicateur d'exécution (†) apparaît aussi, à moins qu'une fonction comme **[RVIEW]** affiche un message ou qu'une fonction comme **[ED]** valide un clavier et un affichage spéciaux. Chaque fois que le programme exécute un label, l'indicateur d'exécution (†) se déplace d'une position vers la droite.

Lignes de programme

En mode programme, le HP-41 affiche une ligne de programme à la fois. Les lignes sont créées automatiquement au fur et à mesure que vous tapez les instructions. A chaque ligne est affecté un nombre qui indique sa position dans le programme; chaque programme a son propre jeu de lignes numérotées. Chaque ligne comprend une instruction complète, composée de :

- Une fonction.
- Une chaîne alphanumérique de 15 caractères maximum.
- Un nombre de 10 chiffres maximum, plus éventuellement une puissance de 10 ayant 2 chiffres.

Le chapitre 9 « Le clavier et l'affichage » explique la saisie des nombres et des chaînes alphanumériques.

Lors de l'affichage d'une ligne de programme, le symbole † indique que la chaîne qui suit comporte une chaîne alphanumérique ou, s'il est précédé de **XEQ**, **GTO** ou **LBL**, un label global. Vous devez appuyer d'abord sur **[XEQ]** pour introduire une fonction dans une ligne de programme par son nom (alphabétique). Sinon, le HP-41 ne reconnaît pas un nom de fonction dans les caractères alphabétiques, qu'il traite alors comme tels et stocke dans le registre ALPHA lorsqu'il exécute cette ligne.

Opérations non programmables

Les opérations suivantes ne sont pas programmables ; certaines peuvent être effectuées par d'autres moyens, les alternatives programmables sont indiquées à la suite entre parenthèses.

- Opérations d'annulation :

↵, **DEL**

CLP (mais **PCLRS** est programmable).

- Opérations de déplacement :

GTO **□** **□**, **GTD** **□**, **SST**, **GST**

- Tous les catalogues (mais **EMDIR** et **ALMCAT** sont programmables)

- Touches fonctionnant en bascule :

ON (mais la touche **OFF** est programmable)

PRGM

USER (indicateur binaire 27)

ALPHA (mais **AON** et **AOFF** sont programmables)

- Opérations diverses :

COPY, **ON**, **PACK**, **R/S** (pour exécuter un programme)

ASN (mais **PAGN** est programmable)

SIZE (mais **PSIZE** est programmable)

Déplacement du pointeur en mémoire programme

Le HP-41 dispose de plusieurs méthodes de déplacement du pointeur en mémoire programme : accès à tous programme en mémoire (c'est-à-dire à tout label global) ou accès à toute ligne d'un programme. Certaines n'opèrent qu'en mode exécution, d'autres en mode programme. Une seule, **GTO** **□**, opère les deux fonctions dans les deux modes.

Utilisation de **GTO** **□**

En mode programme ou exécution :

- Pour placer le pointeur sur tout label global déterminé, appuyez sur **GTO** **□** et spécifiez le label global. La recherche commence par le dernier label global figurant au catalogue 1, et progresse vers le premier ; elle s'arrête au premier qui correspond.
- Pour placer le pointeur à la ligne *nnn* du programme courant, appuyez sur **GTO** **□** *nnn*. Si *nnn* excède le numéro de la ligne, le pointeur reste à la dernière ligne. Si le numéro de la ligne excède 999, appuyez sur **GTO** **□** **EEX** : l'affichage est alors **GTO . 1---** ; tapez les 3 derniers chiffres.

Utilisation du catalogue 1

[CATALOG] ne peut placer le pointeur sur le programme désiré dans quelques cas :

- Le programme ne contient pas de labels globaux.
- Le label concerné figure dans plusieurs programmes et, le HP-41 commençant la recherche par la fin du catalogue 1, s'arrête au premier rencontré, c'est-à-dire au dernier de la liste.
- Vous avez oublié la dénomination exacte du label.

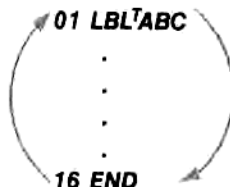
Vous pouvez placer le pointeur sur tout label global ou instruction **[END]** dans la mémoire programme en utilisant le catalogue 1, en mode programme ou exécution, de la manière suivante :

1. Appuyez sur **[CATALOG]** 1 pour afficher tous les labels globaux et instructions **[END]** présents en mémoire programme.
2. Pour accélérer le listage, appuyez sur toute touche autre que **[ON]** et **[R/S]**.
3. Appuyez sur **[R/S]** pour arrêter le listage au label global ou à l'instruction **[END]** désiré.
4. Pour passer au nom suivant ou précédent de la liste, appuyez sur **[SST]** ou sur **[DST]**, respectivement.
5. Pour placer le pointeur sur le nom affiché, appuyez sur **[+]**.

Incrémentement et décrémentation du pointeur (**[SST]**, **[DST]**)

En mode programme, vous pouvez incrémenter ou décrémentation le pointeur, c'est-à-dire le déplacer à la ligne suivante ou à la ligne précédente en appuyant sur **[SST]** ou sur **[DST]**. A noter qu'il y a bouclage entre le début et la fin du programme, c'est-à-dire que **[SST]** à la dernière ligne fait passer à la première et **[DST]** à la première ligne fait passer à la dernière.

La pression sur **[SST]** quand l'ordinateur est positionné à la fin du programme ramène l'ordinateur au début de ce programme.



La pression sur **[DST]** quand l'ordinateur est positionné au début du programme ramène l'ordinateur à la fin de ce programme.

Autres méthodes

Le HP-41 étant en mode exécution, vous pouvez déplacer le pointeur dans toute la mémoire programme de plusieurs façons :

- Appuyez sur **[GTO]** et spécifiez le label global pour y placer le pointeur.
- Pour placer le pointeur à un label global affecté à une touche, appuyez sur cette touche en appuyant sur **[R/S]**, puis relâchez cette touche.
- Pour placer le pointeur au label numérique *nn* dans le programme courant, appuyez sur **[GTO]** *nn*. Le HP-41 recherche **[LBL]** *nn* (cf. « Branchements » au chapitre 20) et s'arrête au premier label qui convient.
- Pour placer le pointeur au début du programme courant, appuyez sur **[RMI]**. L'affichage **00 REG nnn** indique que *nnn* registres sont disponibles. Si vous tapez une instruction, elle occupe la ligne 01. Ceci constitue la plus simple méthode pour ajouter une instruction au début d'un programme.

Mise au point d'un programme

Toute la mise au point de programme, suppression et insertion d'instructions, a lieu en mode programme.

Suppression d'instructions

Suppression d'une ligne. Pour annuler une seule instruction, placez le pointeur sur la ligne concernée, et appuyez sur **[←]**. Cette ligne de programme est supprimée, le pointeur est placé à la ligne précédente et le numéro de chaque ligne qui suit est décrémenté de 1.

Si vous supprimez plusieurs lignes, commencez par celle dont le numéro est le plus grand. Dans l'exemple suivant, il s'agit d'annuler les lignes 02 à 04. Commencez par placer le pointeur à la ligne 4; appuyez sur **[←]**; l'instruction **[END]** est renumérotée 04. Appuyez sur **[←]** à nouveau afin de supprimer la ligne 03. Appuyez sur **[←]** encore une fois pour supprimer la ligne 02. Le pointeur se trouve alors à la ligne 01.

	01 LBL ^T SURFACE		
	02 X ↑ 2	01 LBL ^T SURFACE	
	03 PI	02 X ↑ 2	01 LBL ^T SURFACE
Ligne courante (affichée)	04 *	03 PI	02 X ↑ 2
	05 END	04 END	01 LBL ^T SURFACE
			03 END
			02 END

Suppression de plusieurs lignes. Pour supprimer une longue suite d'instructions :

1. Placez le pointeur sur la première ligne à supprimer (celle ayant le plus petit numéro).
2. Exécutez **[DEL]** (*delete* = annulation).
3. Spécifiez le nombre de lignes à supprimer. S'il y a plus de 1 000 lignes, appuyez sur **[EEK]** ; l'affichage étant **DEL 1_ _ _**, tapez les 3 derniers chiffres.

Dans l'exemple précédent, les lignes 02 à 04 ont été supprimées une par une. Pour les supprimer d'un seul coup, placez le pointeur à la ligne 02 et exécutez **[DEL] 003**. Le pointeur est alors en ligne 01, la ligne précédant la première ligne étant supprimée. Les lignes suivantes sont renumérotées.

Si vous exécutez **[DEL] nnn**, où *nnn* est supérieur au nombre de lignes suivant la ligne courante, toutes ces lignes sont supprimées à l'exception de **[END]**.

Insertion d'instructions

Pour ajouter une instruction à un programme, placez le pointeur à la ligne précédant immédiatement la future ligne, puis tapez la nouvelle instruction. Si vous venez de supprimer une instruction au moyen de **[*]** et désirez alors la remplacer, le pointeur est déjà à la position voulue. La nouvelle instruction devient la ligne courante et toutes les suivantes sont renumérotées.

Si vous devez insérer plusieurs lignes, commencez par la première. Supposez que vous deviez réintroduire les lignes annulées à l'exemple précédent. Sur listage de gauche, le pointeur est en ligne 01. Chaque nouvelle ligne frappée est insérée après la ligne courante et devient la nouvelle ligne courante.

				01 LBL ^T SURFACE
				02 X ↑ 2
				03 PI
				04 *
		01 LBL ^T SURFACE	02 X ↑ 2	03 PI
				04 END
				05 END
Ligne courante (affichée)	[DEL]	[DEL]	[DEL]	[DEL]
	01 LBL ^T SURFACE	02 X ↑ 2	03 PI	04 *
	02 END	03 END	04 END	05 END

Effacement de programmes

La fonction **[CLP]** (*clear program*) efface (supprime) un programme et n'est pas programmable. La fonction **[PCLPS]** (*programmable clear programs*) efface plusieurs programmes et est programmable.

Utilisation de **CLP**

Exécutez **CLP** et spécifiez n'importe quel label du programme à annuler. Le HP-41 :

1. Recherche le label global spécifié, en commençant par la fin du catalogue 1.
2. Annule toutes les instructions (de la ligne 01 à **END**) du premier programme rencontré qui contient le label spécifié.
3. Compacte la mémoire principale.

Pour annuler le programme courant, exécutez **CLP** et appuyez sur **ALPHA** deux fois, sans spécifier de label.

Utilisation de **PCLPS**

Pour annuler un programme et tous les programmes suivants :

1. Spécifiez dans le registre ALPHA un label global du premier programme.
2. Exécutez **PCLPS**.

Pour annuler le programme courant et les programmes suivants, ne spécifiez aucun label dans le registre ALPHA, et exécutez **PCLPS**.

Indicateurs binaires

Table des matières

Introduction	288
Types d'indicateurs binaires	289
Indicateurs binaires de l'utilisateur (00 à 10)	289
Indicateurs binaires de commande (11 à 29)	289
Indicateurs binaires du système (30 à 55)	291
Résumé de l'état des indicateurs binaires	292
Indicateurs binaires et registre X	292
Indicateurs binaires et nombres (octets)	294
Utilisation de <code>X<>F</code>	295
Copies multiples d'indicateurs binaires	295
Utilisation de <code>RCLFLAG</code> et <code>STOFLAG</code>	296

Introduction

Un indicateur binaire possède deux états : *armé* et *désarmé*. Ces états peuvent être interprétés par les deux nombres binaires 0 et 1, les deux états « arrêt » et « marche » d'un interrupteur, les deux états « éteint » et « allumé » d'un témoin ou les deux réponses « non » et « oui » à une décision logique. Le HP-41 possède 56 indicateurs binaires, répartis en trois groupes selon leur utilisation.

Indicateurs binaires de l'utilisateur. Vous pouvez les tester et les modifier; seules vos instructions peuvent les modifier.

Indicateurs binaires de commande. Vous pouvez les tester et les modifier. Le HP-41 en réinitialise certains à chaque mise sous tension et en modifie certains en cours de fonctionnement.

Indicateurs binaires du système. Vous pouvez les tester, mais pas les modifier.

Vous pouvez modifier les indicateurs binaires 00 à 29 (utilisateur et commande) comme suit :

- Pour armer un indicateur binaire, appuyez sur `SF` (*set flag*) et spécifiez son numéro.
- Pour désarmer un indicateur binaire, appuyez sur `CF` (*clear flag*) et spécifiez son numéro.

Pour tester l'un des indicateurs binaires 00 à 55, appuyez sur `FS?` (*flag set ? = indicateur binaire armé ?*) et spécifiez son numéro. L'affichage indique **YES** s'il est armé et **NO** s'il est désarmé. Le test d'indicateurs binaires tel que `FS?` sert principalement à contrôler le déroulement d'un programme (cf. « Branchements » au chapitre 20).

Types d'indicateurs binaires

Indicateurs binaires de l'utilisateur (00 à 10)

Les indicateurs binaires 00 à 10 sont réservés à l'utilisateur, qui leur attribue la signification qu'il veut. Par exemple, un programme peut demander si les angles sont exprimés en degrés ou en radians, et mémoriser la réponse dans l'un de ces indicateurs binaires pour l'utiliser par la suite.

L'état de chaque indicateur binaire de l'utilisateur est gardé en mémoire permanente. Il ne change pas tant que vous ne le modifiez pas volontairement. L'affichage dispose de témoins numériques (**0**, **1**, **2**, **3** ou **4**) associés aux indicateurs binaires 00 à 04.

Les huit premiers indicateurs binaires (00 à 07) peuvent être interprétés comme les 8 bits d'un octet, qui peut être transformé en un nombre décimal dans le registre X (cf. dans ce chapitre après les indicateurs binaires de commande).

Indicateurs binaires de commande (11 à 29)

Les indicateurs binaires 11 à 29 ont chacun une signification particulière pour le HP-41 (décrite ci-dessous). L'état de ces indicateurs binaires représente certaines conditions et options de fonctionnement. Vous pouvez modifier les indicateurs binaires d'options. Le HP-41 modifie les indicateurs binaires de condition, que vous pouvez tester.

Indicateur binaire 11 : exécution automatique. Si vous armez l'indicateur binaire 11 avant d'arrêter le HP-41, la prochaine mise sous tension provoque :

1. L'émission d'une tonalité.
2. L'exécution du programme à partir de la ligne courante.
3. Le désarmement de l'indicateur binaire 11.

Indicateurs binaires 12 à 20 : commande de périphériques externes par le HP-41. Tous sont désarmés à chaque mise sous tension du HP-41. La signification particulière de chacun dépend du dispositif ; consultez le manuel approprié.

Indicateur binaire 21 : autorisation d'imprimer. Il permet au programme de contrôler l'exécution de fonctions telles que `VIEW` et `AVIEW`, selon la présence ou l'absence d'un périphérique de sortie (cf. « Opérations d'impression » à l'annexe D).

Indicateurs binaires 22 et 23 : entrée de données. Ils permettent à un programme qui sollicite une entrée de données de déterminer le type de réponse :

- L'indicateur binaire 22 est armé par la frappe d'un nombre dans le registre X.
- L'indicateur binaire 23 est armé par la frappe de caractères dans le registre ALPHA.

Ces indicateurs binaires sont désarmés automatiquement à la mise en marche du HP-41. Si vous voulez les tester, désarmez-les avant de taper la réponse.

Indicateurs binaires 24 et 25 : pas d'arrêt sur erreur. Toute erreur arrête normalement l'exécution d'un programme. Ces indicateurs binaires permettent de ne pas arrêter l'exécution d'un programme en cas d'erreur et de les utiliser comme outil de programmation.

- Si l'indicateur binaire 24 est armé, le HP-41 ne prend pas en compte *toutes* les erreurs du type hors limites (**OUT OF RANGE**), résultant de tout calcul (sauf les cumuls statistiques) donnant un nombre x tel que $|x| > 9.99999999 \times 10^{99}$. Le nombre $\pm 9.99999999 \times 10^{99}$ est alors pris pour résultat du calcul et l'exécution continue.

L'indicateur binaire 24 est désarmé à chaque mise en marche du HP-41. Si vous l'avez armé, il le reste jusqu'à ce que vous le désarmiez explicitement ou que vous arrêtiez le HP-41. Si vous désirez un branchement sur votre propre sous-programme plutôt que d'utiliser la valeur-butée $\pm 9.99999999 \times 10^{99}$, utilisez l'indicateur binaire 25.

- Si l'indicateur binaire 25 est armé, le HP-41 ne prend pas en compte une erreur et *une seule* (quelconque) et désarme l'indicateur binaire 25. La commande qui a provoqué l'erreur n'est pas exécutée. L'indicateur binaire 25 est désarmé à chaque mise en marche du HP-41.

Si les deux indicateurs binaires 24 et 25 sont armés, un résultat hors limite (**OUT OF RANGE**) est traité par l'indicateur binaire 24, et l'indicateur binaire 25 n'est pas désarmé. A noter que si l'indicateur binaire 25 est armé et l'indicateur binaire 24 désarmé, un résultat hors limite (**OUT OF RANGE**) ne fait pas placer la valeur-butée dans le registre approprié.

Vous pouvez détecter une erreur en armant l'indicateur binaire 25 juste avant une commande, et tester l'indicateur binaire juste après la commande (et le désarmer, car il est dangereux de ne pas prendre en compte des erreurs inattendues). Ceci permet en cas d'erreur un branchement au lieu d'un arrêt d'exécution.

Indicateur binaire 26 : autorisation de tonalité. BEEP, TONE, les alarmes et le chronomètre émettent une tonalité si l'indicateur binaire 26 est armé (il est le seul à être armé par défaut). Aucune tonalité n'est produite si l'indicateur binaire 26 est désarmé.

Indicateur binaire 27 : clavier personnel. Armé quand le clavier personnel est validé, c'est-à-dire quand le témoin **USER** est présent. Un programme peut le tester ou le modifier de la même manière que vous pouvez vérifier le témoin ou appuyez sur USER. L'indicateur binaire 27 est gardé en mémoire permanente.

Indicateurs binaires 28 et 29 : ponctuation à l'affichage. Ils commandent l'utilisation des points et des virgules dans les nombres et sont gardés en mémoire permanente (cf. « Format d'affichage numérique » au chapitre 9).

Exemples de programmes. Les programmes TR et SIGMA du chapitre 22 utilisent les indicateurs binaires 22 et 23 pour l'entrée de données lorsqu'ils proposent de confirmer ou de modifier des valeurs données par défaut. L'utilisateur peut soit confirmer les valeurs en appuyant sur **[RS]**, soit taper d'autres valeurs et appuyer sur **[RS]**. Le programme désarme l'indicateur binaire voulu, puis le teste ; il est armé si l'utilisateur a tapé une autre valeur.

Les programmes utilisent l'indicateur binaire 25 en accédant à des enregistrements du fichier de textes TRECS. Afin d'accéder à tous les enregistrements sans en connaître d'avance le nombre, ils contiennent des boucles qui traitent un enregistrement, arment l'indicateur binaire 25 avant d'accéder à l'enregistrement suivant, puis le testent. S'il n'y a plus d'enregistrements, il y a erreur et l'indicateur binaire 25 est désarmé, ce qui permet au programme de sortir de la boucle.

Indicateurs binaires du système (30 à 55)

Les indicateurs binaires 30 à 55 servent principalement au HP-41. Vous pouvez les tester, quoique certains apparaissent toujours désarmés. Vous ne pouvez les modifier, mais sauvegarder et restituer ceux qui représentent des options accessibles à l'utilisateur. Voici quelques façons d'utiliser certains de ces indicateurs binaires.

Indicateurs binaires représentant une option. Vous pouvez sauvegarder et restituer certaines options, codées sous forme d'indicateurs binaires au moyen de **[STOFLAG]** et **[RCLFLAG]**, décrites à la fin de ce chapitre. Un programme peut ainsi utiliser des options, puis restituer les conditions initiales.

Certains dispositifs externes utilisent des indicateurs binaires du système pour représenter leurs propres options :

- L'indicateur binaire 31 représente le format de la date, décrit page 242.
- Les indicateurs binaires 36 à 39 représentent le nombre de chiffres affichés, décrit pages 160 et 161.
- Les indicateurs binaires 40 et 41 représentent le format de l'affichage, décrit pages 160 et 161.
- Les indicateurs binaires 42 et 43 représentent l'unité d'angle, décrite page 180.

Indicateurs binaires représentant une condition. Les indicateurs binaires suivants fournissent des informations utiles à certains programmes :

- L'indicateur binaire 44 est armé lorsque **[ON]** est exécutée.
- L'indicateur binaire 48 est armé lorsque le clavier alphabétique est validé, c'est-à-dire quand le témoin **ALPHA** apparaît.

- L'indicateur binaire 49 est armé (et le témoin **BAT** apparaît) lorsque la tension de l'alimentation électrique faiblit. Un programme dont l'exécution est longue peut tester l'indicateur binaire 49 de temps en temps, et exécuter une instruction `OFF` s'il est armé. Sinon, le contenu de la mémoire peut être affecté.
- L'indicateur binaire 50 est armé lors de l'affichage d'un message.
- L'indicateur binaire 55 est armé si une imprimante est connectée. Il est en relation avec l'indicateur binaire 21 (autorisation d'imprimer). Leur interaction est décrite à l'annexe D, « Opérations d'impression ».

Résumé de l'état des indicateurs binaires

Le tableau de la page suivante indique l'état des indicateurs binaires lorsque la mémoire permanente a été effacée, et à la mise en marche. Il mentionne deux autres états :

M = maintenu par la mémoire permanente

? = dépend d'autres conditions.

Indicateurs binaires et registre X

Il y a trois raisons d'échanger des données entre les indicateurs binaires et le registre X :

1. Pour sauvegarder et restituer des options (telles que le format de l'affichage) codées par le HP-41 dans un indicateur binaire de commande ou du système.
2. Pour garder plusieurs copies d'un groupe d'indicateurs binaires de l'utilisateur, une seule copie étant active à un instant déterminé.
3. Pour convertir en nombre décimal le nombre binaire représentant l'ensemble des indicateurs binaires de l'utilisateur, et vice versa.

Il y a deux façons d'échanger des données entre les indicateurs binaires et le registre X :

- `X<>F` (*X exchange flags*), fonctionnant en bascule, échange l'état des indicateurs binaires 00 à 07 et le registre X (nombre décimal de 0 à 255). Ce nombre peut servir à des calculs ou des branchements. Cependant `X<>F` ne peut pas affecter les indicateurs de commande ou du système.
- `STOFLAG` (*restore flags*) et `RCLFLAG` (*recall flags*) sauvegardent et restituent l'état des indicateurs binaires 00 à 43, mais ne peuvent le transformer en un nombre décimal.

En résumé, seuls `STOFLAG` et `RCLFLAG` opèrent sur les indicateurs binaires de commande et du système, et seul `X<>F` convertit les états en nombre décimal. Tous peuvent servir à plusieurs copies des indicateurs binaires de l'utilisateur.

Résumé des états des indicateurs binaires

Numéro	Nom	Etat à la réinitialisation	Etat à la mise en marche
00-10	Utilisateur	0	M
11	Exécution automatique	0	0
12-20	Commande de dispositif externe	0	0
21	Autorisation d'imprimer	?	?
22	Entrée de données numériques	0	0
23	Entrée de données alphabétiques	0	0
24	Poursuite malgré erreur de limite	0	0
25	Poursuite malgré erreur	0	0
26	Autorisation de tonalité	1	1
27	Clavier utilisateur	0	M
28	Ponctuation à l'affichage	1	M
29	Séparateur de chiffres	1	M
31	Format de la date	0	M
36	Nombre de chiffres	0	M
37	Nombre de chiffres	1	M
38	Nombre de chiffres	0	M
39	Nombre de chiffres	0	M
40	Format de l'affichage	1	M
41	Format de l'affichage	0	M
42	Unité d'angle	0	M
43	Unité d'angle	0	M
44	Fonction ON	0	0
48	Clavier alphabétique	0	0
49	Tension d'alimentation	?	?
50	Affichage d'un message	0	0
55	Imprimante connectée	?	?

Indicateurs binaires et nombres (octets)

L'*octet* est la clé de la correspondance entre indicateurs binaires et nombres.

Un octet est constitué de 8 *bits*, ou chiffres binaires : 00010100, 10001100, 00100000 et 00001111 sont des exemples d'octets.

- Un octet peut être interprété comme un ensemble de huit indicateurs binaires, chaque 0 ou 1 correspondant à l'état d'un indicateur binaire (désarmé ou armé).
- Un octet peut être également interprété comme un nombre, la somme des puissances de 2.

L'ensemble des 8 indicateurs binaires 00 à 07 peut être considéré comme un nombre binaire à 8 chiffres, ou *octet*, et converti en nombre décimal (0 à 255). Inversement, vous pouvez coder décimalement l'ensemble des 8 indicateurs binaires.

Valeurs décimales et indicateurs binaires 00 à 07

Numéro indicateur	07	06	05	04	03	02	01	00
Etat	Armé ou Désarmé	Armé ou Désarmé	Armé ou Désarmé	Armé ou Désarmé	Armé ou Désarmé	Armé ou Désarmé	Armé ou Désarmé	Armé ou Désarmé
Valeur binaire	2^7 ou Zéro	2^6 ou Zéro	2^5 ou Zéro	2^4 ou Zéro	2^3 ou Zéro	2^2 ou Zéro	2^1 ou Zéro	2^0 ou Zéro
Valeur décimale	128 ou Zéro	64 ou Zéro	32 ou Zéro	16 ou Zéro	8 ou Zéro	4 ou Zéro	2 ou Zéro	1 ou Zéro

Exemple. Soient 4 octets particuliers, représentant 4 combinaisons de 7 indicateurs binaires.

Indicateurs armés	Octet	Valeur décimale
04,02	00010100	$16 + 4 = 20$
07, 03, 02	10001100	$128 + 8 + 4 = 140$
05	00100000	32
03, 02, 01, 00	00001111	$8 + 4 + 2 + 1 = 15$

Utilisation de `X<>F`

Lorsque vous exécutez `X<>F` :

- L'octet représentant les indicateurs binaires 00 à 07 est converti en un nombre décimal compris entre 0 et 255, qui est placé dans le registre X.
- Le nombre contenu dans le registre X (dont la valeur absolue doit être comprise entre 0 et 255, et dont le signe et la partie décimale dans le registre X ne sont pas pris en compte) est converti en binaire et interprété comme l'ensemble des états des indicateurs binaires 00 à 07.

Copies multiples d'indicateurs binaires

Vous pouvez augmenter le nombre des indicateurs binaires de l'utilisateur en sauvegardant puis en restituant l'état des indicateurs binaires réels. Lorsque vous sauvegardez l'état courant d'un indicateur binaire réel, vous en créez une copie « gelée ». Vous pouvez alors utiliser librement l'indicateur binaire, y compris sauvegarder de nouveau son état à un moment déterminé, et ensuite restituer à l'indicateur binaire réel l'état que vous avez sauvegardé précédemment.

L'exemple suivant illustre ce procédé. Les indicateurs binaires 00 et 02 sont armés et l'octet représentant l'état de tous les indicateurs binaires est sauvegardé dans le registre R₀₀. Ensuite, les indicateurs binaires 01 et 03 sont armés et l'octet représentant l'état de tous les indicateurs binaires est sauvegardé dans le registre R₀₁.

01 0	Désarmement des indicateurs binaires 00 à 07.
02 X<>F	
03 SF 00	Armement de l'indicateur binaire 00.
04 SF 02	Armement de l'indicateur binaire 02.
05 0	Désarmement des indicateurs binaires 00 à 07.
06 X<>F	
07 STO 00	Sauvegarde du premier état dans R ₀₀ .
08 SF 01	Armement de l'indicateur binaire 01.
09 SF 03	Armement de l'indicateur binaire 03.
10 RCL 00	Rappel du premier état dans X.
11 X<>F	Chargement du second état dans X et restitution du premier état.
12 STO 01	Sauvegarde du second état dans R ₀₁ .

Ceci crée deux jeux d'états et il est possible d'en créer d'autres. Vous ne pouvez cependant accéder qu'à un seul jeu d'états à la fois : vous devez alors sauvegarder l'état courant, rappeler l'état désiré, puis tester ou manipuler les indicateurs binaires réels.

Exemple. Supposons qu'un programme fasse appel à plusieurs sous-programmes, chacun utilisant plusieurs indicateurs binaires. Avant d'appeler un sous-programme, le programme peut copier dans un registre particulier le jeu courant d'états des indicateurs binaires pour ce sous-programme. Ensuite, chaque fois qu'un sous-programme est appelé, il peut commencer par restituer son propre jeu d'états, tester et manipuler les indicateurs binaires, et sauvegarder son jeu réactualisé. De cette façon, le programme peut appeler les sous-programmes dans n'importe quel ordre, ceci un nombre quelconque de fois : les sous-programmes n'interfèrent pas entre eux.

Supposons que vous analysiez les revenus d'une population par sexe, âge et formation. Vous pouvez affecter les indicateurs binaires comme suit :

Sexe : indicateur binaire 00 armé = féminin ; désarmé = masculin
Age : indicateur binaire 01 armé = âge < 20
 indicateur binaire 02 armé = $20 \leq \text{âge} < 35$
 indicateur binaire 03 armé = $35 \leq \text{âge} < 50$
 indicateur binaire 04 armé = $50 \leq \text{âge}$
Formation : indicateur binaire 05 armé = secondaire
 indicateur binaire 06 armé = BAC
 indicateur binaire 07 armé = supérieure

Chaque personne peut être représentée par deux nombres : son salaire et un nombre de 0 à 255, équivalent décimal du jeu d'états des indicateurs binaires 00 à 07.

Un programme peut produire des résultats par catégories. Par exemple, pour la catégorie hommes, indépendamment de l'âge et de la formation, le programme doit restituer aux indicateurs binaires réels les valeurs correspondant à chaque personne, tester l'indicateur binaire 00, et prendre en compte le revenu des personnes dont l'indicateur binaire 00 est désarmé.

Tous les indicateurs binaires sont impliqués pour définir la catégorie des femmes de 35 à 50 ans ayant une formation supérieure. Il est alors plus simple de comparer le nombre décimal équivalent au jeu d'indicateurs binaires, soit ici à 137 pour les indicateurs binaires 00, 03 et 07.

Utilisation de `RCLFLAG` et `STOFLAG`

Vous pouvez sauvegarder l'état des indicateurs binaires 00 à 43 au moyen de `RCLFLAG`, puis les restituer en partie ou en totalité au moyen de `STOFLAG`. Ces deux fonctions sont inséparables : le résultat de `RCLFLAG` ne peut servir qu'à `STOFLAG` et `STOFLAG` n'opère que sur des données obtenues par `RCLFLAG`.

L'exécution de `RCLFLAG` (*recall flags*) rappelle l'état des indicateurs binaires 00 à 43 dans le registre X. Cet état, dont l'affichage est sans signification, peut être stocké dans un registre ou en mémoire étendue. Comme `RCL`, `RCLFLAG` fait monter la pile, à moins que son déplacement soit inhibé.

L'exécution de `STOFLAG` (*restore flags*) restitue tout ou partie de l'ensemble des indicateurs binaires 00 à 43 :

- Pour restituer l'ensemble des indicateurs binaires 00 à 43, rappelez dans X la donnée d'état et exécutez `STOFLAG` .
- Pour restituer le bloc des indicateurs binaires de *dd* (*début*) à *ff* (*fin*), rappelez dans le registre X la donnée d'état, placez dans le registre X le nombre *dd.ff* (ce qui déplace dans le registre Y la donnée d'état) et exécutez `STOFLAG` . Comme `STO` , `STOFLAG` ne déplace pas la pile. Si $dd \geq ff$, seul l'indicateur binaire *dd* est restitué.

Exemple. Soit un programme qui arrondit les nombres à l'entier le plus proche (avec une instruction `RND`) et un format d'affichage `FIX 0`, mais garde par ailleurs le format d'affichage choisi par l'utilisateur.

1. Une fois que l'utilisateur a choisi son format d'affichage, exécutez `RCLFLAG` pour rappeler dans le registre X l'état des indicateurs binaires 00 à 43, puis `STO nn` pour le stocker dans le registre R_{nn} .
2. Spécifiez aucune partie décimale par `FIX 0`.
3. Arrondissez le nombre contenu dans le registre X par `RND` .
4. Avant d'obtenir une sortie, exécutez `RCL nn` pour rappeler l'état des indicateurs binaires, spécifiez les indicateurs binaires 36 à 41 (format de l'affichage) en tapant **36.41**, puis exécutez `STOFLAG` pour restituer aux indicateurs binaires 36 à 41 leurs états précédents.

Les entrées et sorties peuvent avoir lieu en tout format, tandis que `FIX 0` est le format effectif sur demande.

Copies multiples d'indicateurs binaires. Vous pouvez utiliser `RCLFLAG` et `STOFLAG` plutôt que `X<>F`, avec les différences suivantes :

- Tout groupe d'indicateurs binaires peut être un bloc quelconque, y compris le bloc complet des indicateurs binaires 00 à 10 (alors que pour `X<>F`, tout groupe est nécessairement composé des indicateurs binaires 00 à 07).
- Vous ne pouvez pas utiliser directement la donnée obtenue par `RCLFLAG`, sauf pour restituer aux indicateurs binaires réels leurs valeurs précédentes (alors que le dernier exemple utilisait directement un nombre obtenu par `X<>F`).

Branchements

Table des matières

Introduction	298
Branchement à un label	299
Labels globaux	299
Recherche de labels globaux	299
Labels locaux	299
Recherche de labels locaux	300
Multiplats utilisés par <code>[GTQ]</code>	300
Appel d'un sous-programme	301
Pile de retours après sous-programme	302
Recherche de labels globaux dans un sous-programme	303
Multiplats utilisés par <code>[XEQ]</code>	303
Fonctions conditionnelles	303
Test d'indicateur binaire	304
Comparaisons	304
Boucles	305
Boucle et fonction conditionnelle	305
Fonctions de commande de boucle	306

Introduction

Il y a branchement chaque fois qu'un programme ne s'exécute pas en séquence, c'est-à-dire qu'après une instruction il saute au moins une instruction. Deux types de fonctions provoquent un branchement :

- `[GTQ] label` et `[XEQ] label` provoquent le saut au label spécifié.
- Un test d'indicateur binaire, une comparaison ou une fonction de commande de boucle, selon le résultat, provoquent ou non le saut de l'instruction suivante.

Ces deux types de fonctions opèrent souvent conjointement : le test d'un indicateur binaire peut être suivi par un `[GTQ] label`, afin que son état détermine s'il doit y avoir saut au label spécifié. Ce chapitre décrit l'utilisation de `[GTQ]`, de `[XEQ]`, des fonctions conditionnelles et des boucles.

Branchement à un label

L'unique objet des labels est de servir de cible aux instructions de branchement. Il y a deux types de labels. Les *labels globaux* sont accessibles de tout programme résidant en mémoire principale. Les *labels locaux* ne sont accessibles que de l'intérieur du programme où ils figurent. Tout label autre qu'un label local alphabétique est accessible directement et indirectement.

Labels globaux

Les labels globaux comprennent au maximum 7 caractères alphanumériques (lettres et chiffres « alphabétiques » ; virgule, point et deux points sont illicites). Les lettres seules, **A** à **J** et **a** à **e**, forment des labels alphabétiques locaux ; les labels formés d'une seule autre lettre ou chiffre sont du type global. Les labels globaux occupent quatre multipliets de mémoire programme, plus un multipliet par caractère.

Un label global identifie la ligne à laquelle il se trouve et le programme dans lequel il se trouve. Les fonctions qui agissent sur un programme entier (comme `CLP` et `SAVEP`) accèdent au programme désiré par un de ses labels globaux. Vous pouvez accéder de l'extérieur à la ligne d'un programme qui est repérée par un label global.

Recherche de labels globaux

Lorsque le HP-41 exécute un `GTO label global`, la recherche commence en mémoire programme, de la fin vers le début du catalogue 1. Elle s'arrête à la première coïncidence, si bien qu'un label homonyme se trouvant « avant » dans le catalogue 1 ne peut être trouvé.

Si le HP-41 n'a pas trouvé le label spécifié avant d'avoir atteint le début de la mémoire programme, il poursuit dans le catalogue 2. Si le label global recherché se trouve dans un module enfichable ou dans un périphérique, l'exécution est transférée dans ce module ou ce périphérique à la ligne contenant le label global.

Labels locaux

Les labels locaux sont des repères propres à un programme. Ils servent aux branchements à l'intérieur du programme. Il y a deux types de labels locaux numériques et un type de label local alphabétique.

Labels locaux numériques. Les deux types se distinguent par la distance de branchement dans un programme.

- Les labels 00 à 14 sont du type *courte-distance*. Ils n'occupent qu'un multipliet en mémoire programme, et permettent un branchement à une distance (entre l'instruction `GTO` et le label) inférieure à 112 multipliets.
- Les labels 15 à 99 sont du type *longue-distance*. Ils occupent deux multipliets en mémoire programme, et permettent un branchement à toute distance à l'intérieur d'un programme.

Labels locaux alphabétiques. Ils occupent deux multipléts de mémoire programme et permettent un branchement à toute distance à l'intérieur d'un programme. Ils sont conçus pour l'exécution manuelle : lorsque le clavier personnel est validé, un label local alphabétique est affecté automatiquement à chacune des touches des deux premières rangées du clavier (cf. au chapitre 9 « Les deux rangées supérieures de touches »). Vous pouvez ainsi utiliser ces touches pour exécuter le programme à partir des lignes repérées par les labels locaux correspondants.

Exemple de programme. Le programme SIGMA du chapitre 22 affiche **CLEAR D, J? E**, ce qui propose trois solutions : appuyer sur **[D]** (*days*) pour effacer les jours, **[J]** (*job*) pour effacer une tâche, ou **[E]** (*exit*) pour terminer. Quand vous appuyez sur l'une de ces touches, l'exécution du programme se poursuit à la ligne désignée par le label local alphabétique.

Recherche de labels locaux

La recherche de labels locaux est limitée au programme courant. Elle commence à l'instruction **[GTO]** et se poursuit en séquence jusqu'à la fin du programme. Si elle n'a rien trouvé, elle reprend au début du programme. La recherche dans un programme long peut être assez longue. C'est pourquoi le HP-41 enregistre la distance en mémoire programme séparant l'instruction **[GTO]** du label recherché à la première exécution de **[GTO]**.

Exemple de programme. Pour économiser la mémoire programme, le programme SIGMA du chapitre 22 emploie des labels locaux à courte distance chaque fois que c'est possible. Comme SIGMA a besoin de 28 labels locaux à courte distance et que le HP-41 en dispose de 15, les labels 00 à 03 servent plusieurs fois en limitant leur emploi aux branchements « en avant », c'est-à-dire où **[GTO]** précède **[LBL]**. Ainsi, **LBL 00** apparaît 5 fois, chaque **GTO 00** est destiné au **LBL 00** suivant, et aucune confusion n'est possible.

En revanche, les labels à courte distance 04 à 14 servent ici aux branchements « en arrière », c'est-à-dire où **[GTO]** suit **[LBL]**. Ils ne doivent apparaître qu'une fois puisque l'instruction **[GTO]** recherche en séquence jusqu'à la fin du programme, puis à partir du début jusqu'au label.

Multipléts utilisés par **[GTO]**

Le nombre de multipléts de mémoire programme utilisés par une instruction **[GTO]** dépend du type de label spécifié dans cette instruction :

- Label global de n caractères : $2 + n$ multipléts.
- Label local numérique « longue-distance » ou alphabétique : 3 multipléts.
- Label local numérique « courte-distance » ou adressage indirect : 2 multipléts.

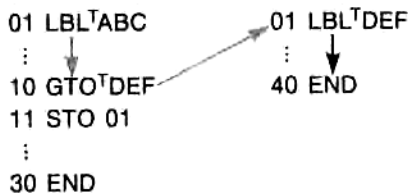
Appel d'un sous-programme

L'objet de l'instruction **XEQ** est d'appeler un sous-programme. **XEQ** et **GTO** ont les ressemblances suivantes :

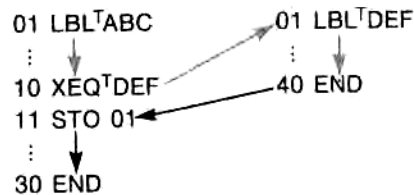
- Transfert de l'exécution à la ligne repérée par le label.
- Tous les types de labels reconnus par **GTO** sont aussi reconnus par **XEQ**.

XEQ diffère de **GTO** en ce que, après exécution complète du programme, c'est-à-dire à la première instruction **RTN** ou **END**, l'exécution reprend dans le programme principal à l'instruction qui suit immédiatement l'instruction **XEQ**, ainsi que le montrent les schémas suivants.

Le programme ABC branche vers le programme DEF. L'exécution s'arrête à l'instruction **END** de DEF.

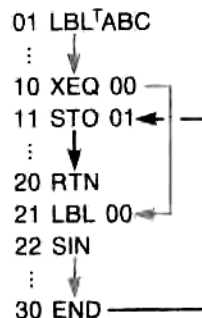


Le programme ABC appelle le programme DEF en tant que sous-programme. Après exécution du **END** de DEF, l'exécution reprend à la ligne 11 de ABC.



L'emploi d'un sous-programme économise la mémoire programme : le sous-programme peut être exécuté plusieurs fois par le même programme (label local) ou par d'autres programmes (labels globaux).

RTN et **END** commandent toutes deux l'exécution de l'instruction qui suit immédiatement celle qui a provoqué le branchement au sous-programme. Cependant, **END** marque la fin d'un programme et affecte la recherche d'un label local et les fonctions qui opèrent sur des programmes entiers ; **RTN** ne marque que la fin d'un sous-programme à l'intérieur d'un programme. Dans l'exemple suivant, **END** termine le sous-programme, et **RTN** le programme. (Il n'y a aucune raison d'exécuter les lignes 22 à 29 sous forme de sous-programme, puisqu'elles ne sont exécutées qu'une seule fois.)



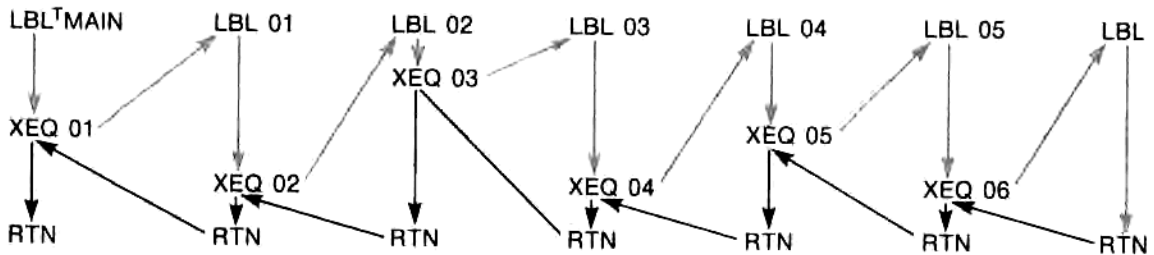
Si vous appelez **ABC** en tant que sous-programme à partir d'un autre programme, l'exécution revient au programme appelant après exécution de **20 RTN**. C'est-à-dire que si un programme appelle un sous-programme qui appelle un second sous-programme, l'exécution passe du second sous-programme au premier sous-programme, puis au programme appelant.

Vous pouvez vous assurer que l'exécution s'arrêtera à la ligne 20, même si **ABC** est appelé en tant que sous-programme, en introduisant l'instruction **20 STOP**. En mode programme, appuyez sur **[RS]** pour introduire une instruction **[STOP]**.

Pile de retours après sous-programme

Quand une instruction **[XEQ]** appelle un sous-programme, le HP-41 mémorise l'adresse de retour (c'est-à-dire celle qui suit immédiatement **[XEQ]**) dans une pile de retour après sous-programme. L'adresse de retour est retirée de la pile dès le retour effectué.

Limites. Quand un sous-programme appelle un autre sous-programme, toutes les adresses de retour en attente dans la pile sont décalées vers le haut. La capacité de la pile est de 6 adresses, si bien que le HP-41 accepte 6 niveaux de sous-programme.



Perte d'adresse de retour. Les adresses en attente dans la pile sont perdues dans les conditions suivantes :

- Si la pile contient déjà 6 adresses de retour au moment de l'appel d'un sous-programme, la première adresse est perdue, si bien que l'exécution ne revient jamais à l'instruction **[XEQ]** qui a appelé le premier sous-programme, mais s'arrête à la fin du premier sous-programme puisqu'il n'y a plus d'adresse de retour dans la pile.
- Toutes les adresses de retour sont perdues en cas d'exécution manuelle du programme. Ainsi, si vous arrêtez le programme **ABC** au milieu d'un sous-programme et exécutez manuellement un programme **DEF**, il est impossible de retourner à **ABC**. Cette règle s'applique aussi lorsqu'une alarme exécute **DEF** alors que **ABC** est arrêté; par exemple, l'exécution d'un label local alphabétique au moyen d'une touche du clavier utilisateur efface la pile des adresses de retour.

Recherche de labels globaux dans un sous-programme

Quand le HP-41 exécute **XEQ** avec label global, il recherche d'abord dans la mémoire programme, comme le fait **RTD**. Cependant, la recherche diffère s'il n'a pas trouvé le label global en mémoire programme : elle se poursuit dans les catalogues 1, 2 et 3.

Recherche dans le catalogue 1. Elle commence par la fin et s'arrête à la première coïncidence. L'exécution reprend à ce label.

Recherche dans le catalogue 2. Si le label n'est pas trouvé en mémoire programme, le HP-41 recherche dans le catalogue 2 un label ou un nom de fonction qui coïncide avec le label spécifié. (Cf. à l'annexe H une description détaillée du contenu du catalogue 2.) L'exécution reprend à ce label, ou la fonction trouvée est exécutée.

Recherche dans le catalogue 3. Si le label n'est pas trouvé dans le catalogue 2, le HP-41 recherche dans le catalogue 3 un nom de fonction qui coïncide avec le label spécifié. S'il le trouve, il l'exécute ; sinon, il affiche le message **NOMEXISTENT**.

Multiplats utilisés par **XEQ**

Le nombre de multiplats de mémoire programme utilisés par une instruction **XEQ** dépend du type de label utilisé dans cette instruction :

- Label global de n caractères : $2 + n$ multiplats.
- Label local : 3 multiplats.
- Adressage indirect : 2 multiplats.

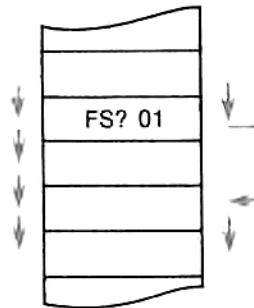
Fonctions conditionnelles

Le test d'indicateurs binaires et les comparaisons sont des fonctions conditionnelles : selon les conditions présentes, leur proposition est vraie ou fausse, ce qui décide de leur effet.

- Si vous exécutez manuellement une fonction conditionnelle, le HP-41 affiche **YES** (oui) si la proposition est vraie à cet instant et **NO** (non) si elle est fausse à cet instant.
- Si un programme exécute une fonction conditionnelle, il passe à la première ligne la suivant si la condition est vraie ou à la seconde ligne la suivant si la condition est fausse.

L'indicateur binaire 01 est-il armé ?

Si oui (YES), exécution de la ligne 01



Si non (NO), exécution de la ligne 02

Test d'indicateur binaire

Les fonctions suivantes testent tous les indicateurs binaires :

FS? *nn* (*flag set?*) : l'indicateur binaire *nn* (de 00 à 55) est-il armé ?

FC? *nn* (*flag clear?*) : l'indicateur binaire *nn* (de 00 à 55) est-il désarmé ?

Deux fonctions testent et désarment les indicateurs binaires 00 à 29 (elles ne peuvent agir sur les indicateurs binaires du système, 30 à 55) :

FS?C *nn* (*flag set? clear*) : l'indicateur binaire *nn* (de 00 à 29) est-il armé ? le désarmer.

FC?C *nn* (*flag clear? clear*) : l'indicateur binaire *nn* (de 00 à 29) est-il désarmé ? l'armer.

Comparaisons

Comparaison de X à zéro. Les 5 fonctions suivantes comparent le contenu *x* du registre X à zéro :

X<0?

X<-0?

X=0?

X≠0?

X>0?

Comparaison de X à Y. Les 5 fonctions suivantes comparent le contenu *x* du registre X au contenu *y* du registre Y.

X<Y?

X<=Y?

X=Y?

X>=Y?

X>Y?

Les fonctions **X<=Y?** et **X=Y?** permettent en outre la comparaison de données alphabétiques. Les trois autres ne le permettent pas et provoquent l'affichage du message d'erreur **ALPHA DATA** si l'un au moins des deux registres contient des données alphanumériques.

Comparaison de X à Y indirect. Les 6 fonctions suivantes comparent le contenu x du registre X au contenu du registre (R_{00} à R_{319} , X, Y, Z, T et LAST X) dont l'adresse est spécifiée dans Y par 00 à 319, X, Y, Z, T ou L.

`[X < NN ?]` `[X <= NN ?]` `[X = NN ?]` `[X ≠ NN ?]` `[X >= NN ?]` `[X > NN ?]`

Ces fonctions opèrent sur toute combinaison de données alphabétiques et numériques. Les chaînes alphanumériques sont comparées d'après les codes des caractères (strictement supérieurs à ceux des chiffres).

Boucles

Une boucle est une séquence d'instructions qui commence à un label et finit à un branchement à ce même label. La plus simple est la boucle infinie suivante :

```
01 LBL^BOUCLE
02 BEEP
03 GTO^BOUCLE
04 END
```

Ce programme doit s'exécuter indéfiniment, jusqu'à épuisement de l'alimentation électrique. Les boucles infinies sont à éviter, mais celles qui comprennent une condition leur permettant de s'achever sont un puissant outil de programmation.

Boucle et fonction conditionnelle

Si vous désirez effectuer une opération jusqu'à ce qu'une certaine condition soit satisfaite, mais ne savez pas exactement combien de fois il convient de répéter l'opération, créez une boucle avec fonction conditionnelle juste avant le `[GTO]`. Par exemple, le programme suivant retire 1 d'un nombre, teste le résultat, et répète l'opération tant que celui-ci est positif. Dès qu'il est nul (en supposant que le nombre de départ était positif), le programme sort de la boucle, émet une tonalité, et s'arrête.

```
01 LBL^ABC
02 1
03 -
04 X > 0 ?
05 GTO^ABC
06 BEEP
07 END
```


Exemple de programme. Les programmes TR et SIGMA du chapitre 22 emploient des boucles avec fonction conditionnelle de manière à accéder à la mémoire annexe. Quand de nombreux enregistrements du fichier TRECS doivent être traités d'une certaine façon, mais que leur nombre est variable, la boucle qui traite chaque enregistrement utilise l'indicateur binaire de non-prise en compte d'erreur (25) pour sortir quand tous les enregistrements ont été traités.

- Après traitement de chaque enregistrement, l'indicateur binaire 25 est armé, ce qui signifie qu'à la prochaine erreur, le HP-41 désarme l'indicateur binaire 25 mais *n'arrête pas* l'exécution.
- L'indicateur binaire 25 étant armé, le programme tente d'accéder à l'enregistrement suivant, puis teste l'indicateur binaire 25. Si l'indicateur binaire 25 a été désarmé, cela signifie qu'il n'y a plus d'enregistrements (erreur END OF FL), et le programme sort de la boucle. Si l'indicateur binaire 25 est encore armé, cela signifie que l'enregistrement suivant existe, et la boucle est répétée afin de le traiter.

Le programme SIGMA illustre cette technique. Les lignes 351 à 362 forment une boucle qui accède à chaque enregistrement de temps. Dans la boucle, les lignes 354 à 360 tentent de placer le pointeur à l'enregistrement suivant; si `SEEKPT` (ligne 358) provoque une erreur, `UTD 12` (ligne 360) achève la boucle.

Fonctions de commande de boucle

Lorsque vous voulez exécuter une fonction un nombre de fois déterminé, vous pouvez utiliser des fonctions spéciales au lieu des fonctions conditionnelles des exemples précédents : `ISG` (*increment, skip if greater* = incrémentation, saut si supérieur) et `DSE` (*decrement, skip if equal* = décrémentation, saut si égal). Ces deux fonctions utilisent un argument numérique stocké dans un registre (mémoire principale, pile ou LAST X), et adressé directement ou indirectement. Son format est : `iiii.fffcc`, où :

- **iiii est la valeur courante du compteur** (1 à 5 chiffres), incrémentée par `ISG` ou décrémentée par `DSE` de la valeur `cc`.
- **fff est la valeur finale du compteur** (3 chiffres obligatoirement : 009, 057, 811 par exemple) à laquelle `iiii` est comparée à chaque exécution de `ISG` ou `DSE`.
- **cc est la valeur du pas de l'incrémentation ou de la décrémentation** (2 chiffres obligatoirement : 02, 15 par exemple). Si `cc` vaut 00 (valeur sans intérêt) ou n'est pas spécifié, le HP-41 prend un pas de 01.

Quand le HP-41 exécute `ISG`, il commence par incrémenter `iiii` de `cc`, puis teste si la nouvelle valeur de `iiii` est plus grande que `fff`. Si oui, il saute l'instruction suivante.

Quand le HP-41 exécute `DSE`, il commence par décrémenter `iiii` de `cc`, puis teste si la nouvelle valeur de `iiii` est égale (ou inférieure) à `fff`. Si oui, il saute l'instruction suivante.

Exemple de programme. Le programme SIGMA du chapitre 22 utilise les fonctions de commande de boucle pour accéder à un bloc de registres successifs. Quand ceux-ci sont remplis de données rappelées de la mémoire annexe (modules d'extension mémoire), ils sont lus par ordre croissant d'adresses. Quand les données sont affichées, on accède aux registres par ordre décroissant d'adresses (l'ordre des données doit être inversé afin de les afficher chronologiquement).

Pour remplir les registres en ordre croissant d'adresses, SIGMA emploie `INC` (ligne 213) pour incrémenter le nombre de commande de boucle (dans R_{05}). Pour afficher les données en ordre inverse, SIGMA emploie `DEC` (lignes 231 et 259) pour décrémenter le nombre de commande de boucle.

SIGMA définit les valeurs initiale et finale des compteurs comme étant les adresses du premier et du dernier registre du bloc. Puis, à chaque exécution de la boucle, la partie entière du nombre de commande de boucle donne l'adresse du registre auquel on désire accéder. Cet usage du nombre de commande de boucle pour l'adressage indirect (lignes 209, 212, 228 et 254) dépend de la valeur initiale suivante :

- `iiii` est l'adresse du premier à adresser indirectement.
- `fff` est l'adresse du dernier à adresser indirectement.
- `cc` n'étant pas spécifié, il a par défaut la valeur 01.

Opérations alphabétiques et interactions

Table des matières

Introduction	308
Registre ALPHA et registre X	309
Traduction de caractères et de nombres	309
Rappel d'un nombre du registre ALPHA	311
Manipulation de chaînes alphabétiques	312
Recherche et rotation dans le registre ALPHA	312
Recherche de la longueur de la chaîne	313
Exemple de manipulations dans le registre ALPHA	314
Demande d'intervention	314
Utilisation de <code>PROMPT</code>	314
Utilisation de <code>PSE</code>	315
Réponse à l'appui sur une touche	317
Utilisation de <code>GETKEY.c</code>	317
Utilisation de <code>GETKEY</code>	317
Sorties	318
Utilisation de <code>AVIEW</code>	319
Utilisation de <code>VIEW</code>	319
Utilisation de <code>PSE</code>	319
Utilisation de <code>TONE</code> et <code>DECP</code>	319

Introduction

Ce chapitre décrit deux caractéristiques du registre ALPHA (alphabétique) : manipulation de données et interactions entre l'utilisateur et un programme.

- Les manipulations dans le registre ALPHA comprennent ses utilisations classiques (telles que messages à afficher et données pour fichiers de texte), et offrent en outre des caractères supplémentaires et la possibilité de stocker des multipléts arbitraires de données.
- L'interaction entre l'utilisateur et un programme comprend les fonctions qui affichent un message et celles qui interprètent la réponse de l'utilisateur.

Registre X et registre ALPHA

Il y a trois façons de transférer des données entre les registres X et ALPHA :

1. L'exécution de **[ARCL]** X copie le contenu du registre X dans le registre ALPHA. **[ASTO]** X copie 6 caractères du registre X dans le registre ALPHA (les chiffres placés dans le registre X par **[ARCL]** sont assimilés à des caractères et ne peuvent servir dans les calculs). Les fonctions **[ASTO]** et **[ASTG]**, qui accèdent en général à des registres de données, sont décrites au chapitre 12, paragraphe « Mémoire principale ».
2. L'exécution de **[XTOA]** (X to ALPHA = X vers ALPHA) convertit un nombre du registre X en un caractère dans le registre ALPHA. Inversement, **[ATOX]** (A to X = A vers X) convertit un caractère du registre ALPHA en son équivalent décimal dans le registre X.
3. L'exécution de **[ANUM]** (alpha number = nombre alphabétique) recherche dans le registre ALPHA une chaîne de chiffres « alphabétiques » (inutilisables dans les calculs) et la convertit en un nombre véritable dans le registre X. Ceci est la seule manière de prendre des chiffres dans le registre ALPHA afin de les utiliser dans un calcul.

Traduction de caractères et de nombres

Le registre ALPHA peut contenir une plus grande variété de caractères que le HP-41 ne peut en afficher ; et l'affichage peut montrer une plus grande variété de caractères que ceux que vous pouvez taper au clavier alphabétique. Les fonctions **[ATOX]** et **[XTOA]** permettent de tirer tout le parti possible du registre ALPHA et de l'affichage.

Représentation de nombres et de caractères par des multipliets. Le chapitre 19 décrit comment un multiplet peut présenter à la fois l'état des indicateurs binaires 00 à 07 et un nombre décimal de 0 à 255. Un multiplet peut aussi représenter un caractère stocké dans le registre ALPHA. Bien que certains multipliets ne puissent être affichés en tant que caractères, ils peuvent être stockés dans le registre ALPHA lui-même. La possibilité qu'a le HP-41 d'opérer sur des multipliets est particulièrement appréciable pour la commande de périphériques. Le multiplet nul 00000000 (d'équivalent décimal 0), possède dans le registre ALPHA une signification spéciale. C'est pourquoi, dans certaines circonstances, vous ne pouvez pas recouvrer un multiplet nul du registre ALPHA ; ces restrictions sont décrites à l'annexe C au paragraphe « Caractères nuls ». Tous les autres multipliets peuvent être librement stockés, manipulés et rappelés du registre ALPHA.

Utilisation de **[XTOA].** Le registre X contenant un nombre de 0 à 255, l'exécution de **[XTOA]** annexe le multipliant équivalent à droite dans le registre ALPHA. Le contenu du registre X est inchangé. Si le registre X contient une chaîne alphabétique, l'exécution de **[XTOA]** annexe cette chaîne au contenu du registre ALPHA. A noter qu'une chaîne de chiffres alphabétiques située dans le registre X est annexée sous la forme de ces chiffres et non du multipliant équivalent.

Le tableau suivant présente tous les caractères affichables. Certains d'entre eux ne sont pas disponibles sur le clavier alphabétique. Vous pouvez néanmoins les annexer dans le registre ALPHA en plaçant dans le registre X leur équivalent décimal et en exécutant **[XTOA]**. Tous les codes décimaux 128 à 255 provoquent l'apparition de tous les segments de l'afficheur. Le tableau indique aussi les caractères ASCII affichables (et pas les caractères de commande) correspondant à un nombre décimal.

Codes des caractères

Code	ASCII	Affichage	Code	ASCII	Affichage	Code	ASCII	Affichage	Code	ASCII	Affichage
0			32	space		64	@	Ⓐ	96		`
1		!	33	!	!	65	A	Ⓐ	97	a	ⓐ
2		"	34	"	"	66	B	Ⓑ	98	b	ⓑ
3		#	35	#	#	67	C	Ⓒ	99	c	ⓒ
4		\$	36	\$	\$	68	D	Ⓓ	100	d	ⓓ
5		%	37	%	%	69	E	Ⓔ	101	e	ⓔ
6		&	38	&	&	70	F	Ⓕ	102	f	ⓕ
7		'	39	'	'	71	G	Ⓖ	103	g	ⓖ
8		(40	((72	H	Ⓗ	104	h	ⓗ
9)	41))	73	I	Ⓘ	105	i	ⓘ
10		*	42	*	*	74	J	Ⓙ	106	j	ⓙ
11		+	43	+	+	75	K	Ⓚ	107	k	Ⓚ
12		,	44	,	,	76	L	Ⓛ	108	l	Ⓛ
13		-	45	-	-	77	M	Ⓜ	109	m	Ⓜ
14		.	46	.	.	78	N	Ⓝ	110	n	Ⓝ
15		/	47	/	/	79	O	Ⓞ	111	o	Ⓞ
16		0	48	0	0	80	P	Ⓟ	112	p	Ⓟ
17		1	49	1	1	81	Q	Ⓠ	113	q	Ⓠ
18		2	50	2	2	82	R	Ⓡ	114	r	Ⓡ
19		3	51	3	3	83	S	Ⓢ	115	s	Ⓢ
20		4	52	4	4	84	T	Ⓣ	116	t	Ⓣ
21		5	53	5	5	85	U	Ⓤ	117	u	Ⓤ
22		6	54	6	6	86	V	Ⓥ	118	v	Ⓥ
23		7	55	7	7	87	W	Ⓦ	119	w	Ⓦ
24		8	56	8	8	88	X	Ⓧ	120	x	Ⓧ
25		9	57	9	9	89	Y	Ⓨ	121	y	Ⓨ
26		:	58	:	:	90	Z	Ⓩ	122	z	Ⓩ
27		;	59	;	;	91	[Ⓛ	123	{	Ⓛ
28		<	60	<	<	92	\	Ⓜ	124		Ⓜ
29		=	61	=	=	93]	Ⓨ	125	}	Ⓨ
30		>	62	>	>	94	^	Ⓩ	126	~	Ⓩ
31		?	63	?	?	95	_	Ⓩ	127		Ⓩ

Utilisation de `ATOX`. L'exécution de `ATOX` place dans le registre X l'équivalent décimal du multiplet le plus à gauche du registre ALPHA. Ce multiplet est perdu puisque le contenu du registre ALPHA se décale à gauche; la pile se décale vers le haut, à moins que son déplacement soit invalidé. Si le registre ALPHA est vide, `ATOX` place zéro dans le registre X.

Tout multiplet nul situé à gauche du registre ALPHA est ignoré par `ATOX`. Un multiplet nul figurant dans une chaîne alphabétique disparaît quand il atteint l'extrémité gauche du registre ALPHA lors d'un décalage ou d'une rotation. Vous pouvez cependant détecter la disparition d'un multiplet nul en comparant la longueur de la chaîne alphabétique avant et après décalage ou rotation du registre ALPHA, comme le décrit le paragraphe « Recherche de la longueur d'une chaîne », page 313.

Exemple de programme. Les programmes TR et SIGMA du chapitre 22 emploient `XFOA` et `ATOX` pour coder le nombre d'heures passées chaque jour sur une tâche. Les multiplets dont les équivalents décimaux sont 1 à 240 représentent de 0,1 à 24,0 heures. Ceci permet un stockage économique des données : 7 multiplets, représentant les données d'une semaine, n'occupent qu'un registre de mémoire annexe dans le fichier de texte TRECS. La solution qui consiste à stocker les heures de chaque jour dans un registre offre une plus grande précision mais occupe 7 fois plus de place en mémoire.

Rappel d'un nombre du registre ALPHA

La fonction `ANUM` recherche les caractères alphabétiques dans le registre ALPHA. Si elle en trouve, elle arme l'indicateur binaire 22 (entrée de données numériques) et place dans le registre X un nombre véritable, utilisable pour des calculs. Si elle n'en trouve pas, l'indicateur binaire 22 et le registre X sont inchangés.

Si vous placez dans le registre ALPHA deux nombres sans les séparer, `ANUM` ne les distingue pas. Si vous placez dans le registre ALPHA des caractères arbitraires, `ANUM` identifie puis évalue la chaîne de chiffres.

La chaîne est identifiée ainsi :

- `ANUM` effectue sa recherche dans le registre ALPHA à partir de l'extrémité gauche.
- Les caractères sont ignorés jusqu'à la rencontre d'un chiffre. Si une ponctuation (cf. définition de l'indicateur binaire 28) ou un signe moins le précède immédiatement, ils sont considérés comme faisant partie de la chaîne.
- Tous les signes moins et marques de ponctuation suivants ainsi que 10 chiffres au maximum sont considérés comme faisant partie de la chaîne.
- Les lettres **E** (exposant) sont considérées comme faisant partie de la chaîne si elles sont suivies de chiffres.
- Le séparateur (cf. définition de l'indicateur binaire 28) est considéré comme faisant partie de la chaîne uniquement si l'indicateur binaire 29 (groupement de chiffres) est à 1.
- `ANUM` considère que la chaîne est terminée dès la rencontre du premier caractère ne satisfaisant pas aux conditions énumérées ci-dessus.
- `ANUM` arrête sa recherche de chiffres à la rencontre du multiplet nul.

Quand `ANIM` identifie une chaîne, il l'évalue ainsi :

- Une chaîne simple, par exemple **12345**, donne le nombre vrai correspondant ($x = 12345$).
- La première lettre **E** rencontrée fait considérer les 1 ou 2 chiffres suivants comme constituant un exposant. Toutes les lettres **E** suivantes sont ignorées. Si la chaîne comporte déjà plus de 8 chiffres, la lettre **E** n'est pas prise en compte, à moins qu'un signe de ponctuation ne précède le huitième chiffre.
- x est positif si la chaîne comporte un nombre pair de signes moins. x est négatif si la chaîne comporte un nombre impair de signes moins. Cette règle s'applique aussi à la sous-chaîne de l'exposant.
- Le premier signe de ponctuation sépare partie entière et partie décimale de x . Tous les signes de ponctuation suivants, ainsi que ceux de la sous-chaîne suivant **E**, ne sont pas pris en compte.
- Si l'indicateur binaire 29 est armé, aucun séparateur n'est pris en compte en tant que tel, mais est considéré comme faisant partie intégrante de la chaîne. Si l'indicateur binaire 29 est désarmé, `ANIM` considère que la chaîne se termine au premier séparateur rencontré.

Manipulation de chaînes alphabétiques

Les données placées dans le registre ALPHA peuvent provenir de plusieurs sources : vos propres fichiers de texte, un périphérique HP-IL ou le clavier alphabétique. Quelle que soit la source, vous pouvez combiner les opérations élémentaires décrites ci-après afin d'effectuer des manipulations complexes de ces informations.

Recherche et rotation dans le registre ALPHA

N'oubliez jamais que les 3 fonctions qui opèrent sur les données du registre ALPHA vers le registre X commencent par affecter l'extrémité gauche du registre ALPHA :

- `ASTD` X recopie dans le registre X les 6 multiplats de gauche.
- `ATOX` convertit et place dans le registre X le multiplat de gauche.
- `ANUM` identifie la chaîne numérique située le plus à gauche, la convertit en nombre véritable, qu'elle place dans le registre X.

Pour opérer sur d'autres données du registre ALPHA que celles situées à gauche, vous pouvez effectuer une rotation de son contenu afin de placer le multiplat voulu à gauche. `POSA` (position dans ALPHA) permet de localiser la donnée voulue dans le registre ALPHA. `AROT` (ALPHA rotate = rotation dans ALPHA) permet d'effectuer une rotation dans le registre ALPHA.

Recherche du registre ALPHA. `POSA` recherche dans le registre ALPHA une donnée spécifiée dans le registre X. Le résultat est un nombre représentant sa position, placé dans le registre X. Le registre ALPHA est inchangé.

La donnée recherchée peut être :

- L'équivalent décimal (0 à 255) d'un multiplet; ou
- Une chaîne alphabétique placée dans le registre X par `ASIO` X.

Les positions dans le registre ALPHA sont comptées de gauche à droite et débutent par le numéro zéro. La position d'une chaîne alphabétique est définie comme étant la position de son premier caractère. Si la chaîne recherchée est trouvée, sa position remplace dans le registre X sa propre spécification. Si elle n'est pas trouvée, le registre X contient - 1.

L'exécution de `POSA` ne trouve que la première apparition d'une chaîne. Si la chaîne recherchée est le multiplet nul (équivalent décimal : zéro), le HP-41 ne trouve qu'un multiplet nul suivant un multiplet non nul.

Rotation dans le registre ALPHA. `AROT` effectue une rotation du contenu du registre ALPHA du nombre de positions spécifié dans le registre X. La rotation a lieu vers la gauche si x est positif ou vers la droite si x est négatif. Seul le contenu courant subit la rotation (et non les 24 positions du registre ALPHA). A noter qu'une exécution de `AROT` immédiatement après `POSA` effectue une rotation de la chaîne spécifiée jusqu'à l'extrémité gauche du registre ALPHA (si ladite chaîne est trouvée). Tous les multiplats nuls atteignant l'extrémité gauche du registre ALPHA disparaissent. Vous pouvez vérifier la disparition de multiplats nuls en suivant la longueur de la chaîne contenue dans le registre ALPHA.

Recherche de la longueur de la chaîne

L'exécution de `ALENG` (*ALPHA length* = longueur ALPHA) place dans le registre X le nombre de caractères contenu dans le registre ALPHA. Le registre ALPHA est inchangé; la pile se déplace vers le haut, à moins que son déplacement soit invalidé. Les situations décrites ci-après sont des indications d'emploi de `ALENG` en conjonction avec d'autres manipulations du registre ALPHA :

- Si vous effectuez plusieurs rotations du registre ALPHA afin de rappeler des nombres par `ANUM`, `ALENG` peut déterminer quand une rotation complète a eu lieu, c'est-à-dire quand le nombre de positions déplacées par rotation est égal au nombre de caractères contenu dans le registre ALPHA, donc quand le registre ALPHA retrouve sa situation initiale.
- Si vous déplacez vers le registre X des multiplats par `ATDX` puis les traitez par un sous-programme, `ALENG` vous permet de déterminer le nombre d'itérations nécessaires.
- Si vous effectuez un décalage ou une rotation sur une chaîne alphabétique qui inclut des multiplats nuls, `ALENG` vous permet de détecter combien de multiplats nuls ont éventuellement disparu en comparant la longueur de la chaîne avant et après manipulation.

Exemple de manipulations dans le registre ALPHA

Supposons qu'un périphérique place dans le registre ALPHA deux chaînes numériques séparées par un espace. Cet exemple montre comment rappeler les deux nombres, placer la première chaîne dans le registre X et la seconde dans le registre Y. Les commentaires se rapportent à la chaîne combinée **123.45 678.90**.

- | | |
|-----------------|--|
| 01 ANUM | La première chaîne (123.45) est placée dans le registre X. Le contenu du registre ALPHA est inchangé. |
| 02 32 | Le code décimal de l'espace (32) est placé dans le registre X afin de servir d'opérande aux deux instructions suivantes. 123.45 est déplacé dans le registre Y. |
| 03 XTOA | Un espace est annexé à la seconde chaîne, afin de maintenir leur séparation lors de la rotation.
Le registre ALPHA contient désormais 123.45 678.90 . |
| 04 POSA | Le nombre 6, représentant la position de l'espace séparateur, remplace 32 dans le registre X. |
| 05 AROT | Le contenu du registre ALPHA subit une rotation à gauche de 6 positions.
Il devient 678.90 123.45 . |
| 06 ANUM | La seconde chaîne (678.90) est placée dans le registre X. 6 passe dans le registre Y et 123.45 dans le registre Z. |
| 07 RCL Z | Le nombre 123.45 est rappelé dans le registre X et déplacé dans le registre Y. |

Demande d'intervention

Plusieurs fonctions et groupes de fonctions demandent à l'utilisateur des interventions de plusieurs types :

1. L'exécution du programme est-elle stoppée jusqu'à ce qu'une réponse intervienne, ou bien continue-t-elle de toute façon ?
2. Quels sont les types de réponses possibles ?

Utilisation de PROMPT

Lors de l'exécution de PROMPT, le HP-41 affiche le contenu du registre ALPHA et arrête l'exécution. Le message doit indiquer le type de réponse attendue : numérique, alphabétique, procédure, appui sur une touche, ou autre. A noter qu'il n'y a que deux façons de reprendre l'exécution d'un programme :

- Appuyer sur la touche **[RS]** pour reprendre l'exécution à la ligne qui suit **[PROMPT]**
- Appuyer sur une touche des deux rangées supérieures correspondant à un label local alphabétique, auquel reprend l'exécution.

Exemples de programmes. Les programmes TR et SIGMA du chapitre 22 emploient **[PROMPT]** dans plusieurs buts. En général, le but est de vous demander de confirmer une donnée affichée, soit numérique (heure ou date), soit alphabétique (nom de tâche). Vous pouvez confirmer la donnée affichée en appuyant simplement sur **[RS]**, ou bien la modifier puis appuyer sur **[RS]**. Les programmes emploient les indicateurs binaires 22 et 23 pour déterminer si vous avez tapé une autre valeur.

- Si vous tapez un nombre dans le registre X, l'indicateur binaire 22 (entrée de donnée numérique) est armé. Le programme peut le tester ultérieurement pour déterminer si un nombre a été tapé.
- Si vous tapez une chaîne alphabétique dans le registre ALPHA, l'indicateur binaire 23 (entrée de donnée alphabétique) est armé. Le programme peut le tester ultérieurement pour déterminer si une chaîne a été tapée.

Par exemple, quand TR demande quelle tâche vous commencez (lignes 135 à 146), il désarme l'indicateur binaire 23 avant d'afficher la question, puis le teste. De plus, il exécute **[AON]** avant **[PROMPT]**, et **[AOFF]** après **[PROMPT]**, afin de valider le clavier alphabétique durant l'arrêt de l'exécution.

Quand TR tente d'agrandir le fichier de texte TRECS alors qu'il n'y a plus de place en mémoire étendue, il utilise **[PROMPT]** pour afficher **NEED ROOM** (besoin de place) et arrêter son exécution (lignes 230 et 231). Votre réponse doit être une procédure, à savoir la réduction de la taille d'autres fichiers. Après action, appuyez sur **[RS]**; TR tente à nouveau d'agrandir TRECS.

Le programme SIGMA emploie **[PROMPT]** pour afficher **CLEAR D, J? E** et arrêter l'exécution (lignes 303 à 305). Quand vous appuyez sur **[D]**, **[J]** ou **[E]** l'exécution reprend au label correspondant (lignes 341, 308 ou 363).

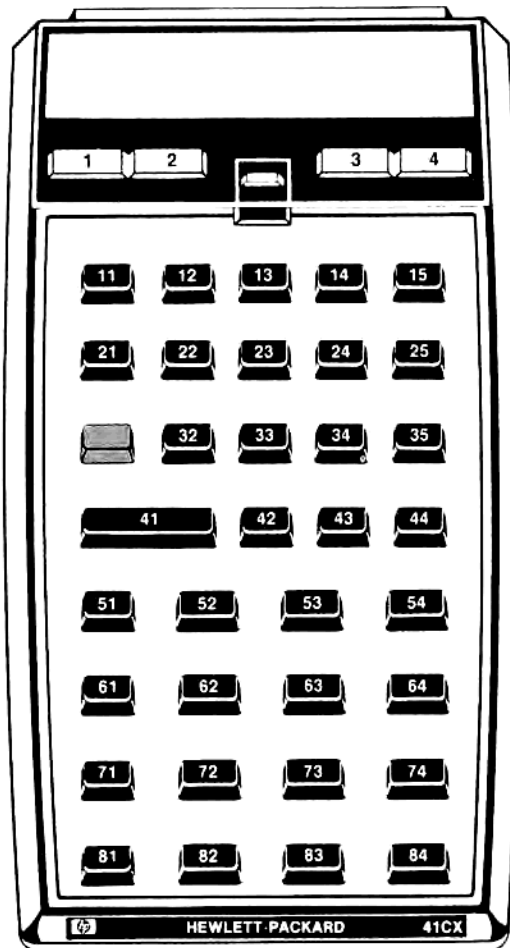
Utilisation de **[PSE]**


[PSE] (pause) est analogue à **[PROMPT]**, avec les différences suivantes :


- **[PSE]** retarde l'exécution d'un peu moins d'une seconde. La frappe d'une donnée numérique ou alphabétique durant une pause provoque la répétition de la pause. L'exécution d'une fonction arrête l'exécution du programme.
- **[PSE]** affiche normalement le registre X, ou le registre ALPHA si **[AVEW]** ou **[AON]** précède **[PSE]**

Une suite d'instructions **[PSE]** consécutives autorise davantage de temps pour commencer la frappe. A chaque exécution de **[PSE]**, le témoin **PRGM** clignote une fois. La frappe de données se termine à la fin de la pause. Si vous tapez quelques chiffres, attendez plus d'une seconde avant de taper d'autres chiffres pour qu'ils soient considérés comme formant un second nombre.

Codes des touches pour `GETKEY` et `GETKEYX`



`GETKEY` reporte dans le registre X un code de touche positif. L'appui sur  donne le code 31.

`GETKEYX` reporte dans le registre Y un code de touche positif ou négatif. L'appui sur  donne d'abord un code négatif pour la seconde touche pressée.

Réponse à l'appui sur une touche

Les deux fonctions `GETKEY` et `GETKEYX` attendent que vous appuyiez sur une touche, dont elles rapportent le code défini page 316.

Utilisation de `GETKEY`

`GETKEY` retarde l'exécution du programme jusqu'à 10 secondes. Si vous appuyez sur une touche pendant ce laps de temps, `GETKEY` rapporte dans le registre X le code de cette touche et décale la pile vers le haut, à moins que son déplacement soit invalidé. Si vous n'appuyez sur aucune touche, `GETKEY` rapporte zéro. Le programme peut afficher un message durant l'attente en exécutant `AVIEW` avant `GETKEY`.

A noter que les valeurs rapportées par `GETKEY` peuvent servir de labels numériques. `GETKEY` et `XEQ IND X` peuvent servir à brancher sur le sous-programme approprié à la réponse de l'utilisateur. Cette technique ressemble à celle des labels alphabétiques, mais elle n'efface pas la pile de retour des sous-programmes. Associez d'abord une lettre à chaque réponse, et trouvez le code de la touche de cette lettre. Puis faites commencer le sous-programme de chaque réponse par un label numérique qui est le code de la touche de la lettre. Si vous voulez que l'utilisateur appuie sur `Y` (*yes* = oui) ou `N` (non), faites commencer le sous-programme « oui » par `L5L 71` et le sous-programme « non » par `L5L 41`. Puis faites suivre `GETKEY` par `XEQ IND X`. Quand `GETKEY` est exécutée et que l'utilisateur appuie sur `Y` ou `N`, le nombre `71` ou `41` est rapporté dans le registre X, et `XEQ IND X` exécute le sous-programme approprié.

Utilisation de `GETKEYX`

`GETKEY` est analogue à `GETKEYX`, avec les particularités suivantes :

Intervalle variable. Le nombre *SS.s*, placé dans le registre X et tel que $SS.s \leq 99,9$, spécifie combien de secondes `GETKEYX` doit attendre une réponse.

Codes des touches avec et sans majuscule. `GETKEYX` rapporte un code de touche dans le registre Y. L'appui sur `■` puis une autre touche affecte d'un signe moins le second code. L'appui sur `■` fait recommencer l'intervalle spécifié ; si vous n'appuyez pas sur une seconde touche lors du second intervalle, `GETKEYX` rapporte zéro.

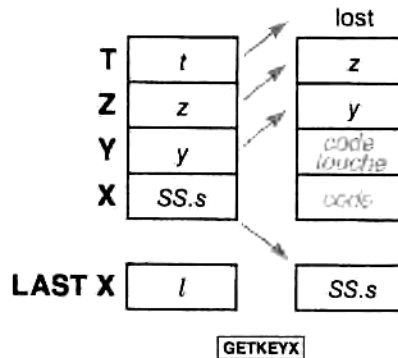
Codes de caractères. `GETKEYX` rapporte dans le registre X un code de caractère selon le tableau de la page 310. Si l'indicateur binaire 48 est armé (clavier ALPHA actif), le code rapporté est celui des lettres. S'il est désarmé, un code est rapporté pour les chiffres, le séparateur décimal et `CHS`, et zéro pour toutes les autres touches. Par exemple, la touche de code 52 donne la lettre R (code 82) sur le clavier ALPHA et le chiffre 7 (code 55) sur le clavier normal.

Action par pression ou relâchement de touche. Le signe de $SS.s$ dans le registre X détermine si le HP-41 agit au moment où une touche est pressée ou bien au moment où elle est relâchée :

- Si $SS.s \geq 0$, le HP-41 agit au moment du relâchement de la touche. C'est l'option la plus simple ; la fonction de la touche n'est jamais exécutée, et il y a une réponse à chaque appui sur la touche.
- Si $SS.s < 0$, le HP-41 agit au moment de la pression sur la touche, ce qui rend la répétition possible. Par exemple, un programme peut comporter une boucle qui place -0.01 dans le registre X, exécute `GETKEYX`, appelle un sous-programme approprié à la réponse de l'utilisateur, et recommence. L'exécution de `GETKEYX` avec une petite valeur négative pour $SS.s$ revient à demander : « y a-t-il une touche enfoncée ? » Si l'utilisateur maintient une touche enfoncée, la boucle contenant `GETKEYX` appelle le sous-programme approprié jusqu'à ce que la touche soit relâchée.

Le fait de relâcher la touche produit l'effet normal d'appuyer dessus ; mais seules les touches `R/S` et `ON` sont actives durant l'exécution d'un programme.

`GETKEYX` déplace les contenus des registres Y et Z dans Z et T, et $SS.s$ de X dans LAST X :



Sorties

Les fonctions suivantes permettent l'affichage d'un message et l'émission de tonalités.

Utilisation de `AVIEW`

L'exécution de `AVIEW` affiche le contenu du registre ALPHA jusqu'à ce que `CLD` (*clear display* = effacement affichage) efface l'affichage ou qu'un autre message soit affiché. `AVIEW` peut aussi arrêter l'exécution du programme, selon l'état des indicateurs binaires 21 (impression) et 55 (imprimante connectée) ; cf. annexe D.

Utilisation de `VIEW`

Pour afficher le contenu de R_n sans le rappeler dans X , exécutez `VIEW nn`. Le registre à examiner peut aussi être spécifié indirectement. Quand un programme exécute `VIEW nn`, le HP-41 affiche le contenu du registre nn à moins que `CLD` n'efface l'affichage ou qu'un autre message ne soit affiché. Les indicateurs binaires 21 et 55 affectent le fonctionnement de `ANVIEW` et de `VIEW`.

Utilisation de `PSE`

`PSE` peut servir à l'affichage bref d'un message; l'exécution du programme est suspendue pendant un peu moins d'une seconde.

- Si l'afficheur contient déjà un message (autre que le témoin d'exécution de programme), ce message reste à l'affichage.
- Si l'afficheur ne contient pas de message et que le clavier alphabétique est actif, le HP-41 affiche le contenu du registre ALPHA.
- Autrement, le HP-41 affiche le contenu du registre X.

Utilisation de `tone` et `BEEP`

L'exécution de `tone n` émet une tonalité simple de fréquence définie par n , compris entre 0 (son le plus grave) et 9 (son le plus aigu).

L'exécution de `BEEP` produit une séquence fixe de 4 tonalités.

Programmes d'enregistrement de temps

Table des matières

Introduction	320
Exemples de programmes	321
Utilisation des programmes	321
Chargement des programmes	321
Création des fichiers	321
Utilisation de TR	323
Utilisation de SIGMA	325
Fichiers servant à l'enregistrement des temps	328
Fichier de texte TRECS	329
Fichier de données LAST	330
Explication de TR	330
Registres et indicateurs binaires employés par TR	331
Listage du programme TR	332
Explication de SIGMA	339
Registres et indicateurs employés par SIGMA	339
Listage du programme SIGMA	340

Introduction

Ce chapitre comprend des programmes servant à enregistrer des temps. Sous le titre « Utilisation des programmes », vous trouverez des instructions relatives à chaque programme. Une fois que vous aurez compris l'objet de ces programmes, vous pourrez étudier leurs listages commentés afin de comprendre leur fonctionnement.

Même si vous n'avez pas à écrire des programmes aussi performants, certaines des techniques qu'ils utilisent peuvent vous servir dans vos propres applications. Chaque programme est divisé en parties correspondant à ses fonctions principales, et chaque partie est divisée en sous-parties effectuant une tâche élémentaire.

Exemples de programmes

Les chapitres 19, 20 et 21 se réfèrent à ces programmes pour illustrer certaines techniques et fonctions. Les sujets présentés dans ces programmes comprennent :

- Indicateurs binaires de saisie de données et de non prise en compte d'erreur (chapitre 19, indicateurs binaires de commande).
- Recherches de labels locaux (chapitre 20).
- Boucles utilisant des fonctions conditionnelles (chapitre 20).
- Stockage efficace de données (chapitre 21).
- Interaction programme-utilisateur (chapitre 21, `PROMPT`).

Utilisation des programmes

La mémorisation de votre temps implique quatre processus :

- Chargement des programmes TR (*time record* ; enregistrement du temps) et SIGMA (résumé).
- Création des fichiers devant garder vos informations (opération unique, sauf si vous éliminez ces fichiers ou réinitialisez le HP-41).
- Mise à jour des informations, nécessaire au début ou à la fin de toute tâche.
- Présentation résumée de l'information.

Vous devez suivre pour chaque processus les étapes suivantes.

Chargement des programmes

Ce chapitre contient un listage des programmes TR et SIGMA. Pour les charger, suivez les instructions données au chapitre 18.

L'annexe J en contient une version en code à barres. Pour les charger au moyen du crayon-lecteur de codes à barres HP 82153, suivez les instructions données dans son manuel.

Création des fichiers

Création de fichier LAST. Ce fichier de données contient trois valeurs qui décrivent le dernier état : « *last job* », nom de la dernière tâche ; « *last time* », heure de la journée ; « *last date* », date. Vous devez créer ce fichier et donner des valeurs initiales pour l'heure et la date.

1. Entrez `LAST` dans le registre ALPHA.
2. Entrez `3` dans le registre X.
3. Exécutez `CRFLD`, afin de créer un fichier de données nommé **LAST**, comportant trois registres. Le pointeur est placé sur le premier registre.

4. Entrez 1 dans le registre X.
5. Exécutez **[STO] 02**. La valeur initiale de « *last time* » est 1.
6. Exécutez **[DATE]**, afin de placer la date courante dans le registre X (le contenu du registre X diffère de l'affichage).
7. Exécutez **[STO] 03**. La valeur initiale de « *last date* » est la date courante.
8. Entrez 1.003 dans le registre X.
9. Exécutez **[SAVERX]**: le contenu des registres R_{01} , R_{02} et R_{03} est recopié dans LAST. (La valeur initiale de « *last job* » est le contenu courant de R_{01} . Vous la remplacerez par un nom réel à la première exécution de TR).

Création du fichier TRECS (*time records* : enregistrements de temps). Ce fichier de texte doit contenir deux enregistrements par tâche concernée. Il vous suffit de le créer ; le programme de mise à jour TR y stockera les enregistrements voulus.

1. Entrez **TRECS** dans le registre ALPHA.
2. Calculez $j(w + 1)$, où j est le nombre de tâches et w le nombre de semaines pour lesquelles vous voulez effectuer des enregistrements. Ceci constitue une estimation du nombre de registres dont TRECS aura besoin ; l'estimation peut être grossière puisque TR ajoute automatiquement des registres à TRECS si besoin est. Tapez cette estimation dans le registre X.
3. Exécutez **[CRFLAS]**, afin de créer le fichier de texte TRECS contenant le nombre estimé de registres.

Création des premiers enregistrements. Exécutez TR une fois pour créer deux enregistrements dans TRECS pour votre première tâche (nom et heure). A moins que vous ne commenciez réellement à travailler à cette tâche, vous pouvez arrêter en répétant TR. Aucune durée ne sera enregistrée si vous arrêtez moins de trois minutes après votre début.

1. Exécutez TR.
2. En réponse au message **JOB NAME = ?** (nom de la tâche = ?), tapez le nom de votre première tâche et appuyez sur **[RS]**. (Il est inutile d'appuyer sur **[ALPHA]** avant ou après la frappe du nom de la tâche).
3. Quand le HP-41 affiche **START job ?** (début de tâche ?), appuyez sur **[RS]** pour confirmer la frappe du nom précédent.
4. Quand le HP-41 affiche **NEW JOB ?** (nouvelle tâche ?), appuyez sur **[RS]** pour confirmer la frappe du nom précédent. Le HP-41 crée deux nouveaux enregistrements pour votre tâche.
5. Quand le HP-41 affiche **START NOW ?** (début maintenant ?), appuyez sur **[RS]** pour confirmer.

La première exécution avec succès de TR est indiquée par un affichage de zéros. Vous pouvez indiquer que vous vous arrêtez maintenant en exécutant à nouveau TR.

6. Appuyez sur **[RS]** pour réexécuter TR. (Si vous effectuez une tâche à la suite d'une autre, vous exécutez TR deux fois de suite).
7. Quand le HP-41 affiche **STOP NOW ?** (arrêt maintenant ?), appuyez sur **[RS]** pour confirmer.

Quand des zéros s'affichent, vous pouvez utiliser TR comme indiqué ci-dessous.

Utilisation de TR

Exécutez le programme TR chaque fois que vous commencez ou arrêtez une tâche. Le programme propose différentes options, décrites plus loin.

Instructions générales

- Trois variables apparaissent dans les instructions de début et d'arrêt :

Last job est le nom de la dernière tâche avec laquelle vous avez exécuté TR. Il ne change que si vous en tapez une autre.

Last time est la dernière heure (début ou arrêt) utilisée dans TR. Elle change à chaque exécution de TR (l'heure de début est mémorisée avec un signe moins ; l'arrêt est codé 1. A l'exécution suivante de TR, le signe indique si la précédente exécution était destinée à un début, ce qui implique une fin pour celle-ci, ou vice-versa).

Last date est la dernière date d'exécution de TR ; elle est traitée automatiquement.

- Une question est généralement suivie d'une proposition de réponse. Si vous voulez la garder, appuyez sur **[RS]**. Sinon, modifiez-la et appuyez sur **[RS]**.

- Pour l'heure, vous avez le choix entre le format 24 heures :

$$HH.MM = HH \text{ heures } (0 \leq HH < 24) \text{ et } MM \text{ minutes } (0 \leq MM < 60)$$

et le format 12 heures, où toute heure d'après-midi est affectée du signe moins :

$$-HH.MM = 12 + HH \text{ heures } (0 \leq HH < 12) \text{ et } MM \text{ minutes } (0 \leq MM < 60).$$

- A la question **JOB NAME ?**, répondez en tapant le nom d'une tâche et appuyez sur **[RS]** (sans appuyer sur **[ALPHA]**). Le système considère que vous commencez à travailler à cette tâche. Le cas n'échoit qu'à la première exécution de TR ou si vous avez effacé deux enregistrements de « *last job* » dans TRECS.
- Si le message **NEED ROOM** (besoin de place) apparaît, réduisez la taille d'un fichier en mémoire étendue autre que LASR ou TRECS de 2 registres au moins, et appuyez sur **[RS]**. Le cas se présente quand TR ne peut pas agrandir TRECS par manque de registres disponibles en mémoire étendue.

Début de tâche.

1. Exécutez TR.
2. A la question **START last job ?**, vous pouvez :
 - Soit appuyer sur **[R/S]** pour confirmer que vous travaillez à la même tâche, et le programme passe à l'étape 4.
 - Soit taper un autre nom et appuyer sur **[R/S]** (inutile d'appuyer sur **[ALPHA]**). Si TR a déjà enregistré ce nom, il passe à l'étape 4.
3. S'il s'agit d'un nouveau nom, TR affiche **NEW JOB ?**. Vous pouvez :
 - Soit confirmer le nouveau nom en appuyant sur **[R/S]**, auquel cas TR crée 2 nouveaux enregistrements pour lui dans TRECS.
 - Soit taper un nom déjà existant et appuyer sur **[R/S]** (inutile d'appuyer sur **[ALPHA]**). Ceci permet de corriger une faute de frappe à l'étape 2.
4. A la question **START NOW ?** (commencer maintenant ?), vous pouvez :
 - Soit appuyer sur **[R/S]** pour confirmer que vous commencez maintenant.
 - Soit taper dans le registre X une autre heure et appuyer sur **[R/S]**.

L'affichage de zéro signale la fin de l'exécution de TR.

Fin de tâche.

1. Exécutez TR.
2. a. Si « *last date* » coïncide avec la date courante, le HP-41 affiche **STOP NOW ?**. Vous pouvez :
 - Soit appuyer sur **[R/S]** pour confirmer que vous arrêtez de travailler.
 - Soit taper dans le registre X une autre heure d'arrêt et appuyer sur **[R/S]**.
- b. Si « *last date* » diffère de la date courante, le HP-41 affiche **PAST 24:00 ?**. Vous pouvez :
 - Soit appuyer sur **[R/S]** pour confirmer que vous avez travaillé après minuit. TR actualise l'enregistrement de temps pour le jour où vous avez commencé (de « *last time* » à minuit) TR actualise aussi « *last time* » et « *last date* » pour indiquer que vous avez commencé à 00:00 heure du jour suivant, puis retourne à l'étape 2a ou 2b.
 - Soit taper dans le registre X une autre heure d'arrêt et appuyer sur **[R/S]**. (Le cas se présente lorsque vous arrêtez de travailler le même jour que vous avez commencé et avez oublié d'exécuter TR).
3. Si vous tapez une heure d'arrêt antérieure à l'heure de début, le HP-41 affiche **START > STOP** et retourne à l'étape 2a ou 2b.

L'affichage de zéro signale la fin de l'exécution de TR. Si vous commencez à travailler à une tâche différente, appuyez sur **[R/S]** pour exécuter TR à nouveau.

Utilisation de SIGMA

Le programme SIGMA possède deux fonctions : récapitulatif et effacement. Il demande d'abord les dates de la période dont vous voulez le récapitulatif, puis fournit le récapitulatif. Il offre ensuite la possibilité d'effacer les enregistrements de toutes les tâches antérieures à une date déterminée ou bien une tâche entière.

Vous pouvez spécifier toute période comprenant des enregistrements, y compris des parties d'une période entière. Par exemple, supposons que vous vouliez rassembler les enregistrements d'un mois. Vous pouvez pendant ce mois récapituler les enregistrements d'une semaine le lundi suivant, et au début du mois suivant récapituler le mois précédent, en effacer les enregistrements, et recommencer le cycle.

Instructions générales.

- Les instructions de récapitulation et d'effacement utilisent deux variables.
 - Earliest date* est la date du plus ancien enregistrement à traiter (récapituler ou effacer). Par défaut, SIGMA prend la date du premier enregistrement. En phase de récapitulatif, vous pouvez spécifier une date ultérieure (mais non postérieure à « *most recent date* »).
 - Most recent date* est la date du plus récent enregistrement à traiter (récapituler ou effacer). Par défaut, SIGMA prend la date du dernier enregistrement. En phase de récapitulation, vous pouvez spécifier une date antérieure (mais non antérieure à « *earliest date* »).
- Quand le programme affiche une question relative à une date, la date affichée est la valeur courante de « *earliest date* » ou de « *most recent date* ». Pour confirmer la date affichée, appuyez sur **[RS]**. Sinon, modifiez-la et appuyez sur **[RS]**.
- Si le programme affiche **ILLEGAL DATE** (date illicite), c'est que la date que vous avez tapée n'est pas valide en elle-même (vérifiez le format courant), ou est antérieure à la date la plus ancienne ou postérieure à la date la plus récente.

Récapitulatif des temps enregistrés.

1. Si vous utilisez une imprimante HP 82143A ou HP 82162A, positionnez son commutateur sur **MAN** (manuel).
2. Exécutez **SIGMA** (touches **[F]** du clavier alphabétique).
3. Lorsque le programme affiche **FROM earliest date?** (DEPUIS?), vous pouvez :
 - Soit appuyer sur **[RS]** pour confirmer la date.
 - Soit taper dans **X** une date ultérieure et appuyer sur **[RS]**. Le programme répète cette étape pour confirmation.
4. Lorsque le programme affiche **THROUGH most recent?** (JUSQU'A?), vous pouvez :
 - Soit appuyer sur **[RS]** pour confirmer la date.
 - Soit taper dans **X** une date antérieure et appuyer sur **[RS]**. Le programme répète cette étape pour confirmation.

5. Si le programme affiche **PACKING** (compactage ; éventuellement très brièvement) et **TRY AGAIN** (essayez à nouveau), appuyez sur **[RS]**. Le cas se présente lorsque le programme tente d'affecter au stockage de données plus de registres qu'il n'y en a de disponibles. Si le compactage automatique libère assez de registres, le programme peut s'exécuter normalement dès que vous appuyez sur **[RS]** ; s'il manque toujours de registres, il affiche **TRY AGAIN**, et vous pouvez :
- Soit exécuter **SIGMA** à nouveau (à partir de l'étape 1) en spécifiant une période plus courte pour le récapitulatif.
 - Soit effacer un programme, des alarmes, ou des affectations de touches du clavier utilisateur ; exécuter **[PACK]** ou **[CTD]** ; puis exécuter à nouveau **SIGMA** (depuis l'étape 1) en spécifiant la même période qu'auparavant.
6. Le programme fournit un récapitulatif des enregistrements de temps pour la période que vous avez spécifiée. L'exemple suivant présente un récapitulatif imprimé. En l'absence d'imprimante, le programme affiche chaque ligne pendant environ une seconde jusqu'à ce que vous appuyez sur **[RS]**, ce qui vous permet de noter les résultats.

```

                                TIME RECORDS
                                from
Date la plus ancienne → 07/01/83
                                through
Date la plus récente → 07/15/83

JOB 1
  07/01  8.2
  07/02  0.0
  07/03  0.0
  07/04  5.5
  07/05  6.5
  07/06  6.8
  07/07  0.0
  07/08  0.0
  07/09  0.0
  07/10  0.0
  07/11  6.3
  07/12  3.6
  07/13  2.0
  07/14  4.5
  07/15  0.7
TOTAL  44.1 ← Horaire total de JOB 1.

```

Horaires quotidiens de JOB 1.

Lundis

		JOB 2	
		07/01	0,0
		07/02	0,0
		07/03	0,0
Lundis	}	07/04	3,1
		07/05	1,8
		07/06	1,4
		07/07	8,2
		07/08	8,3
		07/09	0,0
		07/10	0,0
		07/11	1,7
		07/12	5,3
		07/13	6,1
		07/14	3,6
		07/15	7,8
		TOTAL	47,3

← Horaire total de JOB 2.

		SUMMARY	
		07/01	8,2
		07/02	0,0
		07/03	0,0
Lundis	}	07/04	8,6
		07/05	8,3
		07/06	8,2
		07/07	8,2
		07/08	8,3
		07/09	0,0
		07/10	0,0
		07/11	8,0
		07/12	8,9
		07/13	8,1
		07/14	8,1
		07/15	8,5
		TOTAL	91,4

← Horaire total de JOB 1 et JOB 2.

Effacement des enregistrements.

7. Une fois terminée la partie récapitulative, le programme affiche **CLEAR D, J ? E**, proposant d'effacer au choix :
- [D]** = days ; jours. Quand le programme affiche **THROUGH most recent date ? (JUSQU'A)**, vous pouvez :
- Appuyer sur **[RS]** pour confirmer la date affichée ; le programme efface tous les enregistrements dont la date est antérieure ou égale à la date spécifiée et repose la question **CLEAR D, J ? E**.
 - Taper dans **X** une date antérieure et appuyer sur **[RS]**. Le programme vous demande de confirmer cette date.
- [J]** = job ; tâche. Quand le programme affiche **NAME OF JOB ?** (nom de tâche ?), tapez le nom de la tâche à effacer et appuyez sur **[RS]** (inutile d'appuyer sur **[ALPHA]**).
- Si le programme trouve une tâche répondant au nom que vous avez tapé, il efface le nom et le temps associé et repose la question **CLEAR D, J ? E**.
 - Si le programme ne trouve pas de tâche répondant au nom que vous avez tapé, il affiche **NO SUCH NAME** (nom inconnu) et repose la question **CLEAR D, J ? E**.
- [E]** = exit ; fin. Le programme remet l'indicateur binaire à la valeur antérieure (avec les options qu'il représente) et achève son exécution.
8. Appuyez sur **[GTU] [] []** ou exécutez un autre programme. Ceci est nécessaire car **[D]**, **[J]** et **[E]** agissent comme indiqué ci-dessus tant que **SIGMA** est le programme en cours et que le clavier personnel est activé.

Fichiers servant à l'enregistrement des temps

Les enregistrements de temps sont stockés dans deux fichiers de mémoire étendue ; un fichier de texte nommé **TRECS**, contenant deux enregistrements par tâche, et un fichier de données nommé **LAST**, contenant des informations relatives à la dernière fois où vous avez commencé ou arrêté une tâche.

Fichier de texte TRECS

TRECS rassemble le nom et l'enregistrement de temps en une suite de paires d'enregistrements :

Numéro d'enregistrement	Contenu de l'enregistrement
000	Nom de la tâche 1.
001	Horaire de la tâche 1.
002	Nom de la tâche 2.
003	Horaire de la tâche 2.
⋮	⋮
n	Nom de la tâche $n/2 + 1$.
$n + 1$	Horaire de la tâche $n/2 + 1$.
⋮	⋮

Chaque caractère d'un enregistrement de temps est un multiplet dont la valeur décimale représente le nombre d'heures passées sur une tâche un jour déterminé. Le premier multiplet représente l'horaire du jour courant, et les multiplats suivants les horaires des jours précédents. Cette disposition autorise des enregistrements de longueurs différentes ; le n ième multiplet de chaque enregistrement représente le même jour pour toutes les tâches. Les horaires d'une tâche ancienne, pour laquelle existent des enregistrements sur une longue période, sont plus nombreux que ceux d'une tâche plus récente. A la création d'une nouvelle tâche, deux enregistrements sont ajoutés à la fin de TRECS. La première tâche de TRECS est la plus ancienne tâche, et ses enregistrements d'horaires sont plus (ou aussi) longs que ceux des autres.

L'horaire quotidien est calculé au dixième d'heure près. Cette valeur, multipliée par 10, est la valeur décimale du multiplet qui la code. Le tableau suivant donne quelques exemples.

Horaire (heures/minutes)	Horaire (heures décimales)	Multiplet (valeur décimale)
3 h 20 mn	3.3	33
5 h 30 mn	5.5	55
7 h 00 mn	7.0	70
10 h 42 mn	10.7	107
13 h 15 mn	13.3	133

Deux multiplats font exception :

- Le multiplet nul pouvant disparaître du registre ALPHA, un horaire nul est codé par un multiplet de valeur décimale 254.
- Le dernier multiplet de chaque enregistrement a une valeur décimale de 255.

Fichier de données LAST

Un second fichier (de type données, appelé LAST) stocke des informations sur le dernier moment où vous avez débuté ou arrêté une tâche.

Numéro de registre	Contenu du registre
000	<i>Last job</i> = Nom de la dernière tâche
001	<i>Last time</i> = Dernière heure
002	<i>Last date</i> = Dernière date

Last job. Devant tenir dans un registre, les noms de tâches sont limités à 6 caractères. Si vous commencez et arrêtez plusieurs fois la même tâche, vous n'avez pas à spécifier son nom à chaque fois, puisqu'il est mémorisé dans LAST. Le nom d'une tâche ne peut être composé des premières lettres du nom d'une autre; par exemple, le nom ABCD interdit d'utiliser ABC; mais BCD peut exister.

Last time. Le programme de mise à jour TR affecte du signe moins l'heure de début de tâche, et du signe plus l'heure de fin de tâche, à destination de sa prochaine exécution. Si vous n'oubliez pas d'exécuter TR à chaque début et fin de tâche, vous n'avez pas à taper la date.

Last date. Quand la date stockée dans LAST diffère de la date courante, l'horaire nul est automatiquement ajouté à l'horaire de chaque tâche, afin de garder les enregistrements de temps bien rangés par date.

Le programme TR recopie le contenu des trois registres de LAST dans R₀₁ à R₀₃ (mémoire principale) à chaque début ou arrêt de tâche. TR utilise ces valeurs, les actualise, puis les recopie dans LAST.

Explication de TR

Le programme TR contient plusieurs parties principales :

- Les sous-programmes principaux rappellent les contenus de LAST en mémoire étendue, évaluent ce qui doit être fait, exécutent les sous-programmes appropriés, et recopient les valeurs actualisées dans LAST.
- Les sous-programmes d'arrêt de tâche demandent quand vous avez arrêté, calculent votre horaire de travail, actualisent l'enregistrement approprié dans TRECS, et actualisent *last time*.

- Les sous-programmes relatifs à une nouvelle date ajoutent un multiplet représentant un temps nul à chaque enregistrement de temps, et actualisent *last date*.
- Les sous-programmes de début de tâche demandent sur quelle tâche vous commencez à travailler, quand, et actualisent *last job* et *last time*.

En outre, trois sous-programmes de moindre importance, situées à la fin de TR, effectuent la recherche dans TRECS d'un nom de tâche particulier, la conversion d'un temps, et l'agrandissement du fichier TRECS.

Registres et indicateurs binaires employés par TR

Le programme TR utilise les registres de stockage de données et les indicateurs binaires suivants. Veuillez vous reporter à ces tableaux lors de l'étude du listage de TR.

Registres et leur contenu pour TR

Numéro de registre	Contenu du registre
00	Etat initial de l'indicateur binaire
01	<i>Last job</i>
02	<i>Last time</i>
03	<i>Last date</i>
04	Valeur décimale de l'horaire de travail

Conditions représentées par les indicateurs binaires pour TR

Numéro de registre	Condition représentée par l'indicateur binaire
01	0 = Nom de tâche trouvé dans TRECS
	1 = Nom de tâche non trouvé dans TRECS
02	0 = Arrêt de tâche maintenant
	1 = Début de tâche maintenant
03	0 = Même date (<i>last date</i> = date courante)
	1 = Nouvelle date (<i>last date</i> < date courante)

Listage du programme TR

Sous-programmes principaux. Ils rappellent des données datant de la dernière exécution de TR, déterminent ce qu'il y a à faire, exécutent les sous-programmes appropriés, stockent les données actualisées, et terminent l'exécution de TR.

01 ♦ LBL "TR"	Les lignes 01 à 26 rappellent <i>last job</i> , <i>last time</i> et <i>last date</i> . Les lignes 02 à 05
02 SIZE?	s'assurent que 5 registres au moins sont affectés au stockage des données. Les
03 5	lignes 06 à 09 sauvegardent l'état de l'indicateur binaire avant de choisir le format
04 X>Y?	d'affichage et de mettre à zéro l'i.b. d'autorisation d'impression. Les lignes 10 à
05 PSIZE	14 recopient <i>last job</i> dans R ₀₁ , <i>last time</i> R ₀₂ , et <i>last date</i> dans R ₀₃ , valeurs stockées
06 RCLFLAG	dans LAST lors de la dernière exécution de TR. Les lignes 15 et 16 choisissent
07 STO 00	TRECS comme fichier courant de mémoire étendue. La ligne 17 appelle un
08 FIX 1	sous-programme à la ligne 194 qui vérifie si TRECS contient toujours la tâche
09 CF 21	nommée <i>last job</i> ; si oui, les lignes 18 et 19 provoquent le branchement à la ligne
10 "LAST"	27 . Si vous avez effacé <i>last job</i> de TRECS depuis la dernière exécution de TR, les
11 0	lignes 20 à 23 demandent un nom valide. Les lignes 21 et 23 valident et invalident
12 SEEKPTA	automatiquement le clavier alphanumérique. La ligne 24 change <i>last name</i> pour le
13 1.003	nom que vous venez de taper, et les lignes 25 et 26 indiquent dans <i>last time</i> que
14 GETRX	vous avez arrêté la dernière fois, et donc commencez cette fois-ci.
15 "TRECS"	
16 RCLPTA	
17 XEQ 50	
18 FC? 01	
19 GTO 20	
20 "JOB NAME = ?"	
21 AON	
22 PROMPT	
23 AOFF	
24 ASTO 01	
25 1	
26 STO 02	

27 ♦ LBL 20
 28 CF 02
 29 RCL 02
 30 X>0?
 31 SF 02
 32 CF 03
 33 RCL 03
 34 DATE
 35 X≠Y?
 36 SF 03
 37 FC? 02
 38 XEQ 01
 39 FS? 03
 40 GTO 30
 41 FS? 02
 42 XEQ 40
 43 "LAST"
 44 0
 45 SEEKPTA
 46 1.003
 47 SAVERX
 48 RCL 00
 49 STOFAG
 50 CLX
 51 STOP
 52 GTO "TR"

Les lignes 27 à 52 gèrent l'exécution du reste de TR. Les lignes 28 à 31 désarmer l'i.b. 02 si vous arrêtez la tâche, ou à 1 si vous la commencez (Si, la dernière fois, vous avez commencé la tâche, TR a attribué à *last time* une valeur négative; si vous avez arrêté, *last time* vaut 1). Les lignes 32 à 36 désarment l'i.b. 03 si *last date* est la date courante, si 1 si elles diffèrent. Les lignes 37 à 42 gèrent l'exécution générale du programme. Si vous arrêtez une tâche, les lignes 37 et 38 appellent le sous-programme qui commence à la ligne 58. Si un ou plusieurs jours ont passé depuis la dernière exécution de TR, les lignes 39 et 40 provoquent le branchement à un sous-programme qui commence à la ligne 106 (addition d'un multiplé à chaque enregistrement de temps et branchement à la ligne 27). Si vous commencez une tâche, les lignes 41 et 42 appellent le sous-programme qui commence à la ligne 135. Les lignes 43 à 51 terminent l'exécution de TR. Les lignes 43 à 47 recopient les valeurs courantes de *last job*, *last time* et *last date* dans le fichier LAST, qui les sauvegarde jusqu'à la prochaine exécution de TR. Les lignes 48 et 49 restituent l'état précédent de l'i.b. (et les options qu'il représente), et la ligne 50 efface du registre X l'état de l'i.b. La ligne 51 arrête l'exécution de TR, et la ligne 52 branche à la ligne 01, ce qui permet d'exécuter à nouveau TR en appuyant simplement sur **[RS]**.

Arrêt de tâche. Ces sous-programmes actualisent l'enregistrement de temps de *last job* lors de l'arrêt d'une tâche. Normalement, seuls les second et troisième sous-programmes sont exécutés; le premier ne l'est qu'en cas d'erreur, et le quatrième concerne l'arrêt de tâche un jour différent du jour de début.

53 ♦ LBL 00
 54 "START > STOP"
 55 TONE 5
 56 AVIEW
 57 PSE
 58 ♦ LBL 01
 59 FS? 03
 60 GTO 03
 61 TIME
 62 "STOP NOW?"
 63 PROMPT

Les lignes 53 à 57 affichent un message d'erreur. Si vous tapez une date d'arrêt antérieure à la date de début, la ligne 72 provoque le branchement sur l'affichage du message, avant reprise.

Les lignes 58 à 63 déterminent l'heure d'arrêt. Si vous arrêtez un jour différent de celui où vous avez commencé, les lignes 59 et 60 effectuent un branchement à la ligne 96. Les lignes 61 à 63 demandent si vous arrêtez à l'instant, puis stoppent l'exécution; si vous tapez une autre heure, celle-ci remplace l'heure courante dans X.

64 • LBL 02
65 X<0?
66 XEQ 60
67 X=0?
68 24
69 RCL 02
70 HMS+
71 X<0?
72 GTO 00
73 HR
74 RND
75 10
76 *
77 STO 04
78 XEQ 50
79 GETREC
80 ATOX
81 RCL 04
82 +
83 254
84 X≠Y?
85 MOD
86 CLA
87 XTOA
88 RCLPT
89 INT
90 SEEKPT
91 1
92 DELCHR
93 INSCHR
94 STO 02
95 RTN

Les lignes **64 à 95** actualisent le multiplet de *last date* dans l'enregistrement de temps de *last job*. Si vous avez codé une heure d'après-midi par un signe moins, les lignes **65 et 66** appellent un sous-programme ligne 216 qui convertit l'heure au format 24 heures. Si l'heure d'arrêt est minuit, les lignes **67 et 68** placent 24 dans X. Les lignes **69 et 70** retirent l'heure de début de l'heure de fin: la ligne **69** rappelle *last time*, qui est dans ce cas votre heure (négative) de début; la ligne **70** ajoute votre heure d'arrêt à votre heure (négative) de début, ce qui donne le nombre d'heures. Si le résultat est négatif, les lignes **71 et 72** provoquent le branchement à un sous-programme d'erreur, ligne 53. Les lignes **73 à 77** stockent le nombre d'heures travaillées sous forme de multiplet: les lignes **73 et 74** convertissent le nombre du format heures-minutes-secondes au format décimal (avec un chiffre après la virgule); les lignes **75 et 76** convertissent ce nombre décimal en multiplet, que la ligne **77** stocke dans R₀₄. La ligne **78** appelle un sous-programme à la ligne 194 qui localise l'enregistrement de temps de *last job*; les lignes **79 et 80** placent dans X la valeur décimale du premier multiplet, qui représente l'horaire précédemment passé le jour *last date*. Les lignes **81 à 85** ajoutent à ce multiplet la valeur des heures travaillées à l'instant; les lignes **83 à 85** calculent le résultat dans le cas où la valeur du multiplet précédent est 254 (temps nul). Les lignes **86 et 87** placent dans le registre ALPHA le multiplet se trouvant dans X. Les lignes **91 à 93** actualisent le multiplet. La ligne **94** stocke 1 dans *last time*. La ligne **95** fait reprendre l'exécution à la ligne 39 si le processus d'actualisation est terminé; si vous avez travaillé après minuit, l'exécution reprend à la ligne 104.

96 • LBL 03
97 24
98 "PAST 24.00?"
99 CF 22
100 PROMPT
101 FS? 22
102 GTO 02
103 XEQ 02
104 0
105 STO 02

Les lignes **96 à 105** demandent si vous avez travaillé après minuit. La ligne **60** branche à ce sous-programme si vous arrêtez le travail un autre jour que celui où vous l'avez commencé. Les lignes **97 à 100** demandent si vous travaillez après minuit; si vous tapez une autre heure, elles placent 24 dans X, et les lignes **101 et 102** branchent à la ligne 64. La ligne **103** appelle un sous-programme à la ligne 64 qui actualise le multiplet de *last date* de l'enregistrement de temps de *last job*, de l'heure de début jusqu'à minuit; les lignes **104 et 105** donnent à *last time* la valeur zéro pour indiquer que le début a lieu à minuit.

Ajout de multiplets pour une nouvelle tâche. Lors de votre première exécution quotidienne de TR, le programme exécute ces sous-programmes qui ajoutent un multiplet au début de l'enregistrement de temps de chaque tâche, afin de maintenir une correspondance entre la position d'un multiplet dans l'enregistrement de temps et la date qu'il représente.

106 ♦ LBL 30
 107 ASROOM
 108 2
 109 *
 110 SF 25
 111 SEEKPT
 112 FC?C 25
 113 GTO 04
 114 XEQ 70
 115 GTO 30

Les lignes 106 à 115 s'assurent que TRECS dispose d'assez de multiplets pour en ajouter un à chaque enregistrement de temps. La ligne 107 retourne n , nombre de multiplets disponibles. Supposons que ces n multiplets sont ajoutés aux enregistrements de temps des n premières tâches ; la tâche $n + 1$ est la première à ne pas recevoir de multiplet. Les lignes 108 et 109 retournent $2n$, qui est la valeur du pointeur pour le nom de la tâche $n + 1$. Les lignes 110 et 111 vérifient s'il y a réellement $n + 1$ noms de tâches ; sinon, il y a assez de multiplet disponibles et les lignes 112 et 113 à la ligne 116. La ligne 114 appelle un sous-programme à la ligne 221, qui agrandit TRECS, et la ligne 115 effectue un branchement à la ligne 106.

116 ♦ LBL 04
 117 CLA
 118 254
 119 XTOA
 120 1
 121 SEEKPT
 122 SF 25

Les lignes 116 à 122 ajoutent un multiplet à chaque enregistrement de temps. Les lignes 117 à 119 placent dans ALPHA un multiplet signifiant aucune heure. Les lignes 120 et 121 placent le pointeur au premier caractère du premier enregistrement de temps. La ligne 122 arme l'indicateur binaire de non prise en compte d'erreur ; le programme exécute la boucle qui suit jusqu'à ce qu'une erreur désarme cet indicateur binaire.

123 ♦ LBL 05
 124 INSCHR
 125 2
 126 +
 127 SEEKPT
 128 FS? 25
 129 GTO 05
 130 RCL 03
 131 1
 132 DATE+
 133 STO 03
 134 GTO 20

Les lignes 123 à 134 ajoutent un multiplet à chaque enregistrement de temps. La ligne 124 insère le multiplet correspondant à « aucune heure » à la position courante du pointeur. Les lignes 125 à 127 tentent de mettre le pointeur au premier caractère de l'enregistrement de temps suivant. Si cet enregistrement existe, les lignes 128 et 129 branchent à la ligne 123. Quand tous les enregistrements de temps ont reçu un multiplet, les lignes 130 à 133 ajoutent un jour à *last date* afin de tenir compte du nouveau multiplet ajouté à chaque enregistrement de temps. La ligne 134 provoque le branchement à la ligne 27. (La valeur actualisée de *last date* est comparée à la date courante, et le programme exécute de nouveau ces sous-programmes si nécessaire).

Début de tâche. Ces sous-programmes déterminent sur quelle tâche et à quelle heure vous commencez le travail. Si vous commencez sur la même tâche que celle sur laquelle vous avez travaillé la dernière fois, c'est le premier sous-programme qui s'exécute. Si vous tapez le nom d'une tâche qui existe déjà (que vous avez déjà utilisée avec TR) les second et troisième sous-programmes sont aussi exécutés. Si vous tapez le nom d'une nouvelle tâche, ce sont les quatrième, cinquième et sixième sous-programmes qui sont exécutés. Dans tous les cas, le dernier sous-programme est exécuté pour déterminer l'heure de début.

135 ♦ LBL 40

136 "START"

137 ARCL 01

138 "┌ ?"

139 CF 23

140 AON

141 PROMPT

142 AOFF

143 FS? 01

144 GTO 07

145 FC? 23

146 GTO 11

147 ♦ LBL 06

148 ASTO 01

149 ♦ LBL 07

150 XEQ 50

151 FC? 01

152 GTO 11

153 ♦ LBL 08

154 CF 23

155 "NEW JOB?"

156 AON

157 PROMPT

158 AOFF

159 FS? 23

160 GTO 06

161 ♦ LBL 09

162 CLA

163 ARCL 01

164 SF 25

165 APPREC

166 FS?C 25

167 GTO 10

168 XEQ 70

169 GTO 09

Les lignes 135 à 146 demandent si vous commencez sur la même tâche que celle sur laquelle vous avez travaillé la dernière fois. Les lignes 136 à 141 affichent **START last job ?** Les lignes 140 et 142 valident et invalident automatiquement le clavier alphabétique. Si vous avez demandé un nom de tâche licite dans le premier sous-programme parce que la valeur stockée pour *last job* n'était pas licite, les lignes 143 et 144 provoquent le branchement à la ligne 149 si vous avez simplement confirmé que vous commencez sur *last job*, les lignes 145 et 146 effectuent un branchement à la ligne 183. (Si vous avez tapé un autre nom de tâche, l'exécution continue avec les sous-programmes suivants).

Les lignes 147 et 148 prennent le nom de tâche que vous venez de taper pour valeur provisoire de *last job*.

Les lignes 149 à 152 vérifient l'autre nom de tâche. La ligne 150 appelle un sous-programme à la ligne 194 qui met à 0 l'indicateur binaire 01 si le nom de tâche existe dans TRECS. Si le nom existe, les lignes 151 et 152 provoquent le branchement à la ligne 183.

Les lignes 153 à 160 vérifient si vous avez tapé le nom d'une nouvelle tâche. Les lignes 154 à 157 affichent **NEW JOB ?** Les lignes 156 et 158 valident et invalident automatiquement le clavier alphabétique. Si vous tapez un nouveau nom de tâche (parce que le précédent était mal tapé), les lignes 159 et 160 effectuent un branchement à la ligne 147.

Les lignes 161 à 169 créent un enregistrement contenant le nom de la nouvelle tâche. Les lignes 162 et 163 placent ce nom dans le registre ALPHA. Les lignes 164 à 167 tentent d'annexer ce nom dans TRECS et si elles réussissent, provoquent un branchement à la ligne 170. S'il n'y a pas assez de place dans TRECS, la ligne 168 appelle un sous-programme à la ligne 221 qui agrandit TRECS puis la ligne 169 provoque un branchement à la ligne 161 afin de recommencer.

170 ♦ LBL 10
171 CLA
172 254
173 XTOA
174 1
175 +
176 XTOA
177 SF 25
178 APPREC
178 FS?C 25
180 GTO 11
181 XEQ 70
182 GTO 10

Les lignes **170 à 182** créent un enregistrement de temps pour la nouvelle tâche. Les lignes **171 à 176** placent un multiplet correspondant à aucune heure et un multiplet signifiant fin d'enregistrement de temps dans le registre ALPHA. Les lignes **177 à 180** tentent d'annexer ce multiplet dans TRECS et si elles réussissent provoquent le branchement à la ligne 183. S'il n'y a pas assez de place dans TRECS, la ligne **181** appelle un sous-programme à la ligne 221 pour agrandir TRECS, puis la ligne **182** effectue un branchement à la ligne 170 pour essayer de nouveau.

183 ♦ LBL 11
184 TIME
185 "START NOW?"
186 PROMPT
187 X<0?
188 XEQ 60
189 24
190 MOD
191 CHS
192 STO 02
193 RTN

Les lignes **183 à 193** demandent l'heure de début. Les lignes **184 à 186** affichent **START NOW ?** ; si vous tapez une autre heure, cette heure remplace dans X l'heure courante. Si vous tapez une heure négative pour indiquer « après-midi », les lignes **187 et 188** appellent un sous-programme à la ligne 216 qui convertit l'heure au format 24 heures. Si vous avez tapé 24 pour indiquer minuit, les lignes **189 et 190** changent cette valeur pour 0. Les lignes **191 et 192** donnent à *last time* la valeur de l'heure de départ ; elle est négative pour indiquer qu'il s'agit d'une heure de départ. La ligne **193** fait reprendre l'exécution à la ligne 43.

Recherche d'une tâche dans TRECS. Le programme appelle ce sous-programme pour deux raisons : les lignes **17 et 150** l'appellent pour déterminer si une tâche existe, et la ligne **78** l'appelle pour localiser une tâche dont on connaît l'existence.

195 ♦ LBL 50
195 CLA
196 ARCL 01
197 SF 01
198 0
199 SEEKPT

Les lignes **194 à 199** effectuent la recherche d'un nom de tâche dans TRECS. Les lignes **195 et 196** placent dans ALPHA le nom de tâche. La ligne **197** arme l'indicateur binaire 01 afin d'indiquer que le nom de tâche n'a pas été trouvé ; le sous-programme suivant désarme l'indicateur binaire 01 s'il trouve le nom. Les lignes **198 et 199** placent le pointeur au premier caractère du premier nom de tâche dans TRECS.

200 ♦ LBL 12	Les lignes 200 à 215 recherchent dans TRECS le nom de tâche. La ligne 201 recherche dans TRECS en commençant à la position courante du pointeur. Si le nom de tâche n'est pas trouvé, les lignes 202 et 203 font reprendre l'exécution à la ligne 18 ou 151. L'indicateur binaire 01 étant armé. Les lignes 204 à 208 placent le pointeur au premier caractère de l'enregistrement suivant, qui est généralement l'enregistrement de temps de la tâche. Cependant, des multipliets d'un enregistrement de temps peuvent correspondre au nom de la tâche. Afin de vérifier si la ligne 201 a réellement trouvé un nom de tâche, les lignes 209 à 212 vérifient si la valeur du pointeur retournée par la ligne 201 est le premier caractère d'un enregistrement de numéro pair. Sinon, la ligne 213 provoque le branchement à la ligne 199. Les lignes 214 et 215 désarment l'indicateur binaire 01 afin d'indiquer que le numéro de tâche existe, puis font reprendre l'exécution à la ligne 18, 79 ou 151.
201 POSFL	
202 X<0?	
203 RTN	
204 ENTER	
205 INT	
206 1	
207 +	
208 SEEKEPT	
209 X<>Y	
210 2	
211 MOD	
212 X=0?	
213 GTO 12	
214 CF 01	
215 RTN	

Conversion au format 24 heures

216 ♦ LBL 60	Les lignes 216 à 220 convertissent un nombre négatif (représentant une heure d'après-midi) au format 24 heures. Le programme appelle ce sous-programme des lignes 66 et 188.
217 ABS	
218 12	
219 +	
220 RTN	

Agrandissement de TRECS. Le programme appelle ce sous-programme quand il n'y a pas assez de place dans TRECS pour ajouter un multiplet à chaque enregistrement de temps ou pour annexer un nouvel enregistrement correspondant à une nouvelle tâche.

221 ♦ LBL 70	Les lignes 221 à 233 ajoutent un registre à TRECS. Ce sous-programme est appelé par les lignes 114, 168 et 181. Les lignes 222 à 225 calculent le nombre total de registres souhaité. Les lignes 226 à 229 tentent de changer la taille de TRECS au nombre total désiré et, si elles réussissent, font reprendre l'exécution à la ligne 115, 169 ou 182. Si la mémoire annexe n'a pas de registre disponible, les lignes 230 et 231 affichent un message d'erreur et arrêtent l'exécution. Après que vous ayez réduit la taille d'un autre fichier et appuyé sur [R/S] , la ligne 232 effectue le branchement à la ligne 221 pour essayer de nouveau.
222 "TRECS"	
223FLSIZE	
224 1	
225 +	
226 SF 25	
227 RESZFL	
228 FS?C 25	
229 RTN	
230 "NEED ROOM"	
231 PROMPT	
232 GTO 70	
233 END	

Explication du programme SIGMA

Le programme SIGMA détermine d'abord la période sur laquelle porte le récapitulatif; il produit celui-ci, puis vous permet d'effacer une tâche entière ou bien tous les enregistrements de temps d'une période déterminée.

Registres et indicateurs employés par SIGMA

Le programme SIGMA utilise les registres de stockage de données et les indicateurs binaires suivants. Veuillez vous reporter à ces tableaux lors de l'étude du listage de SIGMA.

Registres et leur contenu pour SIGMA

Numéro de registre	Contenu du registre	
R ₀₀	État précédent de l'i.b.	
R ₀₁	1.ccc, où <i>ccc</i> = valeur du pointeur de la date la plus récente.	
R ₀₂	Date la plus récente à récapituler ou effacer.	
R ₀₃	Date la plus ancienne à récapituler ou effacer.	
R ₀₄	{ Stockage temporaire de dates. <i>nn</i> = plus grande adresse du registre utilisé.	
R ₀₅	Nombre de commande de boucle.	
R ₀₆	Horaire total d'une tâche.	
R ₀₇	Horaire total d'un ensemble de tâches.	
R ₀₈	} Horaire quotidien de	la date la plus récente.
⋮		⋮
R _{<i>nn</i>}		la date la plus ancienne.

Chacun des nombres contenus dans les registres R₀₈ à R_{*nn*} est de la forme *xxx.yyy*. La partie entière est un multiplé représentant *xx.x* heures pour une tâche. La partie fractionnaire est un multiplé représentant *yy.y* heures pour un ensemble de tâches.

Conditions représentées par les indicateurs binaires pour SIGMA

Numéro de registre	Condition représentée par l'indicateur binaire
00	désarmé = demande de récapitulation. armé = demande d'effacement.
01	désarmé = la date tapée est licite. armé = la date tapée est illicite.

Listage du programme SIGMA

01 ♦ LBL "SIGMA" Les lignes **01 à 07** composent le sous-programme initial. Les lignes **02 et 03** stockent l'état courant de l'indicateur binaire dans le registre R_{00} . La ligne **04** invalide l'imprimante (si elle existe). Les lignes **05 et 06** placent dans R_{01} la valeur du pointeur de *first job/most recent date*. La ligne **07** indique que le programme est en cours de récapitulation.

02 RCLFLAG
03 STO 00
04 CF 21
05 1
06 STO 01
07 CF 00

Détermination des dates. Ces sous-programmes déterminent la période sur laquelle porte le récapitulatif. Si vous choisissez ensuite d'effacer des jours, le programme utilise à nouveau la plupart de ces sous-programmes pour déterminer la période à effacer. L'état de l'indicateur binaire 00 indique si le programme récapitule ou efface.

08 ♦ LBL 20 Les lignes **08 à 32** calculent les dates prises par défaut (quand le programme efface, la ligne 343 appelle cette partie du programme en tant que sous-programme). Les lignes **10 à 13** rappellent *last date*. Si le programme est en cours de récapitulation, les lignes **14 et 15** prennent par défaut *last date* comme valeur de *most recent date* dans R_{02} ; si le programme est en cours d'effacement, la date la plus récente qui est récapitulée reste dans R_{02} en tant que valeur prise par défaut. Dans tous les cas, *last date* reste dans la pile qui sert à calculer *earliest date*. Afin d'effectuer ce calcul, les lignes **16 à 27** localisent le multiplet « fin d'enregistrement de temps » (valeur décimale 255) dans le premier enregistrement de temps et utilise la valeur du pointeur de caractère pour calculer la différence en jours entre *last date* et *earliest date*. (A noter que le premier enregistrement de temps est le plus ancien). Les lignes **16 à 26** retournent le nombre de multipliants qui précèdent le multiplet de fin d'enregistrement. Quand la ligne **27** soustrait ce nombre de 1 (laissé dans la pile par la ligne 17), le résultat est négatif afin d'indiquer que l'on remonte dans le temps. La ligne **28** calcule ensuite *earliest date*. La ligne **29** prend cette date par défaut dans R_{03} , et la ligne **30** place cette date dans R_{04} , qui apparaît dans la première demande. Si le programme efface, les lignes **31 et 32** effectuent un branchement à la ligne 45; si le programme récapitule, l'exécution continue par le sous-programme suivant.

09 FIX 4
10 "LAST"
11 2
12 SEEKPTA
13 GETX
14 FC? 00
15 STO 02
16 "TRECS"
17 1
18 SEEKPTA
19 255
20 CLA
21 XTOA
22 CLX
23 POSFL
24 FRC
25 1 E3
26 *
27 -
28 DATE+
29 STO 03
30 STO 04
31 FS? 00
32 GTO 00

33 ♦ LBL 04
 34 "FROM"
 35 RCL 04
 36 XEQ 01
 37 FC? 22
 38 GTO 00
 39 XEQ 02
 40 FC? 01
 41 GTO 04
 42 RCL 03
 43 STO 04
 44 GTO 04

Les lignes **33 à 44** demandent la date de début de récapitulation. Les lignes **34 à 36** demandent si la date placée dans R_{04} est bien celle que vous voulez, et appellent le sous-programme à la ligne 62. Si vous acceptez la date de R_{04} , les lignes **37 et 38** provoquent le branchement à la ligne 45. Si vous tapez une autre date, la ligne **39** appelle le sous-programme à la ligne 71 afin de tester votre date. (Il place votre date dans R_{04} puis arme l'indicateur binaire 01 si votre date est licite, ou le désarme si elle est illicite). Si votre date est licite, les lignes **40 et 41** effectuent un branchement à la ligne 33 pour vous demander sa confirmation. Si votre date est illicite, les lignes **42 à 44** replacent dans R_{04} la valeur par défaut et provoquent un branchement à la ligne 33.

45 ♦ LBL 00
 46 RCL 04
 47 STO 03
 48 RCL 02
 49 STO 04

Les lignes **45 à 49** lient les sous-programmes de demande. Les lignes **46 et 47** remplacent la valeur prise par défaut par *earliest date* par celle que vous avez choisie. Les lignes **48 et 49** placent la valeur prise par défaut par *most recent date* dans R_{04} , laquelle apparaît dans la première question.

50 ♦ LBL 05
 51 "THROUGH"
 52 RCL 04
 53 XEQ 01
 54 FC? 22
 55 GTO 00
 56 XEQ 02
 57 FC? 01
 58 GTO 05
 59 RCL 02
 60 STO 04
 61 GTO 05

Les lignes **50 à 61** vous demandent la date la plus récente pour récapitulatif ou effacement. Les lignes **51 à 53** demandent si la date située dans R_{04} est celle que vous voulez et appellent le sous-programme à la ligne 62. Si vous acceptez la date de R_{04} , la ligne **54 et 55** effectuent un branchement à la ligne 93. Si vous tapez une autre date, la ligne **56** appelle le sous-programme à la ligne 71 pour tester celle-ci. Si votre date est licite, les lignes **57 et 58** provoquent le branchement à la ligne 50 et vous demandent de confirmer votre date. Si votre date est illicite les lignes **59 à 61** prennent la valeur par défaut située dans R_{04} et effectuent un branchement à ligne 50.

62 ♦ LBL01
 63 AVIEW
 64 PSE
 65 " "
 66 ADATE
 67 "┌?"
 68 CF 22
 69 PROMPT
 70 RTN

Les lignes **62 à 70** vous demandent des dates. Ce sous-programme est appelé par les lignes 36 et 53. Les lignes **63 et 64** affichent le contenu du registre ALPHA (**THROUGH** ou **FROM**). Les lignes **65 à 67** placent dans le registre ALPHA la date et un point d'interrogation. La ligne **68** met à 0 l'indicateur binaire de saisie numérique (afin que le programme puisse tester si vous avez tapé une autre date). La ligne **69** affiche le contenu du registre ALPHA et arrête l'exécution. La ligne **70** fait reprendre l'exécution à la ligne 37 ou 54.

71 ♦ LBL 02
72 CF 01
73 STO 04
74 RCL 02
75 SF 25
76 DDAYS
77 FC?C 25
78 GTO 03
79 X<0?
80 GTO 03
81 RCL 03
82 RCL 04
83 DDAYS
84 X<0?
85 GTO 03
86 RTN

Les lignes **71 à 86** testent votre date. Ce sous-programme est appelé par les lignes 39 et 56. La ligne **72** désarme l'indicateur binaire 01 (si votre date est illicite, le programme branche à la ligne 87 et met à 1 l'indicateur binaire 01). La ligne **73** place votre date dans R_{04} . Les lignes **74 à 76** calculent combien de jours séparent votre date de *last date*. Si vous avez tapé un nombre qui ne représente pas une date, la ligne **76** provoque une erreur et les lignes **77 et 78** un branchement à la ligne 87. Si votre date est postérieure à *last date*, les lignes **79 et 80** provoquent le branchement à la ligne 87. Les lignes **81 à 83** calculent combien de jours séparent votre date de *earliest date*. Si votre date est antérieure à *earliest date*, les lignes **84 et 85** effectuent le branchement à la ligne 87. Si votre date passe les trois tests avec succès, la ligne **86** fait reprendre l'exécution à la ligne 40 ou 57, l'indicateur binaire 01 étant désarmé pour indiquer que votre date est licite.

87 ♦ LBL 03
88 "ILLEGAL DATE"
89 AVIEW
90 TONE 5
91 SF 01
92 RTN

Les lignes **87 à 92** traitent les dates illicites. La ligne **88 à 90** affichent un message d'erreur et émettent une tonalité. Les lignes **91 et 92** font reprendre l'exécution à la ligne 40 ou 57 l'indicateur binaire 01 étant armé pour indiquer que votre date est illicite.

93 ♦ LBL 00
94 RCL 04
95 RCL 02
96 DDAYS
97 1
98 %
99 ST + 01
100 FS? 00
101 RTN
102 RCL 03
103 RCL 04
104 STO 02
105 DDAYS
106 8
107 +
108 STO 04
109 .1
110 %
111 6
112 +

Les lignes **93 à 112** terminent le choix de date. Les lignes **94 à 99** actualisent la valeur du pointeur dans R_{01} pour *first job/most recent date*, reflétant ainsi votre choix concernant la date la plus récente à récapituler ou effacer. Si le programme est en cours d'effacement, les lignes **100 et 101** font reprendre l'exécution à la ligne 344. Les lignes 103 et 104 remplacent dans R_{02} *last date* par votre choix de *most recent date*. Les lignes **102, 103 et 105** calculent la différence de jours qui séparent la plus ancienne et la plus récente date entrant dans le récapitulatif. A noter que R_{08} est le registre de plus petite adresse du bloc contenant les horaires quotidiens. Les lignes **106 à 108** placent *mn* dans R_{04} , où R_{mn} est le registre de plus grande adresse du bloc. Les lignes **109 à 112** calculent $6.mn$ qui représentent le bloc de registres qui contiendra les horaires quotidiens et totaux.

Préparation du récapitulatif. Ces sous-programmes effacent des registres et produisent le titre du récapitulatif.

113 ♦ LBL 06
 114 SF 25
 115 CLRGX
 116 FS?C 25
 117 GTO 01
 118 RCL 04
 119 1
 120 +
 121 PSIZE
 122 X<>Y
 123 GTO 06

Les lignes **113 à 123** effacent les registres des horaires quotidiens et totaux. Les lignes **114 et 115** tentent d'effacer ce bloc de registre. S'il y a suffisamment de registres affectés à un bloc entier, c'est-à-dire si *Rnnn* existe, les lignes **116 et 117** branchent à la ligne 124. Sinon, les lignes **118 à 121** affectent *nnn* + 1 registre, la ligne **122** renvoie *6.nnn* dans le registre X et la ligne **123** branche à la ligne 113.

124 ♦ LBL 01
 125 SF 12
 126 CF 13
 127 SF 21
 128 ADV
 129 ADV
 130 "TIME RECORDS"
 131 SF 25
 132 PRA
 133 FS1?C 25
 134 GTO 02
 135 CF 21
 136 AVIEW
 137 PSE

Les lignes **124 à 137** commentent le titre du récapitulatif. Les lignes **125 et 126** choisissent les modes d'impression double largeur et majuscule. La ligne **127** met un indicateur binaire d'autorisation d'impression. Les lignes **128 et 129** font avancer l'imprimante de deux lignes. La ligne **130** place le titre dans le registre ALPHA. La ligne **131** met à 1 l'indicateur binaire de non prise en compte d'erreur pour préparer la ligne **132**, [PRA] (*print ALPHA* ; impression alphabétique), qui est une instruction d'impression. Si une imprimante imprime le titre, les lignes **133 et 134** branchent à la ligne 138. Si aucune imprimante n'est présente ou sous tension, la ligne 135 désarme l'indicateur binaire d'autorisation d'impression (si bien que [AVIEW] n'arrête pas l'exécution) ; et les lignes **136 et 137** affichent le titre.

138 ♦ LBL 02 Les lignes **138 à 154** terminent le titre du récapitulatif. La ligne **139** choisit le
139 SF 13 mode d'impression en minuscules. Les lignes **140 à 145** affichent (impriment)
140 "FROM" *from earliest date* (à partir de la date la plus ancienne); les lignes **146 à 151**
141 XEQ 03 affichent (impriment) *through most recent date* (jusqu'à la date la plus récente).
142 " " Les lignes 141, 145, 147 et 151 appellent un sous-programme à la ligne 155 qui
143 RCL 03 affiche (imprime) le contenu du registre ALPHA. La ligne 152 choisit le mode
144 ADATE d'impression en majuscules. La ligne **153** met à 1 l'indicateur binaire
145 XEQ 03 d'autorisation d'impression; ceci assure que les instructions `AVIEW` suivantes
146 "THROUGH" vont soit imprimer les données du récapitulatif soit arrêter l'exécution afin que
147 XEQ 03 vous puissiez noter chaque donnée affichée. La ligne **154** provoque le
148 " " branchement à la ligne 161.
149 RCL 02
150 ADATE
151 XEQ 03
152 CF 13
153 SF 21
154 GTO 00

155 ♦ LBL 03 Les lignes **155 à 160** affichent (impriment) une ligne du titre. Ce sous-programme
156 ADV est appelé par les lignes 141, 145, 147 et 151. La ligne **156** fait avancer
157 AVIEW l'imprimante (si elle existe). La ligne **157** affiche le contenu du registre ALPHA;
158 FC? 21 si une imprimante est présente et sous tension, le contenu du registre ALPHA est
159 PSE imprimé également. Si l'imprimante est absente ou hors tension, les lignes **158 et**
160 RTN **159** prolongent l'affichage (l'indicateur binaire 21 a été désarmé par la ligne 135
 si l'instruction d'impression `PRA` a provoqué une erreur). La ligne **160** fait
 reprendre l'exécution à la ligne 142, 146, 148 ou 152.

Rappel des enregistrements de temps. Ces sous-programmes testent si une tâche dispose de données pour la période de récapitulatif, et rappelle ces données en mémoire étendue. Le programme exécute une seule fois le premier sous-programme bref; le second sous-programme, une fois par tâche et les deux derniers sous-programmes une fois par jour et par tâche.

161 ♦ LBL 00 Les lignes **161 à 164** préparent l'accession à la première tâche. La ligne **162** choisit
162 FIX 1 le format numérique de la sortie. Les lignes **163 et 164** mettent le pointeur de
163 CLX TRECS (le fichier courant) au premier caractère du premier enregistrement qui est
164 SEEKPT la première lettre du premier nom de tâche.

165 ♦ LBL 07
166 RCL 04
167 .1
168 %
169 8
170 +
171 STO 05
172 CLX
173 STO 06
174 GETREC
175 RCLPT
176 INT
177 RCL 01
178 +
179 .001
180 +
181 SF 25
182 SEEKPT
183 FC?C 25
184 GTO 10
185 ADV
186 ADV
187 AVIEW
188 LASTX
189 -
190 SEEKPT
191 GETREC

Les lignes **165 à 191** préparent le rappel de l'enregistrement de temps d'une tâche à partir de la mémoire annexe. Les lignes **166 à 171** placent dans R_{05} le nombre de commande de boucle *8.nnn*. Les lignes **172 et 173** effacent R_{06} , qui accumule l'horaire total de cette tâche. La ligne **174** rappelle le nom de la tâche dans le registre ALPHA. Avant d'afficher (imprimer) le nom de la tâche, le programme vérifie que l'enregistrement de temps de cette tâche contient les données relatives à la période du récapitulatif. Les lignes **175 à 178** calculent la valeur du pointeur de *most recent date* dans cet enregistrement (si cet enregistrement ne contient pas de multiplet pour *most recent date*, il ne contient pas de données pour la période de récapitulation). Les lignes **179 et 180** avancent la valeur du pointeur d'un caractère pour s'assurer que le multiplet de fin d'enregistrement de temps n'est pas pris pour des données. Les lignes **181 à 184** tentent de mettre le pointeur à cette valeur, et s'ils n'y trouvent pas de multiplet provoquent le branchement à la ligne 243. (Si cette tâche est trop récente pour la période de récapitulatif, toutes les tâches suivantes sont à fortiori trop récentes). Les lignes **185 à 187** font avancer l'imprimante deux fois et affichent (impriment) le nom de la tâche. Les lignes **188 à 190** placent le pointeur sur *most recent date* et la ligne **191** rappelle jusqu'à 24 multiplats de cet enregistrement de temps à destination du registre ALPHA.

192 ♦ LBL 08
193 254
194 ATOX
195 X>Y?
196 GTO 02
197 X≠0?
198 GTO 01
199 GETREC
200 GTO 08

Les lignes **192 à 200** rappellent et testent chaque multiplet. Ce sous-programme et le suivant forment une boucle. Les lignes **193 et 194** placent 254 dans le registre Y et la valeur décimale d'un multiplet dans le registre X. (Pour les exécutions séquentielles de ce programme, la ligne 194 déplace des multiplats de ALPHA dans X de façon à ce que les multiplats se suivent en mémoire étendue.) Quand le multiplet « fin d'enregistrement de temps » (valeur décimale 255) apparaît, les lignes **195 et 196** font sortir de la boucle en effectuant le branchement à la ligne 215. Les lignes **197 et 198** branchent à la ligne 201 à moins que le registre ALPHA soit vide. Si le registre ALPHA est vide et que le multiplet de fin d'enregistrement de temps n'est pas apparu, il y a davantage de multiplats dans cet enregistrement de temps à rappeler en mémoire étendue ; la ligne **199** rappelle jusqu'à 24 multiplats supplémentaires et la ligne **200** branche à la ligne 192.

201 ♦ LBL 01	Les lignes 201 à 214 stockent en mémoire principale la valeur du multiplet correspondant à chaque jour. Les lignes 202 et 203 changent les multiplets relatifs à un horaire nul (valeur décimale 254) pour une valeur décimale de 0. Les lignes
202 X=Y?	204 et 205 ajoutent à R_{06} (horaire total de cette tâche) et à R_{07} (horaire total pour
203 CLX	toutes les tâches) la valeur du multiplet. Pour une valeur de multiplet nnn , les
204 ST+ 06	lignes 206 à 208 renvoient $nnn.nnn$ dans X. Les lignes 209 et 210 rappellent
205 ST+ 07	$.ppp$, où ppp est la valeur accumulée de toutes les tâches de ce jour. Les lignes 211
206 .1	et 212 placent $nnn.qqq$ dans le registre correspondant à ce jour, où $qqq = nnn +$
207 %	ppp . Les lignes 213 et 214 incrémentent le nombre de commande de boucle, et à
208 +	moins que la période de récapitulatif ne soit terminée, provoque le branchement à
209 RCL IND 05	la ligne 192.
210 FRC	
211 +	
212 STO IND 05	
213 ISG 05	
214 GTO 08	

Sortie du récapitulatif de chaque tâche. Après que chaque tâche ait été rappelée de mémoire annexe en mémoire principale, les sous-programmes principaux affichent ou impriment le récapitulatif de cette tâche avant d'appeler la tâche suivante.

215 ♦ LBL 02	Les lignes 215 à 226 préparent l'affichage (impression) de l'horaire quotidien de
216 8.007	cette tâche. La ligne 216 introduit un nombre de commande de boucle
217 RCL 05	représentant une seule exécution du prochain sous-programme: il accède
218 INT	uniquement à R_{08} (uniquement pour <i>most recent date</i>). Les lignes 217 à 220
219 9	calculent combien de registres après R_{08} (jours avant <i>most recent date</i>) reçoivent des
220 -	données pour cette tâche. (La tâche peut ne pas disposer de données au début de la
221 +	période de récapitulatif.) Le résultat m sert deux fois. D'abord les lignes 221 et
222 STO 05	222 placent un nombre de commande de boucle $(8 + m).007$ dans R_{05} , où $R_8 + m$
223 RCL 02	est le registre de plus grande adresse qui reçoit un multiplet pour cette tâche. (La
224 LASTX	ligne 217 rappelle la valeur finale du nombre de commande de boucle à partir du
225 CHS	sous-programme précédent; le registre auquel le sous-programme précédent a eu
226 DATE+	accès en dernier lieu sera accédé en premier par le prochain sous-programme.)
	Les lignes 223 à 226 retiennent m jours de <i>most recent date</i> ; le résultat est la première
	date de la période de récapitulatif de cette tâche.

227 ♦ LBL 09 Les lignes **227 à 242** affichent (impriment) le récapitulatif de cette tâche. Le
228 RCL IND 05 programme exécute une boucle allant de la ligne 228 à la ligne 232 une fois par
229 INT jour. Les lignes **228 à 229** rappellent la valeur du multiplet de l'horaire de cette
230 XEQ 00 tâche pour ce jour. La ligne **230** appelle un sous-programme à la ligne 264 qui
231 DSE 05 affiche (imprime) la date et l'heure. Les lignes **231 et 232** décrémentent le nombre
232 GTO 09 de commande de boucle et, s'il reste des jours, provoquent le branchement à la
233 RCL 06 ligne 227. La ligne **233** rappelle la valeur du multiplet de l'horaire total de cette
234 XEQ 01 tâche, et la ligne **234** appelle un sous-programme à la ligne 278 qui affiche
235 RCLPT (imprime) **TOTAL** et les heures. Les lignes **235 à 240** tentent de placer le pointeur
236 INT de TRECS au nom de la prochaine tâche. S'il y a une autre tâche dans TRECS, les
237 1 lignes **241 et 242** effectuent un branchement à la ligne 165.
238 +
239 SF 25
240 SEEKPT
241 FS?C 25
242 GTO 07

Sortie du récapitulatif final. Une fois que toutes les tâches ont été récapitulées, le programme affiche (imprime) un récapitulatif final de l'horaire total de chaque jour et de l'horaire total de la période complète.

243 ♦ LBL 10 Les lignes **243 à 252** préparent l'affichage (impression) du récapitulatif final. Les
244 RCL 04 lignes **244 à 247** stockent *mm.007* dans R_{08} , où R_{mm} est le registre contenant
245 .007 l'horaire quotidien de *earliest date*. Les lignes **258 et 249** font avancer l'imprimante
246 + deux fois. Les lignes **250 et 251** affichent (impriment) **SUMMARY**. La ligne **252**
247 STO 05 rappelle *earliest date*.
248 ADV
249 ADV
250 "SUMMARY"
251 AVIEW
252 RCL 03

253 ♦ LBL 11 Les lignes **253 à 263** affichent (impriment) le récapitulatif final. Le programme
254 RCL IND 05 exécute une boucle allant de la ligne 254 à la ligne 260 une fois par jour. Les lignes
255 FRC **254 à 257** rappellent la valeur du multiplet de l'horaire total de ce jour, et la ligne
256 1 E3 **258** appelle un sous-programme à la ligne 264 qui affiche (imprime) la date et les
257 * heures. Les lignes **259 et 260** décrémentent le nombre de commande de boucle, et
258 XEQ 00 s'il reste des jours dans la période, provoquent un branchement à la ligne 253. La
259 DSE 05 ligne **261** rappelle la valeur du multiplet de l'horaire total pour la période
260 GTO 11 complète et la ligne **262** appelle un sous-programme à la ligne 278 qui affiche
261 RCL 07 (imprime) **TOTAL** et l'horaire.
262 XEQ 01
263 GTO 03

Sous-programmes de sortie. Ces sous-programmes affichent (impriment) respectivement la date, **TOTAL**, et l'horaire. Le programme appelle les premier et second sous-programmes, les deux transférant l'exécution au troisième, lequel fait reprendre l'exécution à la ligne suivant l'appel au sous-programme.

<p>264 ♦ LBL 00 265 X<>Y 266 1 267 RCL Y 268 " " 269 ADATE 270 "†" 271 DOW 272 X=Y? 273 ADV 274 RDN 275 DATE+ 276 X<>Y 277 GTO 02</p>	<p>Les lignes 264 à 277 placent la date dans le registre ALPHA. Ce sous-programme est appelé par les lignes 230 et 258 ; il s'attend à trouver la date voulue dans Y et la valeur du multiplet représentant l'horaire quotidien dans X. Les lignes 265 à 267 placent la date dans X et Z, 1 dans Y, et la valeur du multiplet dans T. Les lignes 268 à 270 placent la date dans ALPHA. Les lignes 271 à 273 font avancer l'imprimante si la date est un lundi. La ligne 274 fait passer 1 dans X, la date dans Y, et la valeur du multiplet dans Z. La ligne 275 calcule la date suivante (en préparation pour la prochaine exécution de la boucle). La ligne 276 renvoie la valeur du multiplet dans X (où elle se trouvait avant ce sous-programme). La ligne 277 branche à la ligne 280.</p>
<p>278 ♦ LBL 01 279 "TOTAL"</p>	<p>Les lignes 278 et 279 placent TOTAL dans ALPHA. Ce sous-programme est appelé par les lignes 234 et 262.</p>
<p>280 ♦ LBL 02 281 10 282 / 283 LASTX 284 X>Y? 285 "†" 286 X†2 287 X>Y? 288 "†" 289 ARCL Y 290 AVIEW 291 RCL Z 292 RTN</p>	<p>Les lignes 280 à 292 annexent l'horaire dans ALPHA, dont elles affichent (impriment) le contenu total. Les lignes 281 et 282 convertissent la valeur du multiplet en heures et dixièmes d'heures. Les lignes 283 à 288 formatent l'impression des heures avec justification à droite. La ligne 289 annexe l'horaire à ALPHA et la ligne 290 affiche (imprime) son contenu total. La ligne 291 rappelle la prochaine date dans X (en préparation pour la prochaine exécution de la boucle). La ligne 292 fait reprendre l'exécution à la ligne 231, 235, 259 ou 263.</p>

Effacement de jours ou tâches.

<p>293 ♦ LBL 03 294 CF 21 295 CLA 296 14 297 PASN 298 15 299 PASN 300 25 301 PASN 302 SF 27</p>	<p>Les lignes 293 à 302 préparent l'affichage du menu d'effacement. La ligne 294 désarme l'indicateur binaire d'autorisation d'impression. Les lignes 295 à 301 annulent toute affectation des touches [D], [E], [J]. La ligne 302 valide le clavier utilisateur.</p>
<p>303 ♦ LBL 12 304 "CLEAR D, J? E" 305 PROMPT 306 TONE 5 307 GTO 12</p>	<p>Les lignes 303 à 307 affichent le menu d'effacement. Les lignes 304 et 305 vous demandent d'appuyer sur la touche [D] (effacement de jours), [J] (effacement d'une tâche), ou [E] (sortie). (Quand vous appuyez sur l'une de ces touches, l'exécution commence au label correspondant.) Si vous avez appuyé par inadvertance sur [R/S], la ligne 306 émet une tonalité et la ligne 307 effectue un branchement à la ligne 303.</p>

Effacement d'une tâche.

<p>308 ♦ LBL J 309 CF 23 310 "NAME OF JOB?" 311 AON 312 PROMPT 313 AOFF 314 FC?23 315 GTO 12 316 CLX</p>	<p>Les lignes 308 à 316 préparent l'effacement d'une tâche. Les lignes 309 à 312 vous demandent le nom de la tâche que vous voulez effacer. Les lignes 311 et 313 valident et invalident automatiquement le clavier alphabétique. Si vous appuyez sur [R/S] sans taper de nom, les lignes 314 et 315 provoquent un branchement à la ligne 303. La ligne 316 fournit une valeur de pointeur pour le début de TRECS en tant que valeur initiale pour le prochain sous-programme.</p>
--	---

317 ♦ LBL 13
318 SEEKPT
319 POSFL
320 X<0?
321 GTO 00
322 2
323 MOD
324 X=0?
325 GTO 01
326 RCLPT
327 int
328 1
329 +
330 GTO 13

Les lignes **317 à 330** localisent la tâche à effacer. Les lignes **318 et 319** recherchent dans TRECS la tâche que vous avez nommée dans ALPHA, commençant la recherche à la valeur du pointeur se trouvant dans X. Si le nom n'existe pas dans TRECS, les lignes **320 et 321** provoquent le branchement à la ligne 331 à un sous-programme d'erreur. Si la valeur du pointeur renvoyée par la ligne 319 est valide pour un nom, les lignes **322 à 325** provoquent le branchement à la ligne 337. Si la valeur du pointeur n'est pas valide (c'est-à-dire si les multipliants d'un enregistrement de temps coïncident avec le nom), les lignes **326 et 330** calculent la valeur du pointeur pour le prochain enregistrement et branchent à la ligne 317.

331 ♦ LBL 00
332 TONE 5
333 "NO SUCH NAME"
334 AVIEW
335 PSE
336 GTO 12

Les lignes **331 à 336** constituent un sous-programme d'erreur. La ligne **332** émet une tonalité; les lignes **333 à 335** affichent un message d'erreur; et la ligne **336** effectue le branchement à la ligne 303.

337 ♦ lbl 01
338 DELREC
339 DELREC
340 GTO 12

Les lignes **337 à 340** effacent la tâche de TRECS. La ligne **338** efface le nom de la tâche et la ligne **339** efface l'enregistrement de temps de la tâche. La ligne **340** branche à la ligne 303.

Effacement de jours.

341 ♦ LBL D
342 SF 00
343 XEQ 00
344 1.001
345 RCL 01
346 X<Y?
347 X<>Y
348 CLA
349 255
350 XTOA

Les lignes **341 à 350** préparent l'effacement des jours. Les lignes **342 et 343** appellent un sous-programme à la ligne 08 qui détermine la première et la dernière date à effacer. (L'objectif principal du sous-programme de la ligne 08 est de déterminer la période sur laquelle porte le récapitulatif; l'armement de l'indicateur binaire 00 modifie le fonctionnement du sous-programme si bien qu'elle détermine la période à effacer.) Les lignes **344 à 347** renvoient 1.001, où 001 est la valeur du pointeur de caractère pour la date la plus récente qui est à effacer. La ligne **344** introduit la valeur minimale (au moins un multipliant par jour doit être laissé dans chaque enregistrement de temps) et la ligne **345** rappelle la valeur qui correspond à la date la plus récente à effacer. Les lignes **346 et 347** placent dans X la plus grande de ces valeurs. Les lignes **348 à 350** placent 255 dans X et dans ALPHA le multipliant de fin d'enregistrement de temps.

351 ♦ LBL 14
352 DELCHR
353 INSCR
354 X<>Y
355 2
356 +
357 SF 25
358 SEEKPT
359 FC?C 25
360 GTO 12
361 X<>Y
362 GTO 14

Les lignes **351 à 362** effacent des jours dans l'enregistrement de temps de chaque tâche. Le programme exécute cette boucle une fois par enregistrement de temps qui contient des données concernant la période d'effacement. La ligne **352** efface tous les multiplètes dans la période de récapitulation et la ligne **353** insère un nouveau multiplète fin de temps d'enregistrement. Les lignes **354 à 358** essaient de mettre le pointeur au multiplète de la date la plus récente à effacer dans l'enregistrement de temps de la prochaine tâche. S'il n'y a pas de multiplète à cet endroit, les lignes **359 et 360** provoquent le branchement à la ligne 303 (ceci arrive quand il n'y a plus d'enregistrement ou quand cette tâche et toutes les suivantes n'ont pas de données concernant la période d'effacement). La ligne **361** renvoie 255 dans X et la ligne **362** effectue le branchement à la ligne 351.

Sortie du programme SIGMA.

363 ♦ LBL E
364 RCL 0
365 STOFLAG
366 CLX
367 END

Les lignes **363 à 367** terminent l'exécution de SIGMA. Les lignes **364 et 365** redonnent aux indicateurs binaires leur valeur précédente. La ligne **366** efface du registre X la donnée d'état des indicateurs binaires.

Messages d'erreur et d'état

Cette annexe liste tous les messages d'erreur et d'état du HP-41CX.

Lorsque le HP-41CX rencontre une opération illégale, il ne l'exécute pas mais affiche un message d'erreur. Vous pouvez effacer ce message en appuyant sur \square . Si l'erreur est apparue pendant l'exécution d'un programme, vous pouvez placer le HP-41 en mode programme pour afficher la ligne ayant provoqué l'erreur.

Certains messages sont appelés messages d'état car ils vous informent d'un état du HP-41 et non pas d'une condition d'erreur.

Les variables x , y et z ci-dessous se réfèrent aux contenus des registres X, Y et Z respectivement.

Affichage	Fonctions	Signification
ALPHA DATA	Mathématique Temps Mémoire annexe Toute autre fonction utilisant une donnée numérique	Donnée alphabétique utilisée pour une fonction demandant une donnée numérique : le registre X (ou le registre Y ou Z) contient une donnée alpha.
CHKSUM ERR	GETP GETSUB	Une partie du fichier programme a été perdue.
DATA ERROR	Mathématique SDEV Temps \square T+X TONE MEAN OCT DEC FACT %CH	Opération mathématique illicite avec l'opérande donnée (division par zéro, racine carrée d'un nombre négatif). Nombre non valide dans le registre X. $x \leq 0$ et $y = 0$ ou x non entier et $y < 0$. $x < -9999,595999$ ou $x > 9999,595999$ ou MM ou SS > 59 . $x \geq 10$ ou $x < 0$. $n = 0$. $ x > 1073741823_{10}$ ou x non entier. $x = 8$ ou 9 ou x non entier. $x < 0$ ou non entier. $y = 0$.

Affichage

Fonctions

Signification

DATA ERROR
(suite)

FF

|Gr

EMQ

 $|n| \geq 10.$

GETKEYX

 $x \geq 100.$

ARDT

POSA

XTOA

X<>F

 $x > 255.$

PSIZE

CLRGX

BEEKPT

BEEKPTA

SWPT

 $x > 999.$

CRLFAS

CRFLD

 $x = 0.$

EMDIRX

CLALMX

RCLALM

RSZFL

 $x = 0$ ou $x > 999.$

STOFLAG

x (ou y , si x est une gamme d'indicateurs de la forme $dd.ff$) n'a pas été obtenue par `RCLFLAG`.

X=NN?

X≠NN?

X<NN?

X<=NN?

X>NN?

X>=NN?

Le registre Y contient des données alphanumériques autres que X, Y, Z, T ou L, ou $y > 999.$

DATA ERROR X

DDAYS

x est une date qui n'existe pas, où l'année est composée de plus de quatre chiffres.

XYZALM

$x \geq 24$ ou le contenu au format **HH.MMSS** n'existe pas.

DATA ERROR Y

DATE+

DDAYS

XYZALM

y est une date qui n'existe pas, ou l'année est composée de plus de 4 chiffres.

DATA ERROR Z








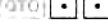





XYZALM

La valeur de $z \geq 10000$ heures où le contenu au format **HHHH.MM.SS** n'existe pas.

Affichage	Fonctions	Signification
DUP FL <i>(duplicate file)</i>	Mémoire annexe	Un fichier portant le même nom existe déjà en mémoire annexe. Texte, données et fichiers programme ne doivent pas ainsi avoir le même nom.
END OF FL <i>(end of file)</i>	Mémoire annexe	Pointeur à la fin du fichier. (Avec <code>SAVER</code> et <code>SEEKPTA</code>) le fichier spécifié devient le fichier en cours, bien que le fichier et les pointeurs ne changent pas.)
	<code>APPCHR</code> <code>APPREC</code> <code>INSCHR</code> <code>INSREC</code>	Il n'y a pas assez de place pour insérer le caractère ou l'enregistrement.
	<code>GETAS</code> <code>SAVEAS</code>	
END OF REC <i>(end of record)</i>	<code>SEEKPT</code> <code>SEEKPTA</code>	Pointeur à la fin de l'enregistrement.
ERROR = Dnn^* <i>(delta split error in R_{nn})</i>	<code>SW</code>	Le contenu stocké dans R_{nn} ou R_{nn-1} n'est pas au format HH.MMSS ; ou la fraction stockée dans R_{nn} est plus petite que celle stockée dans R_{nn-1} .
ERROR = Rnn^* <i>(split error in R_{nn})</i>	<code>SW</code>	La partie entière dans R_{nn} dépasse 99.
FL NOT FOUND <i>(file not found)</i>	Mémoire annexe	Le fichier spécifié (qui doit être celui en cours) n'existe pas en mémoire annexe. (Après l'effacement d'un fichier, il n'y a pas de fichier en cours.)
FL SIZE ERR <i>(file size error)</i>	<code>RESZFL</code>	Le nouveau fichier spécifié (x) est plus petit que le fichier en cours ; il se peut que certaines données soient effacées.
FL TYPE ERR <i>(file type error)</i>	Mémoire annexe	Le fichier en cours n'est pas d'un type adapté à la fonction exécutée.
KEYCODE ERR <i>(keycode error)</i>	<code>PASM</code>	Le code numérique de la touche n'est pas valable.

* Ce message d'erreur est seulement une information et n'a pas l'effet d'une véritable condition d'erreur.

Affichage	Fonctions	Signification
MEMORY LOST		La mémoire permanente a été effacée et réinitialisée.
NAME ERR (file name error)	Mémoire annexe	Le nom du fichier n'est pas valable (le registre Alpha est vide), ou le registre Alpha contient 7 octets de valeur décimale 255.
	PCLPS SAVEP	Le programme n'existe pas en mémoire principale.
NO	Indicateurs Conditionnel	<i>Message d'état.</i> Le résultat d'un test d'indicateur ou d'un test conditionnel est faux.
NO DRIVE (no drive device present)	GETAS SAVEAS	Le module HP-IL n'est pas enfiché ou l'unité de stockage de masse n'est pas sur la boucle d'interface.
NONEXISTENT	Stockage Rappel Mémoire annexe Test conditionnel CLRGX SWPT	Un ou plusieurs registres spécifiés n'existent pas.
	ASN GTD XEQ Alarmes	Le label d'un programme spécifié ou appelé n'existe pas. (Si la fonction nécessite un label global et qu'un label global a été spécifié, cela peut provoquer cette erreur.)
	ASN XEQ Alarmes	La fonction appelée n'existe pas. Pour des alarmes, la fonction appelée doit être une fonction programmable du catalogue 2. Si une fonction du catalogue 2 est appelée, le module correspondant doit être enfiché dans le HP-41.
	STOFLAG	Un ou plusieurs indicateurs spécifiés sont en dehors de la gamme 0 à 43.
NO ROOM	Mémoire annexe	Il n'y a pas assez de place en mémoire annexe pour le programme ou le fichier spécifiés.
	GETP GETSUB	Il n'y a pas assez de place en mémoire principale pour le programme spécifié.
	PSIZE	(Lorsqu'elle est exécutée comme une instruction de programme.) Pas assez de place en mémoire principale.

Affichage	Fonction	Signification
NO ROOM (suite)		Pas assez de place en mémoire annexe pour accroître la taille du fichier.
		Pas assez de place en mémoire pour armer l'alarme ; ou le nombre de registres maximum (253) pour le stockage de l'alarme risque d'être dépassé.
	Éditeur de texte*	Pas assez de place pour ajouter des caractères ou des enregistrements.
NO SUCH ALM (no such alarm)	Alarmes	Cette alarme n'existe pas.
NULL	Toutes.	<i>Message d'état.</i> Fonction annulée en maintenant la touche correspondante enfoncée.
OUT OF RANGE	Numérique	Nombre trop grand. Dépassement de capacité : ± 9.999999999999 .
PACKING TRY AGAIN		} La date obtenue n'existe pas.
		
		} <i>Message d'état.</i> Compacte la mémoire programme ; répète l'opération. Si TRY AGAIN apparaît à nouveau, c'est qu'il n'y a plus assez de place en mémoire principale. Vous devez augmenter la mémoire programme.
		
		
		
		
	Mode programme	
		} Compacte la mémoire ; répète l'opération. Si TRY AGAIN apparaît à nouveau, c'est qu'il n'y a plus assez de place pour augmenter la mémoire programme.
		
PRIVATE	Lecteur de cartes ROM personnalisées	Essaie de visualiser un programme privé ; reportez-vous au manuel d'utilisation du lecteur de cartes HP 82104A.
RAM		Essaie de dupliquer en RAM un programme dont le label global spécifié existe déjà en mémoire principale (RAM).

* Ce message d'erreur est seulement une information et n'a pas l'effet d'une véritable condition d'erreur.


Affichage	Fonctions	Signification
REC TOO LONG	Éditeur de texte <input data-bbox="336 233 413 261" type="button" value="APPCHR"/> <input data-bbox="336 266 413 293" type="button" value="INSCHR"/>	} Enregistrement trop long (longueur maximale: 254). (Invalide l'éditeur de texte qui était validé.)
ROM		} Vous ne pouvez pas accéder à un programme se trouvant dans une mémoire morte (ROM, module d'application).
YES	Indicateurs Test conditionnel	} <i>Message d'état.</i> Le résultat d'un test d'indicateurs ou d'un test est validé.

Pour en savoir davantage sur les alarmes périmées

Table des matières

Conditions provoquant l'exécution des alarmes périmées	361
Condition arrêt/horloge (Off/Clock)	361
Condition d'alarme	361
Réponse des alarmes périmées en condition d'alarme	362
Changements de mode	362
Interruption d'une alarme périmée par une autre alarme périmée	363
Alarmes et niveaux de sous-programme	363
Exemple de séquence d'alarmes périmées	363


Lorsque pour l'une des raisons décrites au chapitre 16 sous le titre « Alarmes périmées », une alarme n'est pas activée lorsqu'elle arrive à échéance, le HP-41 la garde en mémoire jusqu'à son activation ou sa suppression. Cette opération vous permet de vous souvenir qu'une alarme n'a pas été utilisée pour le but prévu. Si vous laissez plusieurs alarmes périmées s'accumuler en mémoire, des séquences d'activation automatiques d'alarmes échues peuvent survenir (s'il y a des alarmes périmées en mémoire, l'ordre dans lequel elles seront activées peut devenir complexe). Si vous prévoyez d'utiliser dans vos applications des alarmes périmées, les informations de cette annexe vous seront utiles. Les alarmes simultanées sont activées dans le même ordre que les alarmes périmées.

Si la mémoire contient des alarmes périmées, elles seront automatiquement activées lorsque vous éteindrez le HP-41 ou appuierez sur  **ON**. Cette opération a pour but de vous rappeler l'existence de ces alarmes. De même, si une alarme arrive à échéance alors que des alarmes périmées *inactivées* existent, le HP-41 active ces dernières avant l'alarme échue*. Cette annexe décrit les règles régissant les séquences d'activation dans ces deux cas (consultez la description et la classification des alarmes périmées au chapitre 16).



* Une alarme périmée est ignorée (voir chapitre 16) soit parce qu'elle a été sélectionnée dans le passé, soit parce qu'une fonction de changement de temps a été sélectionnée (**SETIME**, **SETDATE**, **T-X** ou **CORRECT**).

Conditions provoquant l'exécution des alarmes périmées

Condition arrêt/horloge (Off/Clock)

L'extinction du HP-41 ou l'affichage de l'horloge avec  active la condition arrêt/horloge. Si des alarmes périmées existent lorsque cette condition apparaît, le HP-41 *essaie* de toutes les activer, en commençant par la plus ancienne. S'il rencontre une alarme contrôle ou une alarme conditionnelle périmée, le HP-41 s'éteint momentanément, ce qui a pour effet de supprimer la condition arrêt/horloge, puis se rallume dans la condition d'alarme décrite au paragraphe suivant. L'alarme de commande/conditionnelle est alors exécutée dans cette condition.

En condition arrêt/horloge

- *Tant que le HP-41 ne rencontre pas d'alarme de commande/conditionnelle périmée*, il active les alarmes-messages dans l'ordre chronologique en commençant par la plus ancienne. Chaque alarme termine son cycle d'activation avant que la suivante ne soit activée. Ces alarmes ne s'interrompent pas mutuellement.
- Si vous appuyez sur la touche  pendant l'activation d'une alarme-message périmée, celle-ci s'arrête sans être acceptée et le HP-41 s'éteint ou affiche l'horloge.
- Si aucune alarme de commande ou conditionnelle périmée existe et si vous n'appuyez pas sur  pendant l'activation, le HP-41 s'éteint ou affiche l'horloge après l'activation de toutes les alarmes périmées.

Condition d'alarme

Cette condition d'alarme est déclenchée lorsque :

- Une alarme future arrive à échéance.
- Une alarme de commande se trouvant dans la condition « Arrêt/Horloge » lance un programme ou exécute une fonction.

Lorsque l'on se trouve dans la condition d'alarme, le calculateur n'active que les alarmes périmées non activées existant en mémoire (en commençant par la plus ancienne et en procédant par ordre chronologique). Les alarmes périmées activées sont ignorées et restent en mémoire. Lorsqu'une alarme future arrive à échéance alors qu'il y a des alarmes périmées non activées, le calculateur passe en condition d'alarme et l'alarme future passe à l'état d'alarme périmée non activée. Elle sera activée à son tour après que toutes les alarmes périmées non activées plus anciennes auront été activées par le calculateur.

Réponse des alarmes périmées en condition d'alarme

Les termes ci-dessous sont utilisés dans la suite de cette annexe pour décrire les modes de fonctionnement du calculateur qui ont une influence sur la réaction des alarmes.

Réponses des alarmes périmées

Type d'alarme	Mode du calculateur			
	Arrêt (Off)	Horloge (clock)	Clavier (keyboard)	En cours (Running)
Commande de non-interruption (↑)	Exécution du programme	Exécution du programme	Série de tonalités et clignotement de l'affichage	Émet deux tonalités et passe à l'état d'alarme périmée activée
Commande d'interruption (↑↑)	Exécution du programme			Exécute le programme comme un sous-programme du programme en cours
Message	Série de tonalités et clignotement de l'affichage			

- Arrêt (Off) : le calculateur est hors tension.
- Horloge (clock) : l'horloge est affichée.
- Clavier (keyboard) : le calculateur est sous tension, mais n'affiche pas l'horloge et n'exécute pas de programme.
- En cours (Running) : un programme est en cours d'exécution.

Le tableau ci-dessus résume les réponses du HP-41 lorsqu'une alarme arrive à échéance dans chacun des modes ci-dessus.

Changements de mode

La fonction ou le programme spécifié par une alarme de commande peut modifier le mode de fonctionnement du calculateur :

- Si une alarme de commande lance un programme, le calculateur passe immédiatement en mode « En cours » (le changement de mode se produit avant l'exécution de la première instruction du programme).
- Une fonction exécutée par une alarme de commande peut également changer le mode de fonctionnement. Si, par exemple, l'horloge est affichée lorsqu'une alarme qui exécute la fonction **PRX** (impression du registre X) de l'imprimante est activée, le calculateur passera du mode « Horloge » au mode « Clavier ». De la même façon, si le calculateur est en cours d'exécution d'un programme lorsqu'une alarme qui provoque l'exécution de la fonction **CLOCK** est activée, le calculateur passera du mode « En cours » au mode « Horloge ».

Interruption d'une alarme périmée par une autre alarme périmée

- Un programme lancé par une alarme de commande/conditionnelle périmée sera suspendu temporairement par les alarmes périmées non activées suivantes avant l'exécution de la première instruction du programme.
- Une alarme message ou une fonction exécutée par une alarme de commande/conditionnelle périmée ne peut pas être interrompue par une alarme périmée non-activée.

Alarmes et niveaux de sous-programme

Toute alarme programme qui interrompt une alarme programme précédente se comporte comme un sous-programme. S'il y a plusieurs alarmes de commande périmées exécutant des programmes (ce qui est peu probable puisque les alarmes de commande périmées devraient être des alarmes périmées non-activées), les alarmes utilisent plusieurs niveaux de sous-programme.

Exemple de séquence d'alarmes périmées

Supposons que le calculateur soit *hors tension*, l'heure actuelle 9 h 59 du matin et que les quatre alarmes suivantes aient été enclenchées :

Registre ALPHA	Heure	Etat
MESSAGE 1	4 h 00 du matin	Périmée activée
† ABC	5 h 00 du matin	Périmée non activée
†† XYZ	6 h 00 du matin	Périmée non activée
MESSAGE 2	10 h 00 du matin	Enclenchée sur une heure future

Remarque : La situation décrite par cet exemple est peu probable (puisque, dans la plupart des applications, on ne trouve pas d'alarmes périmées non activées). Cependant, ce mélange d'alarmes facilite la compréhension de certains aspects de la réaction des alarmes.

Comme il y a des alarmes périmées non activées, lorsque l'on atteindra 10 h 00 du matin, l'alarme **MESSAGE 2** provoquera l'entrée en condition d'alarme. On aboutira ainsi à la séquence suivante :

1. L'alarme †**ABC** (l'alarme périmée *non activée* la plus ancienne) met sous tension le calculateur et lance le programme **ABC**. (C'est la première alarme activée. L'alarme **MESSAGE 1**, qui est plus ancienne, est une alarme périmée qui a été précédemment activée et, par conséquent, ne le sera pas de nouveau.)
2. L'alarme ††**XYZ** suspend immédiatement l'exécution du programme **ABC** et lance le programme **XYZ** comme un sous-programme de **ABC**.

3. L'alarme **MESSAGE 2** (qui est maintenant devenue une alarme périmée non-activée) interrompt immédiatement l'exécution du programme XYZ et fait clignoter le message **MESSAGE 2** à l'affichage (si vous n'acceptez pas l'alarme au clavier, le HP-41 commence à émettre des bips).
4. Après l'acceptation de l'alarme **MESSAGE 2** (ou l'achèvement de son cycle), le HP-41 exécute le programme XYZ. L'exécution revient ensuite au programme ABC (en supposant que XYZ n'ait pas éteint le HP-41 ou n'ait pas utilisé trop de niveaux de sous-programme).
5. Le programme ABC est exécuté.

Si le programme XYZ éteint le HP-41 (en exécutant OFF), le programme ABC ne sera pas exécuté. L'alarme †ABC ayant déjà été activée, elle n'existe plus en mémoire.

Si l'alarme XYZ était de type conditionnelle, elle aurait émis deux bips et serait devenue une simple alarme périmée (puisque l'alarme †ABC aurait fait passer le HP-41 en mode exécution). Consultez le tableau de la page 362 et le paragraphe « Changements de mode » en page 362. En règle générale, chaque fois qu'une alarme de commande ou conditionnelle périmée est activée et lance l'exécution d'un programme, toutes les alarmes conditionnelles périmées suivantes ne provoquent que l'émission de deux bips lorsqu'elles sont activées. Elles deviennent en outre des alarmes périmées activées.

Si le HP-41 est en mode Clavier au lieu d'être en mode Arrêt, la séquence est la suivante : l'alarme †ABC est activée comme une alarme message et provoque l'affichage de †ABC. Le programme XYZ est ensuite lancé (pas comme un sous-programme), est interrompu par l'alarme **MESSAGE 2** et est finalement exécuté.

Caractères nuls

Table des matières

Caractères nuls et registre Alpha	366
Traitement des caractères nuls	366

Caractères nuls et registre Alpha

Le caractère nul est représenté par le symbol — et correspond au code de caractère 0* ‡‡ . Le HP-41 ne génère normalement pas de caractère nul, vous pouvez cependant, sous certaines conditions, le placer dans des chaînes alphanumériques.

Le caractère nul n'étant pas courant, le HP-41 l'utilise comme un indicateur particulier. De ce fait, la présence de caractères nuls dans le registre Alpha peut provoquer des affichages inattendus comme décrits dans cette annexe.

Traitement des caractères nuls

Lors du traitement des caractères nuls, il est important de bien différencier le *registre Alpha* et l'*affichage Alpha*.

- Le registre Alpha contient toujours 24 caractères ; lorsqu'il est « vide », il contient en fait 24 caractères nuls. Lorsque vous entrez des caractères dans le registre Alpha par le côté droit, ils remplacent les caractères nuls. Tous les caractères nuls d'en-tête (ceux que vous avez entrés et ceux qui y étaient déjà) restent mais sont ignorés par le HP-41.
- L'affichage Alpha se compose de tous les caractères du registre Alpha après les nuls d'en-tête. Il commence par le premier (le plus à gauche) caractère non nul et comporte tous les caractères le suivant, *y compris les caractères nuls*.

Le HP-41 et ses fonctions considèrent toujours qu'une *chaîne alphanumérique commence au premier caractère non nul*. Les caractères nuls insérés entre des caractères non nuls sont conservés. Cependant, si vous effectuez une rotation d'une chaîne alpha jusqu'à ce qu'un caractère nul inséré devienne un nul d'en-tête, ce nul et tous les suivants sont perdus.

* Le caractère nul n'a aucun rapport avec le message NULL. (qui apparaît lorsqu'une fonction a été annulée).

‡‡ L'impression d'un caractère nul sur les imprimantes HP 82143A et HP 82162A est \blacklozenge (qui correspond aux codes de caractère 0 et 10).

Ajout de caractères. Si vous ajoutez un caractère au registre Alpha (en utilisant $\boxed{\text{I}}$), l'affichage diffère du contenu du registre Alpha *si le dernier caractère (avant l'ajout) était nul*.

Si le dernier caractère du registre Alpha est nul, alors — *lors de l'entrée de caractères à ajouter* — le HP-41 considère que le registre est vide et n'affiche que les caractères que vous ajoutez (l'indicateur de saisie (—) est présent à l'affichage lors de l'ajout des caractères). Le registre Alpha conserve *cependant* la chaîne originale et la combine avec la chaîne ajoutée.

Vous pouvez afficher le contenu total en appuyant sur $\boxed{\text{VIEW}}$ ou $\boxed{\text{ALPHA}} \boxed{\text{ALPHA}}$ (souvenez-vous que le HP-41 n'affiche jamais les nuls d'en-tête).

Suppression de caractères pendant l'ajout. Si vous utilisez $\boxed{\text{L}}$ ou $\boxed{\text{APPL}}$ et si le dernier caractère de la chaîne alpha est nul, l'utilisation de la touche $\boxed{\text{R}}$ pour supprimer le dernier caractère *efface le registre Alpha en entier* car lorsque le HP-41 rencontre un caractère nul lors d'une suppression de caractère, il suppose que ce doit être le premier des nuls d'en-tête et que le registre est vide, donc il efface tout.

Chaînes Alpha dans les registres de données ou de la pile. Si vous stockez une chaîne alpha contenant des nuls dans un registre de données ou de la pile, le HP-41 n'affiche pas (et n'imprime pas) ces nuls. Cependant, si vous rappelez ces chaînes dans le registre Alpha et les affichez ensuite ($\boxed{\text{BUIZ}}$), le HP-41 affiche *tous* les caractères de la chaîne (sauf les nuls d'en-tête).

Si vous imprimez le contenu Alpha d'un registre de données ou de la pile, le HP-41 n'imprime que les caractères à gauche du premier nul.

Un caractère nul inséré dans une chaîne alpha dans le registre X indique la fin de la chaîne dans laquelle la fonction $\boxed{\text{POSA}}$ (*position dans Alpha*) cherchera une correspondance avec le contenu du registre Alpha.

Nom de fichier. Les fonctions utilisant des noms de fichier dans le registre Alpha ignorent les nuls insérés dans le nom s'il y en a.

Opérations d'impression

Table des matières

Avance papier	368
Contrôle de l'exécution et de l'affichage avec les indicateurs 21 et 55	368
Heure et date sur les listages de programme	369

Avance papier

La fonction programmable **ADV** commande un saut de ligne du papier dans l'imprimante. Si aucune imprimante n'est connectée au HP-41, ce dernier ignore cette fonction (de même si l'imprimante est connectée mais mise hors tension ou si l'indicateur 21 (voir ci-dessous) est effacé).

Contrôle de l'exécution et de l'affichage avec les indicateurs 21 et 55

Le HP-41 arme ou désarme automatiquement les indicateurs 21 et 55 lors de la mise sous-tension. Normalement, ils sont tous les deux armés si une imprimante est connectée ou tous les deux désarmés dans le cas contraire.

En utilisant les fonctions **VIEW** et **AVIEW** et en manipulant l'indicateur 21 (qui peut être changé par l'utilisateur; 55 ne le peut pas), vous pouvez contrôler l'affichage des messages et des résultats pendant l'exécution de programmes; c'est-à-dire contrôler si l'exécution est interrompue par l'affichage du résultat ou si elle reprend immédiatement.

Les états des indicateurs 21 et 55 déterminent la façon dont les fonctions **VIEW** et **AVIEW** affectent un programme en cours d'exécution. Lorsque leurs états sont identiques (cas usuel, par défaut) le résultat est le suivant :

- S'il n'y a pas d'imprimante, **VIEW** ou **AVIEW** affiche le contenu du registre spécifié ou du registre Alpha jusqu'à ce qu'une autre instruction place une nouvelle donnée à l'affichage. **VIEW** et **AVIEW** n'arrêtent pas l'exécution d'un programme.
- Si une imprimante est connectée *et mise sous tension*, le HP-41 agit de la même façon que ci-dessus et imprime en outre les données affichées.

Il y a deux raisons d'utiliser **VIEW** et **AVIEW** dans un programme. 1) Un message peut vous indiquer ce que le programme est en train de traiter — quel sous-programme il exécute par exemple. Il n'est cependant pas nécessaire de conserver une trace de ces messages. 2) D'autres messages vous donnent les résultats de votre programme et vous voulez sans doute les conserver. Si vous n'avez pas d'imprimante, vous devez arrêter le programme lors de l'affichage d'un résultat pour le noter.

Remarquez que les opérations ci-dessus n'arrêtent pas l'exécution (pour que vous puissiez noter un résultat) même s'il n'y a pas d'imprimante connectée et imprime *tous* les messages ou données affichés par **VIEW** ou **VIEW** si une imprimante est présente. L'armement ou le désarmement de l'indicateur 21 avant l'exécution de **VIEW** ou **VIEW** vous permet de contrôler l'arrêt du programme pour l'affichage des messages qu'une imprimante soit présente ou non.

- **Désarmez l'indicateur 21 pour afficher sans imprimer les messages.** Si l'indicateur 21 est désarmé lors de l'exécution de **VIEW** ou **VIEW** et s'il n'y a pas d'imprimante (ou encore si l'imprimante est hors tension), les messages et les résultats sont affichés sans interruption du programme.

Si une imprimante est présente, le message est affiché mais non imprimé et l'exécution n'est pas interrompue.

- **Armez l'indicateur 21 pour enregistrer les résultats** — soit sur une imprimante soit à la main. Si l'indicateur 21 est armé lors de l'exécution de **VIEW** ou de **VIEW** et s'il n'y a pas d'imprimante *ou encore si l'imprimante est hors tension*, l'exécution s'arrête pour que vous puissiez noter le résultat. Il suffit d'appuyer sur **R/S** pour relancer l'exécution.

Si une imprimante est présente, le résultat est imprimé et l'exécution n'est pas interrompue.

Par conséquent, même avec une imprimante connectée vous pouvez choisir d'imprimer tous les affichages ou non. Si vous n'avez pas d'imprimante vous pouvez choisir d'interrompre ou non l'exécution pour les messages ou les résultats.

Heure et date sur les listages de programme

Si vous listez un programme sur une imprimante HP-IL (HP 82162A) en utilisant la fonction **PRP** ou **LIST** du module HP-IL (HP 82160A), le HP-41 affiche et imprime l'heure et la date en tête du listage (l'heure et la date affichées ne sont pas dans le registre X; appuyez sur **←** pour rappeler le contenu du registre X à l'affichage).

Modules d'extension mémoire

Table des matières

Utilisation des modules d'extension mémoire	370
Configurations	370
Installation des modules	371
Retrait des modules	371
Organisation de la mémoire annexe	372

Utilisation des modules d'extension mémoire

ATTENTION

Veillez à toujours éteindre le HP-41CX avant d'insérer ou de retirer des modules. Le non-respect de ce conseil peut endommager le HP-41CX ou perturber son fonctionnement.

Vous pouvez aisément identifier le module d'extension mémoire HP 82181A grâce au label « X MEMORY » imprimé sur son boîtier.

Configurations

Si vous ajoutez un seul module d'extension mémoire à votre HP-41CX, vous pouvez le placer dans un logement quelconque parmi les quatre disponibles sur le HP-41CX. Suivez la procédure ci-dessous pour l'installation du module.

Si vous ajoutez un second module d'extension mémoire, les deux modules doivent être positionnés conformément à l'une des configurations suivantes. N'installez jamais deux modules d'extension mémoire l'un au-dessus de l'autre.

Configurations possibles pour deux modules d'extension mémoire HP 82181A

X MEMORY	X MEMORY

X MEMORY	X MEMORY

X MEMORY	
	X MEMORY

	X MEMORY
X MEMORY	

Installation des modules

Procédure :

1. **Éteignez le HP-41CX !**
2. Tenez le HP-41CX clavier vers le haut.
3. Retirez le capuchon du logement.
4. Tenez le module comme indiqué ci-contre. Vous devez pouvoir lire « X MEMORY ».
5. Enfoncez le module dans le logement sans forcer.



Vous sentirez le module s'enclencher dans le logement lorsqu'il sera correctement positionné.

Retrait d'un module

Procédure :

1. **Éteignez le HP-41CX !**
2. Faites sortir la petite poignée avec votre ongle.
3. Tirez sur cette poignée pour retirer le module du logement.
4. Remettez un capuchon sur le logement.



Vous perdez le contenu du module lorsque vous le retirez du HP-41CX. **Vous pouvez en outre perdre le contenu des deux modules même si vous n'en retirez qu'un.** Pour éviter cela, veillez à suivre les règles ci-dessous lors du retrait de module.

- Si les modules d'extension mémoire ont été installés à différents moments, retirez-les dans l'ordre inverse de leur insertion.
- Si les modules ont été installés au même moment, retirez d'abord le module situé dans le logement 2 ou 4 puis l'autre (la numérotation des logements est indiquée au dos du HP-41CX).

Vous pouvez déterminer les fichiers qui disparaîtront avant de retirer un module. Vous pouvez si nécessaire supprimer des fichiers antérieurs moins importants de façon à assurer la sauvegarde de certains fichiers.

1. Vérifiez la taille de chaque fichier en utilisant la fonction `EMDIR` décrite en page 206.
2. Pour chaque fichier, ajoutez deux registres pour l'en-tête.
3. Ajoutez le nombre total des registres des fichiers 1 et 2 pour connaître la position du dernier registre du second fichier.
4. Ajoutez le nombre total de registres de chaque fichier successif au total précédent pour connaître la position du dernier registre de chaque fichier.

Les fichiers pour lesquels la position du dernier registre dépasse 364 sont perdus lors du retrait d'un des modules d'extension mémoire. Ceux pour lesquels la position du dernier registre dépasse 126 sont perdus lors du retrait des deux modules.

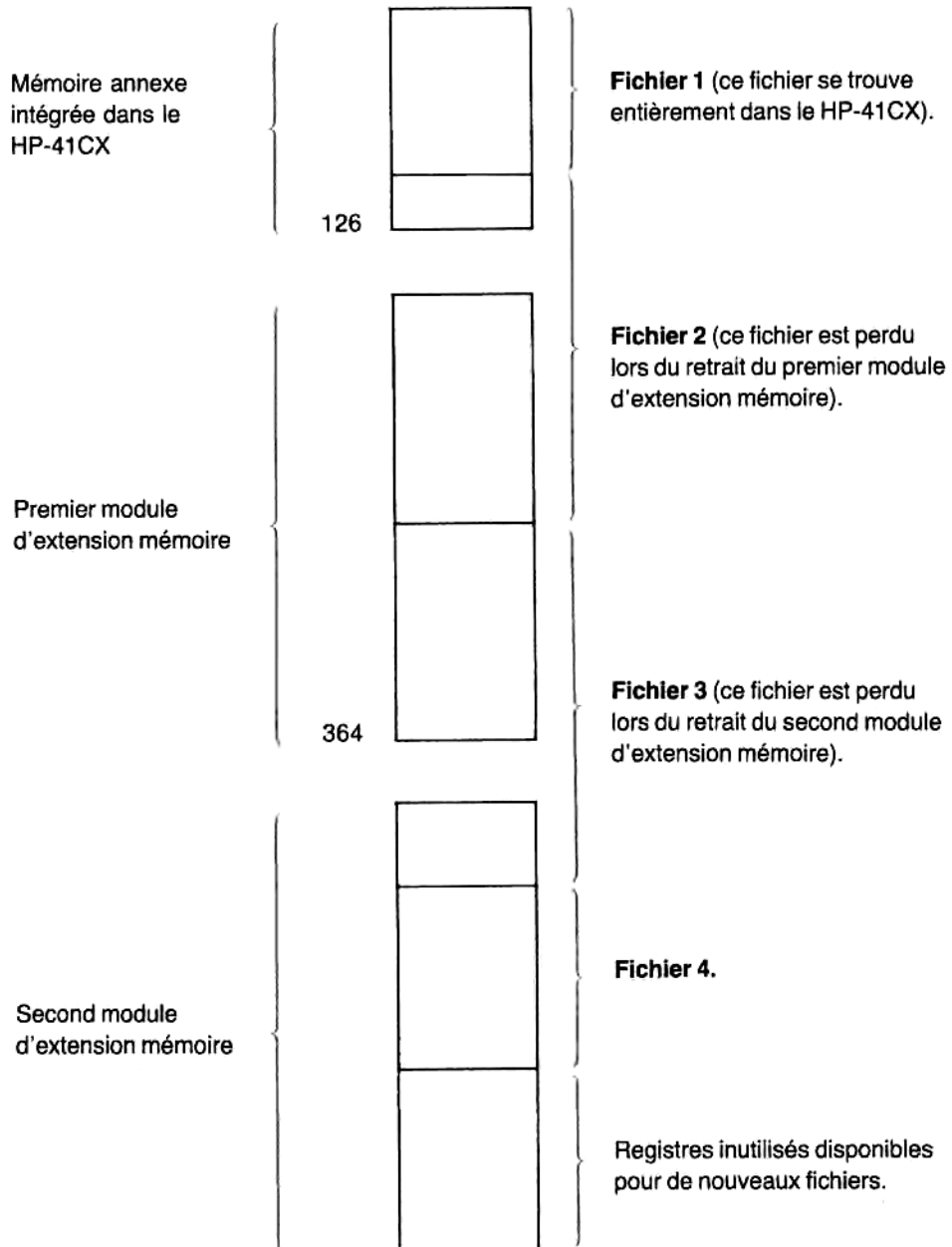
Organisation de la mémoire annexe

Voici une illustration de la mémoire annexe du HP-41CX lorsque vous installez deux modules. Le bloc supérieur représente les 126 registres de mémoire d'extension interne au HP-41CX. Le fichier 1 se trouve entièrement dans le HP-41CX et sera donc conservé même si vous retirez les deux modules.

Le bloc central représente les 238 registres du premier module d'extension mémoire (le premier module installé et utilisé ou le module dans le logement 1 ou 3). Si vous retirez ce module, vous perdez les fichiers 2, 3 et 4.

Le bloc inférieur représente les 238 registres du second module d'extension mémoire (le second module installé ou le module dans le logement 2 ou 4). Si vous retirez ce module, vous perdez les fichiers 3 et 4.

Organisation de la mémoire annexe



Spécifications de l'horloge

Table des matières

Facteur d'exactitude	374
Correction du temps avec le facteur d'exactitude (CORRECT)	375
Rappel, définition et effacement du facteur d'exactitude	376
Calcul du facteur d'exactitude	377
Spécifications	377
Précision	378
Exactitude	378
Chronomètre	378

Facteur d'exactitude

Les fonctions programmables d'ajustement de l'horloge vous permettent de corriger l'horloge et de définir et contrôler le facteur d'exactitude.

De même que la plupart des horloges, l'exactitude et la précision de celle du HP-41CX peuvent être affectées par les variations de l'alimentation, de la température et des procédés de fabrication. Bien que les effets de ces variations soient mineurs, vous pouvez utiliser le facteur d'exactitude pour compenser ces variations, *si elles sont suffisamment régulières*. Le facteur d'exactitude (utilisé avec les fonctions **CORRECT**, **RCLAF** et **SETAF**) sert à compenser les variations à long terme et non pas les conditions anormales (telles que l'exposition à des températures extrêmes pendant plusieurs jours). Dans ce cas il pourra être nécessaire de ré-ajuster l'horloge (**T-X**).

L'horloge du HP-41 est similaire à une montre à quartz typique. Son exactitude, lorsqu'ajustée par le facteur d'exactitude, est du même ordre de grandeur que celle d'une montre à quartz. L'exactitude *sans ajustement* n'est pas aussi bonne que celle d'une montre à quartz mais reste supérieure à celle d'une montre à ressort.

Le facteur d'exactitude est l'intervalle (en secondes) après lequel le HP-41CX ajoute ou retire une impulsion (d'environ $9,8 \times 10^{-5}$ secondes) à la base de temps de 10240 Hz de l'horloge. Le tableau ci-contre indique le format et les limites du facteur d'exactitude.

Un facteur d'exactitude de $-10,5$ soustrait une impulsion toutes les 10,5 secondes. Un facteur d'exactitude de $0,1$ ajoute un battement toutes les 0,1 secondes.

Le facteur de précision

Facteur de précision	Effet	
±99,9 s ⋮ ± 0,1	Ajoute/retire une impulsion chaque <i>n</i> secondes.	Augmente la fréquence des corrections.
0,0	Défaut	Pas de correction

Vous pouvez déterminer le facteur d'exactitude par calcul ou automatiquement avec la fonction `CORRECT`.

Pour plus d'informations sur l'exactitude et la précision de l'horloge du HP-41, consultez le paragraphe « Précision et exactitude ».

Correction du temps avec le facteur d'exactitude (`CORRECT`)

La fonction `CORRECT` met l'horloge à l'heure que vous spécifiez et ajuste automatiquement le facteur d'exactitude. Lorsque vous exécutez `CORRECT` avec une valeur de format **HH.MMSShh** dans le registre X :

- Le système met l'horloge à l'heure spécifiée de la même façon que `SETIME`.
- Le système ajuste automatiquement le facteur d'exactitude en utilisant un calcul interne basé sur la dérive* et l'intervalle de temps depuis la dernière exécution de `SETIME`, `SETDATE`, `SETAF` ou `CORRECT`. Le système ajuste alors automatiquement et continuellement la base de temps de l'horloge selon la nouvelle valeur du facteur d'exactitude.

Lorsque vous exécutez `CORRECT` au clavier, la précision de l'opération de mise à l'heure dépend de l'exécution de la séquence de touches. L'exécution a lieu lorsque vous relâchez la touche qui exécute `CORRECT` **. L'intervalle de temps entre la dernière exécution de `SETIME`, `SETDATE`, `SETAF` ou `CORRECT` et l'exécution courante de `CORRECT` doit être suffisant de façon à rendre insignifiante l'erreur de précision de la pression de touche. Dans la plupart des cas, cet intervalle doit être au minimum de 30 heures. Un intervalle supérieur augmente la probabilité d'une valeur plus faible pour le facteur d'exactitude***.

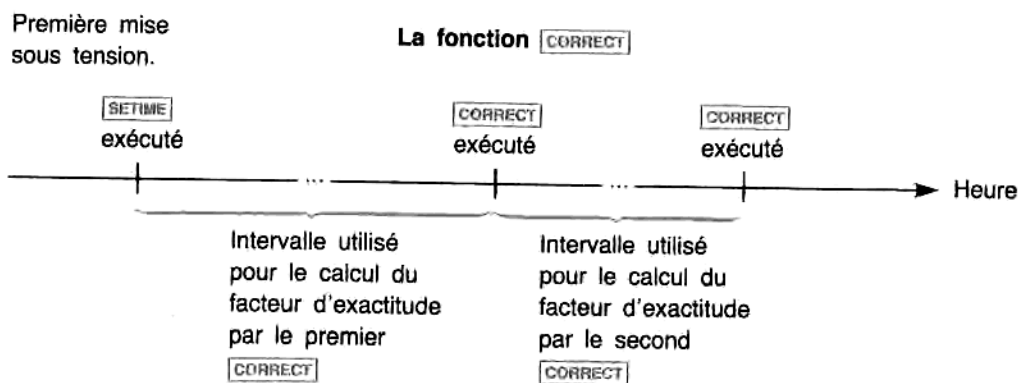
* La dérive est la déviation par rapport à l'heure exacte, due aux variations de l'alimentation, de la température et du matériel. La valeur utilisée par le module horloge pour la dérive est la différence entre l'heure actuelle de l'horloge et la nouvelle heure de l'horloge (spécifiée dans le registre X) au moment où vous exécutez la fonction `CORRECT`.

** Pour la plupart des utilisateurs, la précision maximale de la cadence de frappe est de $\pm 0,1$ seconde. Vous pouvez réduire l'erreur de précision en exécutant `CORRECT` sous la forme d'une fonction affectée à une touche au lieu de `XEQ ALPHA CORRECT ALPHA`. Ceci est dû au fait que le calculateur prend moins de temps pour localiser et exécuter une fonction affectée à une seule touche.

*** Plus longtemps vous attendez pour exécuter `CORRECT` et plus faible sera l'erreur due aux variations de la cadence de frappe par rapport à l'erreur résultant de la combinaison de toutes les causes d'erreur. En pratique, pour la plupart des applications, on estime cette durée à 1 semaine.

Note : La fonction `CORRECT` utilise la dérive calculée pour déterminer le facteur d'exactitude. De ce fait, si vous voulez utiliser `CORRECT` pour améliorer l'exactitude à long terme de l'horloge, vous ne devez pas utiliser `T+X` pour supprimer les erreurs dues à une dérive normale car la modification ne serait pas détectée par la fonction `CORRECT`. L'utilisation de `T+X` pour corriger des erreurs dues à une dérive normale peut donc diminuer la fiabilité du facteur d'exactitude.

L'ajustement du facteur d'exactitude effectué par `CORRECT` dépend de la différence entre l'heure courante et la nouvelle heure au moment où vous exécutez `CORRECT`. Si vous n'avez pas préalablement défini d'heure avec `SETIME`, l'exécution de `CORRECT` peut donner un facteur d'exactitude incorrect. Néanmoins, une fois l'heure définie par `SETIME`, vous pouvez utiliser `CORRECT` aussi souvent que nécessaire.



Souvenez-vous que l'augmentation des intervalles entre les exécutions de `SETIME` et de `CORRECT` ou de deux `CORRECT` augmente la précision du facteur d'exactitude.

Rappel, définition et effacement du facteur d'exactitude

La fonction `RCLAF` rappelle dans le registre X la valeur du facteur d'exactitude. Les contenus de la pile montent d'un registre comme lors du rappel du contenu d'un registre de données.

La fonction `SETAF` définit le facteur d'exactitude égal au contenu du registre X. Le facteur d'exactitude représente un intervalle de temps mesuré en secondes (voir page 375). Le HP-41 arrondit le facteur d'exactitude au dixième de seconde le plus proche (**SS.t**) ou le définit égal à zéro selon les conditions suivantes :

- Le facteur d'exactitude vaut 0 si $x = 0$ ou $x \geq |99,95|$.
- Le facteur d'exactitude vaut 0,1 si $0 < x \leq 0,1$ ($x \neq 0$) et $-0,1$ si $-0,1 \leq x < 0$.
- Si le contenu du registre X (x) est dans l'intervalle suivant: $0,1 < x < 99,949$ ou $-99,95 < x < -0,1$, le HP-41 arrondit le facteur d'exactitude au format **±SS.t**.

Lorsque vous exécutez `SETAF`, le HP-41 modifie automatiquement et continuellement l'heure de l'horloge en fonction du facteur d'exactitude spécifié.

Pour effacer le facteur d'exactitude, placez 0 dans le registre X et exécutez **[SETAF]**.

Si vous savez que l'alimentation du HP-41 va être interrompue (lors du retrait de la batterie par exemple), il est judicieux de rappeler le facteur d'exactitude et de le noter de façon à pouvoir le ré-utiliser lorsque vous remettez le HP-41 sous tension.

Calcul du facteur d'exactitude

La fonction **[CORRECT]** vous permet de corriger aisément la base de temps de l'horloge (grâce au calcul automatiquement de l'erreur moyenne accumulée). Néanmoins, si vous désirez calculer un facteur d'exactitude sur un intervalle de temps relativement court (tel que 36 heures), toute erreur de pression de touche ayant lieu lors de l'exécution de **[CORRECT]** peut avoir une influence considérablement supérieure à celle existante après des intervalles supérieurs. En calculant vous-même le facteur d'exactitude puis en l'entrant à l'aide de **[SETAF]**, vous pouvez obtenir un facteur d'exactitude sur des intervalles courts plus précis que ceux obtenus par **[CORRECT]**. De plus, si vous modifiez la dérive en utilisant **[T+X]**, le facteur d'exactitude résultant des exécutions ultérieures de **[CORRECT]** sera probablement faux. Lorsque vous modifiez la dérive à l'aide de **[T+X]**, la meilleure méthode pour déterminer le facteur d'exactitude est votre propre calcul en suivant la formule ci-après :

Formule de calcul du facteur d'exactitude :

$$F.E = \frac{1}{\frac{1}{FEI} - \frac{10240}{86400} ERR_{spj}}$$

où :

- FEI = facteur d'exactitude initial
(Si FEI est nul, substituez « 0 » à $1/FEI$).
- ERR_{spj} = erreur courante en secondes par jour.
(Une horloge « lente » a une erreur négative et une horloge « rapide » a une erreur positive).
- 10240 = impulsions par seconde de la base de temps de l'horloge.
- 86400 = nombre de secondes dans une journée.

Une fois le facteur d'exactitude calculé, vous devez l'arrondir à une décimale puis l'entrer dans le HP-41 avec la fonction **[SETAF]**.

Spécifications

Précision. Vous pouvez obtenir une précision d'environ 0,1 seconde lors de la mise à l'heure de l'horloge au clavier. Néanmoins, la mise à l'heure peut être moins précise selon le temps de réponse de l'utilisateur. Vous pouvez ajuster l'heure courante avec une précision maximale de 0,01 seconde en utilisant la fonction $\boxed{T+X}$.

Exactitude. L'exactitude de l'horloge dépend d'une base de temps à quartz stabilisée. De même que pour toute horloge basée sur un quartz, la stabilité à un moment donné dépend des conditions d'environnement (variations de température et de tension d'alimentation). Si vous utilisez votre HP-41 dans un environnement stable, l'inexactitude totale peut être rendue négligeable grâce à l'utilisation du facteur d'exactitude. L'exactitude de l'horloge du HP-41 à 25 °C est $\pm 3,02$ secondes par jour (± 35 ppm), le vieillissement ne modifie pas l'exactitude de l'horloge de plus de $\pm 1,30$ seconde par jour (± 15 ppm).

Chronomètre. La précision du chronomètre et la différence entre les temps intermédiaires est de 0,01 seconde. Pour obtenir une exactitude optimale, les temps intermédiaires doivent être pris à des intervalles supérieurs à 0,08 seconde; sinon le résultat peut être affecté par une erreur due à la durée de traitement de l'opération. En mode chronomètre, une pression trop rapide des touches peut provoquer une suppression temporaire de tout ou partie de l'affichage, mais n'a pas d'effet sur le fonctionnement du chronomètre.

Piles, garantie et maintenance

Table des matières

Logements d'entrée-sortie	381
Batterie et alimentation	381
Consommation	381
Consommation des périphériques	382
Effacement de la mémoire, interruption de l'alimentation et baisse de tension	382
Indicateur de baisse de charge de la batterie	383
Remplacement de la batterie	383
Vérification du fonctionnement	385
Garantie	386
Modifications	386
Informations	387
Maintenance	388
Maintenance en Europe	388
Dans les autres pays	389
Coût de la maintenance	389
Garantie sur les réparations	389
Instructions d'expédition	389
Conditions d'environnement	390

Logements d'entrée-sortie

Veillez à remettre le capuchon sur tous les logements inutilisés de façon à protéger ces derniers contre la poussière.

ATTENTION

Ne placez pas vos doigts, des outils ou tout autre objet dans les logements des modules. Ceci peut ré-initialiser la mémoire permanente et endommager le logement ou le HP-41.

Batterie et alimentation

Le HP-41 est alimenté par quatre piles. Celles fournies avec le HP-41 sont de type alcaline et peuvent durer six mois ou plus selon l'utilisation. Vous pouvez néanmoins les remplacer par des piles d'accumulateurs rechargeables.

La durée de vie d'une batterie dépend du type d'opérations effectuées et du nombre de périphériques utilisés. Sans périphériques, un jeu de quatre piles alcalines neuves donne entre 45 et 85 heures d'exécution *continue* de programme (cf. « Consommation maximale »). L'affichage seul sans opération consomme très peu de courant.

La durée de vie réelle des piles dépend du degré d'utilisation du HP-41 et de ses périphériques, de l'utilisation de programmes ou de calculs au clavier et du type de fonctions utilisées*. Les opérations utilisant le plus de courant sont, dans l'ordre : l'utilisation de périphériques, l'exécution de programmes, l'affichage/utilisation du chronomètre, l'utilisation des catalogues et l'affichage de l'horloge. Les catalogues 4, 5 et 6 utilisent autant de courant que l'exécution d'un programme, même lorsqu'ils sont arrêtés (c'est la raison du temps plus court avant l'extinction automatique).

Si le HP-41 reste éteint, un jeu de piles alcaline neuves conservera le contenu de la mémoire permanente aussi longtemps que ces piles dureraient en dehors du HP-41 (environ un an et demi).

Consommation

L'horloge du HP-41 consomme en permanence une certaine puissance de la batterie. La consommation à un moment donné dépend du type d'utilisation du HP-41. Il y a trois modes principaux de consommation :

- **Consommation élevée** (5 à 20 mA). Se produit lors de l'utilisation de la fonction **[ALMCAT]** (catalogue des alarmes) ou lorsque le mode chronomètre est actif. Ce mode correspond en outre à l'exécution d'un programme, d'un catalogue ou de l'éditeur.

- **Consommation moyenne** (0.5 à 2.0 mA). Se produit lors de l'affichage de l'horloge (**CLKT**). Ce mode correspond en outre au HP-41 allumé et attendant une entrée au clavier. L'affichage de l'heure (**CLKT**) utilise deux fois plus de courant que celui de l'heure et la date (**CLKTD**). Avec **CLKT** le HP-41 met l'affichage à jour plus souvent.
- **Consommation faible** (0.01 à 0.05 mA). Se produit lorsque le HP-41 est éteint. La base de temps de l'horloge continue à fonctionner ainsi que le chronomètre s'il était déclenché.

Une batterie HP 82120A complètement rechargée a une capacité de 65 mAH (milliampère-heure). Une pile alcaline neuve fournit environ 500 mAH.

Consommation des périphériques

Lorsque vous utilisez des périphériques qui s'alimentent sur les piles du HP-41 (tels que le lecteur de carte et le lecteur optique), la durée de vie de la batterie diminue considérablement. Si vous voulez utiliser ces périphériques fréquemment, nous vous conseillons d'alimenter le HP-41 par une batterie d'accumulateurs rechargeables HP 82120A.

Effacement de la mémoire, interruption de l'alimentation et baisse de tension

Effacement de la mémoire. La réinitialisation de la mémoire (**↔**/**ON**) *n'affecte pas* :

- l'horloge et les formats associés (**CLK12** - **CLK24** et **CLKT** - **CLKTD**).
- le chronomètre.

La réinitialisation de la mémoire *efface* :

- les mémoires principale et annexe,
- toutes les alarmes.
- toutes les affectations de fonctions utilisateur.
- tous les indicateurs prennent leurs états par défaut. Voir chapitre 19.

Voir chapitre 1 pour d'autres renseignements concernant l'effacement et la réinitialisation de la mémoire.

Interruption temporaire de l'alimentation. Une interruption de l'alimentation (y compris le retrait de la batterie) de durée suffisante peut remettre le HP-41 à l'état initial. Toutes les informations sont effacées : mémoire permanente, horloge et alarmes. L'heure et la date prennent alors les valeurs par défaut suivantes : midi le 1^{er} janvier 1900. Ces interruptions peuvent introduire d'autres erreurs dans le fonctionnement de l'horloge. Nous vous recommandons donc de vérifier l'heure et la date après une interruption de l'alimentation. Si elles ne sont pas correctes, suivez la procédure ci-dessous :

1. Remettez le HP-41 à l'heure et à la date (et).
2. Redéfinissez les formats d'affichage s'ils sont incorrects (ou , ou).
3. Initialisez le chronomètre et arrêtez-le s'il est déclenché (0).

Baisse de tension. Lorsque la batterie est trop faible pour afficher l'horloge, l'exécution de ou de éteint le HP-41 (cependant, l'horloge continuera à fonctionner). Dans la plupart des cas, ceci ne se produira que si le témoin **BAT** est allumé.

Indicateur de baisse de charge de la batterie

Le témoin **BAT** apparaît à l'affichage en cas de baisse de charge de la batterie. Si vous utilisez un périphérique, son retrait (après avoir éteint le HP-41 et le périphérique) augmente la durée de vie de la batterie.

Muni d'une batterie alcaline et sans périphérique :

- le HP-41 reste opérationnel durant 2 à 7 heures d'exécution continue de programme après l'apparition du témoin **BAT***.
- le HP-41 hors tension conserve le contenu de sa mémoire pendant environ un mois après l'apparition du témoin **BAT**.

Remplacement de la batterie

La batterie fournie avec le HP-41 ainsi que celles listées ci-dessous ne sont *pas* rechargeables.

AVERTISSEMENT

Ne pas tenter de recharger une batterie de piles alcalines, ne pas stocker les batteries près d'une source de chaleur, ne pas les jeter au feu, elles pourraient fondre ou exploser.

* Remarquez que cette durée est le temps minimum disponible pour une exécution continue. Si vous utilisez l'ordinateur en mode manuel, cette durée est considérablement augmentée.

Nous recommandons les piles alcalines suivantes pour le HP-41 :

Eveready E90
National AM5(s)

Mallory MN9100
Panasonic AM5(s)

VARTA 7245

Le HP-41 conserve le contenu de sa mémoire pendant un court instant lorsque vous retirez la batterie (si vous avez éteint le HP-41 avant d'enlever la batterie). Vous avez ainsi le temps de remplacer la batterie sans perdre le contenu de la mémoire. Si vous ne remplacez pas la batterie suffisamment vite, le contenu de la mémoire peut être perdu.

Utilisez la procédure suivante pour installer une nouvelle batterie :

1. Éteindre le HP-41.
2. Tenez le HP-41 comme indiqué ci-contre et faites sortir le support-batterie en poussant vers le haut.



3. Retirez les piles du support.



ATTENTION

Lorsque vous remplacez la batterie du HP-41, remplacez les *quatre* piles par des nouvelles. Une vieille pile peut couler. Veillez en outre à ne pas insérer les piles à l'envers, ceci effacerait le contenu de la mémoire permanente.

- Placez les nouvelles piles dans le sens indiqué par les symboles sur le support. Si une des piles est installée à l'envers, vous ne pourrez pas mettre le HP-41 sous tension.



- Insérez le support dans le HP-41 de façon que les extrémités libres des piles soient dirigées vers les logements d'entrée-sortie.
- Poussez le haut du support aussi loin que vous pouvez dans le HP-41 et appuyez sur le bas pour fixer le support.



- Allumez le HP-41. Si la mémoire a été réinitialisée (son contenu est perdu), l'affichage contient le message **MEMORY LOST**. La pression d'une touche quelconque efface ce message.

Vérification du fonctionnement

Si vous suspectez un mauvais fonctionnement de votre HP-41, suivez la procédure ci-dessous :

- Vérifiez que toutes les piles sont placées dans le bon sens et que les contacts ne sont pas sales.
- Si le HP-41 ne répond pas aux pressions de touches, appuyez sur les touches **ON** et **ENTER** simultanément, puis relâchez-les. Rallumez l'ordinateur, si nécessaire, et testez une séquence de touches.
- Si le HP-41 ne répond toujours pas, retirez et ré-installez les piles.

Si l'ordinateur ne s'allume pas, installez de nouvelles piles.

Si cela ne suffit pas, retirez la batterie et laissez le HP-41 se décharger pendant toute une nuit. Lorsque vous ré-installez la batterie et que vous mettez l'ordinateur sous tension, l'ordinateur et la mémoire doivent être vides et le HP-41 doit afficher **MEMORY LOST**.

4. Si le HP-41 ne répond toujours pas aux pressions de touches, retirez la batterie et court-circuitez les contacts dans le HP-41 (*un contact momentané suffit*) puis ré-installez les piles. Le contenu de la mémoire est perdu et il peut être nécessaire d'appuyer plusieurs fois sur **ON** pour rallumer le HP-41.
5. Si le HP-41 ne s'allume pas, il doit être envoyé à l'un de nos centres de maintenance.

Garantie

Le HP-41 est garanti par Hewlett-Packard contre tout vice de matière et de fabrication pour une durée d'un an à partir de la date de livraison. Hewlett-Packard s'engage à réparer ou, éventuellement, à remplacer les pièces qui se révéleraient défectueuses pendant la période de garantie. Cette garantie couvre les pièces et la main-d'œuvre*. Elle disparaît en cas d'utilisation en dehors des spécifications, de modification ou de maintenance par un centre non reconnu par Hewlett-Packard.

Seuls les essais effectués à partir des programmes Hewlett-Packard seront considérés comme faisant foi lors de litiges concernant le fonctionnement du matériel. La responsabilité de Hewlett-Packard ne peut être engagée dans le cas d'une application particulière. La société ne peut pas être tenue pour responsable des dommages indirects.

Si vous revendez ou offrez ce matériel, la garantie est automatiquement transférée pour la durée initiale d'un an.

Modifications

Les appareils vous sont livrés selon les spécifications en vigueur au moment de la fabrication. Hewlett-Packard n'est pas tenu de modifier les appareils déjà vendus.

* Valable en France seulement.

Lorsque l'acheteur est non professionnel ou consommateur au sens de la loi 78-23 du 10 janvier 1978, les obligations de HP définies ci-dessus ne sont pas exclusives de la garantie légale en matière de vices cachés (Article 1641 et suivants du Code Civil).

Informations

Pour toute question concernant les termes de cette garantie, veuillez contacter :

- **France**

Hewlett-Packard France
Division Informatique Personnelle
Parc d'activités du Bois-Briard
Avenue du Lac
91040 Evry Cedex

- **Europe**

Hewlett-Packard S.A.
150, route du Nant-d'Avril
P.O. Box
CH-1217 Meyrin 2
Genève
Suisse
Tél. : (022) 83 81 11

Note : N'envoyez pas votre calculateur à cette adresse.

- **Autres pays**

Hewlett-Packard Intercontinental
3495 Deer Creek Road
Palo Alto, California 94304
U.S.A.
Tél. : (415) 857.1501

Note : N'envoyez pas votre calculateur à cette adresse.

Maintenance

Les appareils sont généralement réparés et ré-expédiés dans un délai de cinq jours ouvrables à dater de leur réception. Il s'agit d'un délai moyen pouvant varier selon l'époque de l'année et la charge de travail du service après-vente.

Aucun contrat de maintenance n'est prévu. Les schémas et circuits sont la propriété de Hewlett-Packard et ne peuvent être ni diffusés ni commercialisés.

Maintenance en Europe

Si votre calculateur doit être réparé, adressez-vous à un revendeur officiel Hewlett-Packard qui nous le fera parvenir ou envoyez-le à l'une des adresses suivantes :

FRANCE

HEWLETT-PACKARD FRANCE
S.A.V. Calculateurs de Poche
Division Informatique Personnelle
F-91947 Les Ulis Cedex
Tél. : (6) 907.78.25

AUTRICHE ET PAYS DE L'EST

HEWLETT-PACKARD GmbH
Wagramestr.-Liebegasse
A - 1220 Vienne
Tél. : (0222) 23.65.11

BELGIQUE

HEWLETT-PACKARD BELGIUM SA/NV
Boulevard de la Woluwe, 100
B - 1200 Bruxelles
Tél. : (2) 762.32.00

DANEMARK

HEWLETT-PACKARD A/S
Datavej 52
Dk - 3460 Birkerød (Copenhague)
Tél. : (02) 81.66.40

HOLLANDE

HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
ND-1181 KK Amstelveen (Amsterdam)
P.O. Box 667
Tél. : (020) 472021

NORVÈGE

HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterdalen 18
N-1345 Oesteraas (Oslo)
Tél. : (2) 17.11.80

FINLANDE

HEWLETT-PACKARD OY
Revontulentie 7
02100 Espoo 10 (Helsinki)
Tél. : (90) 455.02.11

ESPAGNE

HEWLETT-PACKARD ESPAÑA S.A.
Calle Jerez 3
E-Madrid 16
Tél. : (1) 458.2600

SUÈDE

HEWLETT-PACKARD SVERIGE AB
Skalholtsgratan 9, Kista
Box 19
16393 Spanga (Stockholm)
Tél. : (08) 750 2000

ALLEMAGNE

HEWLETT-PACKARD GmbH
Vertriebszentrale
Bernner Strasse 117
Postfach 560 140
D-6000 Frankfurt 56 (Francfort)
Tél. : (611) 50041

ITALIE

HEWLETT-PACKARD (ITALIANA)
S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 Cernusco Sul Naviglio
(Milan)
Tél. : (2) 90.36.91

SUISSE

HEWLETT-PACKARD (SCHWEIZ) AG
Kleinrechner-Service
Allmend 2
CH-8967 Widnau
Tél. : (057) 31 21 11

ROYAUME-UNI

HEWLETT-PACKARD Ltd
King Street Lane
Winnersh, Wokingham
GB-Berkshire RG11 5AR
Tél. : (734) 784774

Dans les autres pays

Tous les centres de maintenance Hewlett-Packard ne sont pas équipés pour assurer la maintenance des calculateurs. Cependant, si vous avez acheté votre calculateur chez un revendeur agréé HP, vous pouvez être sûr que HP dispose d'un centre de maintenance dans ce pays.

En dehors de ces pays, vous pouvez contacter le bureau commercial HP le plus proche pour plus d'informations et, si l'appareil ne peut y être réparé, veuillez l'envoyer au centre de maintenance le plus proche.

Les frais d'expédition et éventuellement de douane sont à votre charge.

Coût de la maintenance

Les réparations hors garantie sont effectuées pour un prix forfaitaire incluant pièces et main-d'œuvre. Ce forfait est sujet à la TVA en France ou taxes similaires dans les autres pays. Ces taxes apparaissent en détails sur les factures.

Les calculateurs endommagés par accident ou utilisation hors des spécifications ne sont pas couverts par le forfait. Le prix de la réparation est alors fonction des pièces changées et du temps passé.

Garantie sur les réparations

Tout appareil réparé par Hewlett-Packard est garanti, pièces et main-d'œuvre, pendant 90 jours à compter de la date de réparation.

Instructions d'expédition

Si vous devez nous renvoyer votre calculateur pour réparation, conformez-vous aux indications suivantes :

- Joignez au calculateur la carte de maintenance portant la description de la panne.
- Si l'appareil est sous garantie, joignez une copie de la facture ou une preuve de la date d'achat.
- Expédiez le calculateur et les différents documents dans la boîte d'origine ou dans un autre emballage de protection pour éviter toute détérioration en cours de transport (ces dommages ne seraient pas couverts par la garantie). Nous vous conseillons d'assurer le colis. Celui-ci doit être expédié au centre de maintenance HP le plus proche.
- Que le calculateur soit sous garantie ou non, les frais d'expédition et éventuellement de douane sont à votre charge. Le retour est effectué port payé.

Conditions d'environnement

Respectez les limites suivantes pour l'environnement de votre HP-41.

- Température de fonctionnement : 0° à 45°C.
- Température de stockage : 0° à 45°C.

*Si vous retirez la batterie ou si vous acceptez une diminution de la précision de l'horloge, l'intervalle des températures de stockage du HP-41 est** :

– 20° à 60°C.

* Si le fonctionnement de l'horloge est affecté par une exposition *temporaire* à des températures extrêmes, ne la remettez *pas* à l'heure en utilisant le facteur d'exactitude mais utilisez la fonction [T+X] (cette fonction remet l'horloge à l'heure sans modifier le facteur d'exactitude). Comme indiqué en annexe F page 374, le facteur d'exactitude compense les effets de conditions relativement constantes et ne doit pas servir à compenser les changements brusques.

Périphériques, extensions et HP-IL

Table des matières

Périphériques du HP-41	392
Lecteur de cartes HP 82104A	392
Imprimante 82143A	392
Lecteur optique HP 82153A	393
Extensions	393
Modules d'extension mémoire HP 82181A	393
Modules d'applications	393
Boucle d'interface Hewlett-Packard (HP-IL) et périphériques	394
Fonctions et nombres XROM	394
Catalogue 2 : le catalogue des fonctions auxiliaires	394
Programmes et fonctions en ROM auxiliaire	395
Comment des fonctions XROM sont affichées comme instructions de programme	395
Copie des nombres XROM	397

L'ordinateur de poche HP-41 devient un *contrôleur* d'un système informatique lorsqu'il est connecté aux périphériques et aux extensions HEWLETT-PACKARD. De plus, le module HP-IL (boucle d'interface HP) relie l'ordinateur à 30 périphériques (maximum) en série formant ainsi un circuit de communications. Quatre logements d'entrée-sortie permettent l'enfichage de modules d'extensions — un par logement —. (Le module HP-IL est enfichable dans un logement alors que chacun des périphériques HP-IL est relié par câble au module ou à d'autres périphériques HP-IL).

Périphériques du HP-41

Lecteur de cartes HP 82104A

Le lecteur de cartes vous permet de conserver les programmes, les registres de données et les affectations de touches du HP-41 sur des cartes magnétiques. De même, programmes, données et affectations stockés sur cartes magnétiques peuvent être chargés en mémoire principale d'un HP-41 à l'aide du lecteur de cartes.

Le lecteur de carte permet l'enregistrement et le stockage immédiats des informations (sans aucune instruction !). Tous les programmes personnalisés sont sur cartes magnétiques. De plus, vous pouvez utiliser sur le lecteur du HP-41 les cartes des HP-67 et HP-97 ; elles sont traduites automatiquement en code HP-41.

Imprimante HP 82143A

Le HP 82143A permet l'impression silencieuse des instructions et des programmes sur un papier thermosensible d'une largeur de 24 caractères. Elle imprime des caractères numériques et alphabétiques, majuscules et minuscules, ainsi que des caractères en double largeur. Elle dispose en outre de plusieurs modes d'impression ; vous pouvez par exemple vérifier des programmes ou des longs calculs.

Lecteur optique HP 82153A

Le lecteur optique lit des programmes codés sous forme de codes-barres HP et les stocke en mémoire principale du HP-41, ce qui est beaucoup plus rapide et beaucoup plus fiable que la saisie au clavier. Les données et les fonctions peuvent également être chargées à partir de codes-barres dans votre HP-41. Tous les programmes personnalisés et les livrets d'application HP concernant le HP-41 sont livrés avec leur version codes-barres.

Extensions

Modules d'extension mémoire HP 82181A (mémoire annexe)

Un module d'extension mémoire offre 238 registres de données supplémentaires (1666 octets) de mémoire annexe au HP-41 (le HP-41CX est livré avec 124 registres de mémoire annexe). Vous pouvez ajouter un ou deux modules d'extension mémoire. Pour plus de renseignements, reportez-vous à l'annexe E.

Livrets d'applications

Les livrets d'applications sont des logiciels d'application écrits par Hewlett-Packard destinés à résoudre des problèmes dans des domaines bien spécifiques tels que l'analyse de circuits ou les décisions financières. Vous pouvez enficher quatre modules d'application au maximum. Les programmes et les fonctions contenus dans chaque module sont listés par catalog 2.

Boucle d'interface Hewlett-Packard (HP-IL) et périphériques

L'enfichage du module HP-IL HP 82160A dans l'un des quatre logements du HP-41 permet de créer une boucle d'interface pouvant gérer jusqu'à 30 périphériques compatibles HP-IL. Le HP-41 est le contrôleur de la boucle, surveillant et régissant l'activité des périphériques. Grâce à ses fonctions de temps, le HP-41CX peut gérer les acquisitions de données par des périphériques HP-IL et générer une sortie automatique des résultats. Le module HP-IL possède les fonctions nécessaires pour contrôler des périphériques d'impression ou de stockage de masse.

Il existe de nombreux périphériques HP-IL pour le stockage de masse, l'affichage vidéo, l'impression, le traçage et la mesure. De plus, le module entrées-sorties étendues HP 82183A permet l'extension des fonctions du module HP-IL pour le contrôle de périphérique d'E-S, et le module traceur HP 82184A offre des fonctions de traçage évoluées (dont la génération de codes-barres).

Pour plus de renseignements, contactez le bureau commercial HP le plus proche.

Fonctions et nombres XROM

Toute fonction ou tout programme accessible à l'utilisateur à l'aide d'un périphérique ou d'une extension du HP-41 est considéré comme une fonction « ROM auxiliaire » (*XROM*). Le catalogue 2 (voir ci-dessous) donne une liste des périphériques auxiliaires. Il liste également toutes les fonctions de chacun de ces périphériques. Chaque fonction ROM auxiliaire est enregistrée sous un numéro XROM composé de 2 parties.

Catalogue 2 : Le catalogue des fonctions auxiliaires

Le catalogue 2 (voir chapitre 9) est un listage complet des fonctions et des programmes XROM de chaque périphérique.* En ce qui concerne le HP-41CX, le catalogue 2 donne uniquement les noms des périphériques source (les titres ROM) jusqu'à ce que vous arrêtiez le catalogue et appuyiez sur **ENTER ↑**. Ce qui lance le listage des fonctions/programmes spécifiques au dernier périphérique dont le nom figurait à l'affichage. Pour revenir au listage des périphériques source, arrêtez à nouveau le catalogue et appuyez sur **ENTER ↑**.

Les touches **SST** et **MSY** fonctionnent comme pour les autres catalogues. Ces touches ne permettent pas l'accès à la liste de titres et à la liste de fonctions (seule **ENTER ↑** le permet). Lorsque la liste des fonctions spécifiques à un périphérique arrive à sa fin, le listage continue sur la prochaine liste de fonctions concernant un autre périphérique.

* Certaines fonctions auxiliaires, comme les fonctions d'horloge et les fonctions d'extension mémoire, figurent dans le catalogue 2. Voir annexe I.

Programmes et fonctions en ROM auxiliaire

Une opération en ROM dans un module d'application ou d'extension ou dans un périphérique se présente sous la forme d'un *programme* ou d'une *fonction*. Un programme est copié en mémoire utilisateur, puis listé et modifié. Une fonction par contre, ne peut être visualisée ; elle est seulement utilisée. Lors du listage du catalogue 2, l'ordinateur distingue les programmes des fonctions à l'aide d'un T surélevé figurant devant les noms des programmes.

SECUR 1B	←	Titre ROM (identification du périphérique)
TBONDS		
TSTOCK	←	Programme
⋮		
TATP		
JDAY	←	Fonction
TBEP		
⋮		

Comment les fonctions XROM sont affichées comme des instructions de programme

Lorsqu'une fonction auxiliaire est écrite dans une instruction de programme, l'affichage de cette instruction est soumis à deux conditions : si le module contenant cette fonction est enfiché ou non dans le HP-41 et si cette fonction XROM se présente comme programme ou comme fonction.

Le numéro XROM identifie une fonction XROM par son périphérique (numéro d'identification de ROM) et par l'adresse de celui-ci (numéro de fonction).

Si un module indispensable n'est pas enfiché, le HP-41 n'a aucune connaissance des fonctions XROM de ce module — *à moins* qu'une fonction ait été affectée à une touche utilisateur, auquel cas son numéro XROM est connu parce qu'il était affecté à cette touche. De même, si un module est enlevé *après* qu'une de ses fonctions soit entrée dans un programme, l'ordinateur identifie la fonction « manquante » par son numéro XROM.

Par conséquent :

- Si l'ordinateur a accès à une fonction XROM, elle sera entrée dans une ligne de programme
 soit comme *label* pour une fonction auxiliaire,
 soit comme **XROM^T label** pour un programme auxiliaire.

Il en est de même dans le cas où une touche utilisateur est utilisée pour entrer une instruction de programme.

- **XROM** *nombre, nombre* remplace l'affichage *label* ou **XROM^T label** d'une instruction de programme lorsque le module correspondant est retiré. Le numéro XROM reste affiché aussi longtemps que le module n'est pas enfiché ; c'est-à-dire que l'affichage d'origine est restitué lorsque le module est à nouveau dans son logement. C'est ce que l'on obtient également si une touche est utilisée avec le module correspondant non enfiché.*
- Si le module correspondant n'est pas enfiché et que vous faites une exécution Alpha d'une fonction XROM pour une ligne de programme, la ligne de programme sera

XEQ^T label tout comme un rappel de programme en mémoire principale.

Lorsque le module est par la suite enfiché, la ligne de programme *ne change pas* :

XEQ^T label

Affichage d'une instruction de programme

A. Si le module correspondant est enfiché ou une touche utilisateur est utilisée

XFROM^T label (programme)

ou

label (fonction)

(Cette instruction de programme utilise 2 octets de mémoire.)

B. Si le module correspondant n'est pas enfiché et qu'une touche utilisateur n'est pas utilisée :

(Cette instruction de programme utilise 2 octets plus 1 octet par caractère dans le label.)

* Une fonction auxiliaire ne peut être affectée à une touche utilisateur que lorsque le module contenant cette fonction est connecté au HP-41. Sinon, on obtient le message d'erreur NONEXISTENT

Temps d'exécution. Bien que l'instruction `XEQT label` — entrée alors que le module est non enfiché — soit opérationnelle pour l'exécution de fonctions ou de programmes auxiliaires spécifiques lorsque le module est ré-enfiché (cas B page 396), elle n'est pas vraiment équivalente à `XROMT label` ou `label` (cas A page 396). Le cas B est moins efficace et le temps d'exécution est plus long, ceci parce qu'un appel XROM (avec simplement `label` pour une fonction XROM) va rechercher directement cette fonction XROM dans le catalogue 2. Par contre, une instruction `XEQT label` va *d'abord* dans le catalogue 1, cherchant dans tous les programmes. Si elle ne trouve pas le label en question, elle continue sa recherche dans le catalogue 2.

Espace mémoire. Une instruction `XROMT label` ou `label` (cas A) utilise 2 octets de mémoire, alors qu'une instruction `XEQT label` en utilise 2 plus 1 octet par caractère de label.

Copie de numéros XROM

Tous les modules ROM enfichables ont un numéro d'identification ROM, et certains sont dupliqués. *Ce qui évite d'utiliser simultanément un module ROM avec un numéro d'identification ROM dupliqué.* Bien qu'elles soient des fonctions *auxiliaires* (non intégrées) au HP-41CX, toutes les fonctions d'extension mémoire et les fonctions étendues utilisent 25 comme numéro d'identification XROM, et toutes les fonctions horloge utilisent 26. (Aucune des autres fonctions auxiliaires n'a de numéro XROM.)

Comparaison avec le HP-41C/CV

Table des matières

Généralités	397
Catalogue des nouvelles fonctions	398
Manuel d'utilisation	398
Configuration de la mémoire	398
Fonctionnement des catalogues	399
Catalogues 1, 2 et 3	399
Nouveaux catalogues : 4, 5 et 6	399
Fonctions d'horloge	399
Mémoire annexe et fonctions étendues	400
Mémoire annexe	401
Fonctions étendues	401
Nouvelle terminologie utilisée dans ce manuel	401

Si vous êtes un utilisateur expérimenté du HP-41C/CV vous connaissez déjà le fonctionnement et la plupart des caractéristiques du HP-41CX. Cette annexe résume les différences entre le HP-41CX et le HP-41C/CV.

Les programmes écrits pour le HP-41C/CV (y compris les modules enfichables) sont compatibles avec le HP-41CX. Par contre, ceux écrits pour le HP-41CX ne sont pas forcément compatibles avec le HP-41C/CV.

Généralités

L'ordinateur de poche HP-41CX est basé sur le HP-41CV (lui-même basé sur le HP-41C mais ayant cinq fois plus de mémoire) et comporte en plus toutes les fonctions du module horloge HP 82182A ainsi que les fonctions et l'espace mémoire du module des fonctions d'extension mémoire HP 82180A. Le HP-41CX comporte en outre plusieurs nouvelles alarmes, un chronomètre, un bloc d'extension mémoire et d'autres fonctions en plus de modifications mineures dans le fonctionnement des catalogues. La répartition de la mémoire lors de la réinitialisation de la mémoire permanente est différente (100 registres pour le stockage des données). La suite de cette annexe traite ces différences une par une et donne des références aux présentations de ces caractéristiques dans le manuel.

Catalogue des nouvelles fonctions

Le HP-41CX conserve les catalogues du HP-41C/CV. Le catalogue 2 contient toutes les nouvelles fonctions (non présentes dans le HP-41C/CV standard) et le catalogue 3 contient les fonctions standard (identiques à celles du HP-41C/CV). Ceci permet de maintenir une homogénéité avec les produits précédents : le jeu des fonctions standard est inchangé et les fonctions d'horloge et d'extension mémoire sont dans le catalogue 2, comme lorsque vous connectez les modules correspondant au HP-41C/CV. Toutes les fonctions d'horloge et d'extension mémoire du HP-41CX utilisent par conséquent les numéros XROM (voir annexe H) des modules originaux comme identification. Le numéro d'identification de ROM pour les fonctions d'extension mémoire est 25 et celui des fonctions d'horloge est 26 (veillez à ne pas utiliser de modules enfichables ayant les mêmes numéros d'identification).

Manuel d'utilisation

Nous avons entièrement ré-écrit le manuel d'utilisation pour ce produit. De nombreuses présentations de caractéristiques ont été modifiées et clarifiées. De ce fait, certains termes utilisés dans ce manuel sont différents, particulièrement dans les domaines de programmation, mémoire et alarmes (voir page 401). Nous avons en outre modifié la convention d'impression pour les fonctions secondaires des touches du clavier et pour les fonctions non présentes sur le clavier (voir au dos du manuel).

Configuration de la mémoire

Le tableau ci-dessous illustre la répartition de la mémoire dans les diverses versions du système de calcul HP-41.

Configuration de la mémoire

Appareil	Mémoire principale			Total Mémoire annexe
	Total	Configuration initiale		
		Données	Non affectée*	
HP-41CX	319	100 (R ₀₀ -R ₉₉)	219	124
HP-41CV	319	273 (R ₀₀ -R ₂₇₂)	46	0
HP-41C	63	17 (R ₀₀ -R ₁₆)	46	0

* Les espaces mémoires pour les instructions de programmes, les alarmes et les affectations au clavier proviennent de la mémoire non affectée (voir chapitre 12).

Fonctionnement des catalogues

Lors du listage d'un catalogue, la pression d'une touche quelconque sauf **[R/S]** et **[ON]** accélère le listage (sur le HP-41C/CV ceci ralentissait le listage).

Catalogues 1, 2 et 3

Le listage du catalogue 1 indique maintenant le nombre d'octets pour chaque programme (page 171). Le listage du catalogue 2 donne les fonctions par groupes (voir ci-après) et celui du catalogue 3 est inchangé.

Nouveaux catalogues : 4, 5 et 6

Le HP-41CX possède trois nouveaux catalogues : 4, 5 et 6. Ils utilisent l'énergie de la batterie au même rythme qu'un programme en exécution, même lorsqu'ils sont arrêtés, contrairement aux catalogues 1, 2 et 3. Par conséquent, ils se terminent automatiquement en deux minutes (une seule si la batterie est faible) lorsque vous les arrêtez.

Les nouveaux catalogues répondent en outre à **[SST]** en fin de fichier et à **[EST]** en début de fichier : l'affichage clignote une fois et le listage de catalogue s'arrête (dans le HP-41C/CV, **[SST]** en fin de listage termine la fonction catalogue).

Catalogue 4 : Catalogue de mémoire annexe (page 206). Les fonctions **[CATALOG] 4** et **[EMDIR]** permettent de lister le catalogue de la mémoire d'extension. **[CATALOG] 4** n'est pas programmable alors que **[EMDIR]** l'est. La fonction **[EMDIR]** est presque identique à celle du module de fonctions d'extension mémoire *sauf* qu'elle opère maintenant comme un catalogue : **[R/S]** lance et arrête le listage, **[SST]** et **[BST]** permettent de se déplacer pas à pas dans le listage et une imprimante ne l'accepte qu'en mode Trace.

Catalogue 5 : Catalogue d'alarme (page 255). Les fonctions **[CATALOG] 5** et **[ALMCAT]** permettent de lister toutes les alarmes actives en mémoire. **[CATALOG] 5** n'est pas programmable alors que **[ALMCAT]** l'est. Cette dernière est presque identique à celle du module horloge *sauf* que s'il n'y a pas d'alarme en mémoire, le listage donne **CAT EMPTY**.

Catalogue 6 : Catalogue des affectations de touches (page 168). La fonction **[CATALOG] 6** permet de lister toutes les affectations de touches pour les fonctions et les labels globaux dans l'ordre des codes de touche. La séquence **[C]** permet d'annuler une affectation.

Fonctions d'horloge

La Partie IV de ce manuel (« Description détaillée des fonctions d'horloge ») décrit toutes les fonctions d'horloge du HP-41CX : « Fonctions d'horloge » (15), « Alarmes » (16) et « Fonctionnement du chronomètre » (17). Le HP-41CX comporte en plus de nouvelles fonctions d'alarme, particulièrement pour l'effacement des alarmes et une fonction supplémentaire pour l'utilisation en chronomètre :

- Vous pouvez effacer de la mémoire une alarme message répétitive en appuyant sur **[C]** pour l'accepter pendant que l'alarme s'éteint (page 255).
- Fonctions programmables d'effacement d'alarme : **[CLALMA]** (effacement d'alarme par ALPHA), **[CLALMX]** (effacement d'alarme par X) et **[CLRALMS]** (effacement de toutes les alarmes). (Page 258.)

- La fonction programmable **[RCLALM]** rappelle les paramètres d'une alarme dans les registres de la pile et ALPHA. (Page 252.)
- L'intervalle minimal de répétition pour une alarme est 1 seconde au lieu de 10 secondes. (Page 250.)
- Vous pouvez activer le chronomètre et en positionner les marqueurs avec la fonction programmable **[SWPT]** (page 274).

Les fonctions provenant du module d'horloge sont listées dans le catalogue 2 sous le titre — **TIME 2x** (*fonctions TIME, révision 2x*). Les nouvelles fonctions d'horloge se trouvent dans le catalogue 2 sous le titre — **CX TIME** (*fonctions horloge du HP-41CX*).

Mémoire annexe et fonctions étendues

Le HP-41CX comprend une mémoire annexe et des fonctions d'extension qui proviennent principalement du module de fonctions d'extension mémoire; certaines fonctions sont nouvelles. Les possibilités offertes par cette mémoire et ces fonctions sont les suivantes :

- 124 registres de *mémoire annexe* pour les fichiers programmes, données et texte ASCII.
- Fonctions pour créer et manipuler des fichiers dans la mémoire annexe.
- Fonctions pour manipuler les indicateurs binaires, les données et les chaînes alpha.
- Nouveaux tests conditionnels pour les branchements.
- Diverses nouvelles fonctions.

Les fonctions provenant du module de fonctions d'extension mémoire apparaissent dans le catalogue 2 du HP-41CX sous le titre — **EXT FCN 2x** (*fonction d'extension, révision 2x*). Les nouvelles fonctions d'extension mémoire apparaissent dans le catalogue 2 sous le titre — **CX EXT FCN** (*fonctions d'extension HP-41CX*).

Mémoire annexe

Les chapitres 13 (« Mémoire annexe ») et 14 (« Éditeur de texte ») traitent de la mémoire annexe et des manipulations de son contenu. Les fonctions listées ci-après n'existent pas dans le module d'extension mémoire du HP-41C/CV.

L'éditeur de texte est une des principales nouveautés du HP-41CX. La fonction **[ED]** redéfinit le clavier et l'affichage de telle façon que vous puissiez appeler un fichier texte et voir le contenu de l'enregistrement sur lequel vous travaillez au lieu de manipuler le texte par l'intermédiaire d'opérations élémentaires sur le registre ALPHA. Le clavier de l'éditeur est reproduit au dos du HP-41CX et comprend le jeu de caractères alpha (l'illustration au dos du HP-41C/CV ne comporte que le clavier alpha).

Les autres nouvelles fonctions d'extension mémoire sont les suivantes :

- **[ASROOM]** (*espace ASCII*) et **[EMROOM]** (*mémoire annexe*) donnent la quantité d'espaces mémoire restant dans un fichier texte ASCII (page 222) et dans la mémoire annexe (page 208).
- **[EMDIRX]** (*catalogue de mémoire annexe par X*) rappelle le nom et le type d'un fichier et le définit comme fichier courant (page 207).
- **[RESZFL]** change la taille d'un fichier texte ou données (page 213).

Fonctions étendues

Les fonctions d'extension du HP-41CX se répartissent en trois grandes catégories présentées ci-dessous. Certaines de celles-ci proviennent du module d'extension fonctions/mémoire du HP-41C/CV et sont listées dans le catalogue 2.

Fonctions manipulant les indicateurs binaires, les données et les chaînes alpha. Ces fonctions proviennent du module d'extension fonctions/mémoire du HP-41C/CV et sont traitées au chapitre 12, « Mémoire principale », au chapitre 19 « Indicateurs binaires » et au chapitre 21 « Opérations alpha et opérations interactives ».

Tests conditionnels. Ces tests vous permettent de comparer le contenu du registre X à celui d'un registre de données quelconque. Voir chapitre 20.

Divers. Les autres fonctions provenant du module d'extension fonctions/mémoire sont :

- **[PASH]** (affectation programmable) et **[CLKEYS]** (effacement d'affectation de touches), voir chapitre 9, « Le clavier et l'affichage ».
- **[SIZE ?]** (taille mémoire ?) et **[PSIZE]** (taille programmable), voir chapitre 13, « Mémoire principale ».
- **[PCLPS]** (effacement programmable de programmes), voir chapitre 18, « Bases de programmation ».
- **[GETKEY]**, voir chapitre 21, « Opérations interactives ». Cette fonction interrompt l'exécution du programme jusqu'à ce que vous appuyez sur une touche. Le code de celle-ci peut alors servir à déterminer un branchement à un sous-programme.

Les fonctions diverses ne provenant pas du module de fonctions d'extension mémoire sont :

- **[ΣREG ?]** (registres statistiques ?), voir chapitre 11, « Fonctions numériques ».
- **[CLRREG X]** (efface registre par X), voir chapitre 12, « Mémoire principale ».
- **[GETKEYX]** (prend touche par X), voir chapitre 21, « Opérations interactives ».

Nouvelle terminologie utilisée dans ce manuel

De nombreux termes utilisés dans ce manuel diffèrent de ceux utilisés dans la littérature du HP-41 et de ses modules. Si vous êtes habitués à la terminologie précédente, consultez le tableau suivant.

Termes équivalents

HP-41CX	HP-41C/CV	Commentaires
Exécution alphabétique d'une fonction	Exécution d'une fonction à partir de l'affichage	
Alarme périmée inactivée	Alarme périmée non activée	
Alarme de commande sans interruption	Alarme de commande de non-interruption	Pour souligner la différence entre les alarmes conditionnelles et les alarmes de commande

Termes équivalents (suite)

HP-41CX	HP-41C/CV	Commentaires
Alarme de commande avec interruption	Alarme de commande d'interruption	
Fichier en cours	Fichier de travail	
Indicateurs binaires : de l'utilisateur (00-10) de contrôle (11-29)	à usage général (00-10) spécialisés (11-29)	Les indicateurs binaires de l'utilisateur sont exclusivement ceux définis par l'utilisateur alors que les indicateurs de contrôle, eux, sont définis par le HP-41. Cependant, vous pouvez modifier n'importe quel indicateur de l'utilisateur ou de contrôle (mais pas les indicateurs du système, 30-35).
Indicateur de saisie	Soulignement (_)	
Claviers : – du catalogue d'alarmes – alphabétique – normal – du chronomètre – personnel	Modes	Ces états spécifiques sont caractérisés par une redéfinition du clavier.
Modes : – prise de temps intermédiaires – différence (de temps intermédiaires) – rappel (de temps intermédiaires) – stockage (de temps intermédiaires)	Opérations :	Ces états définissent un mode de fonctionnement plus qu'une opération ou une fonction particulière.
Registres au-dessus de R ₉₉	Registres (de données) secondaires	Évite la confusion avec les registres de la mémoire annexe.
Fichiers de texte	Fichiers ASCII	
Registres non affectés, non spécialisés	Registres de programme, mémoire programme	Cette partie de la mémoire ne stocke pas uniquement des programmes.
GTO	GTO	Toutes les fonctions secondaires sont en jaune. Voir au début du manuel.

Codes-barres

Table des matières

SETALM	403
ALMREL	403
SPLITS	404
TR	405
SIGMA	408

SETALM (chapitre 16)

Registres de programme nécessaires : 11

RANGÉE 1 (1 : 3)



RANGÉE 2 (3 : 7)



RANGÉE 3 (7 : 9)



RANGÉE 4 (10 : 12)



RANGÉE 5 (13 : 22)



RANGÉE 6 (23 : 27)



ALMREL (chapitre 16)

Registres de programme nécessaires : 11

RANGÉE 1 (1 : 3)



(continue)

ALMREL (suite)

RANGÉE 2 (3 : 5)



RANGÉE 3 (5 : 9)



RANGÉE 4 (9 : 19)



RANGÉE 5 (20 : 28)



RANGÉE 6 (29 : 34)

**SPLITS (chapitre 17)****Registres de programme nécessaires : 14**

RANGÉE 1 (1 : 3)



RANGÉE 2 (4 : 6)



RANGÉE 3 (6 : 8)



RANGÉE 4 (9 : 16)



RANGÉE 5 (17 : 23)



RANGÉE 6 (24 : 30)



RANGÉE 7 (30 : 38)



RANGÉE 8 (39 : 40)



TR (chapitre 22)

Registres de programme nécessaires : 70

RANGÉE 1 (1 : 6)



RANGÉE 2 (6 : 12)



RANGÉE 3 (12 : 15)



RANGÉE 4 (15 : 20)



RANGÉE 5 (20 : 22)



RANGÉE 6 (23 : 31)



RANGÉE 7 (32 : 38)



RANGÉE 8 (39 : 43)



RANGÉE 9 (43 : 48)



RANGÉE 10 (49 : 54)



RANGÉE 11 (54 : 57)



RANGÉE 12 (58 : 62)



RANGÉE 13 (62 : 68)



(continue)

TR (suite)

RANGÉE 14 (68 : 78)



RANGÉE 15 (78 : 85)



RANGÉE 16 (86 : 93)



RANGÉE 17 (94 : 98)



RANGÉE 18 (98 : 103)



RANGÉE 19 (103 : 111)



RANGÉE 20 (111 : 117)



RANGÉE 21 (118 : 124)



RANGÉE 22 (125 : 133)



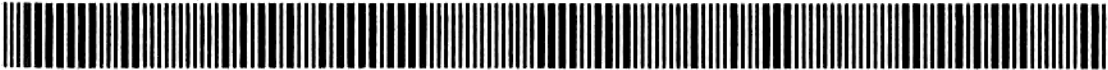
RANGÉE 23 (134 : 137)



RANGÉE 24 (137 : 144)



RANGÉE 25 (145 : 151)



RANGÉE 26 (152 : 155)



(continue)

TR (suite)

RANGÉE 27 (155 : 163)



RANGÉE 28 (163 : 169)



RANGÉE 29 (169 : 177)



RANGÉE 30 (177 : 183)



RANGÉE 31 (184 : 185)



RANGÉE 32 (185 : 194)



RANGÉE 33 (194 : 202)



RANGÉE 34 (203 : 213)



RANGÉE 35 (214 : 222)



RANGÉE 36 (222 : 227)



RANGÉE 37 (228 : 230)



RANGÉE 38 (230 : 233)



SIGMA (chapitre 22)**Registres de programme nécessaires : 100**

RANGÉE 1 (1 : 7)



RANGÉE 2 (7 : 12)



RANGÉE 3 (13 : 18)



RANGÉE 4 (18 : 25)



RANGÉE 5 (25 : 34)



RANGÉE 6 (34 : 37)



RANGÉE 7 (37 : 44)



RANGÉE 8 (44 : 51)



RANGÉE 9 (51 : 56)



RANGÉE 10 (56 : 64)



RANGÉE 11 (65 : 71)



RANGÉE 12 (72 : 79)



RANGÉE 13 (80 : 88)



(continue)

SIGMA (suite)

RANGÉE 14 (88 : 90)



RANGÉE 15 (90 : 99)



RANGÉE 16 (99 : 109)



RANGÉE 17 (109 : 117)



RANGÉE 18 (118 : 126)



RANGÉE 19 (127 : 130)



RANGÉE 20 (130 : 135)



RANGÉE 21 (135 : 140)



RANGÉE 22 (140 : 145)



RANGÉE 23 (145 : 147)



RANGÉE 24 (147 : 153)



RANGÉE 25 (153 : 162)



RANGÉE 26 (163 : 173)



(continued)

SIGMA (suite)

RANGÉE 27 (174 : 181)



RANGÉE 28 (181 : 190)



RANGÉE 29 (190 : 197)



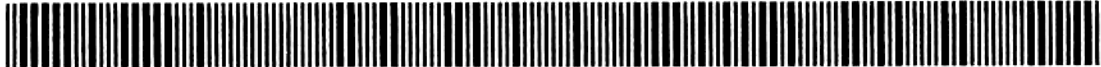
RANGÉE 30 (198 : 205)



RANGÉE 31 (206 : 214)



RANGÉE 32 (214 : 222)



RANGÉE 33 (223 : 231)



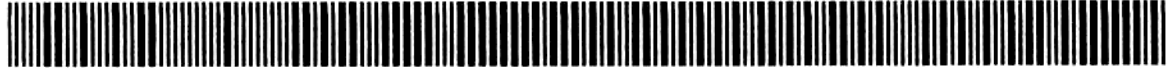
RANGÉE 34 (231 : 239)



RANGÉE 35 (239 : 245)



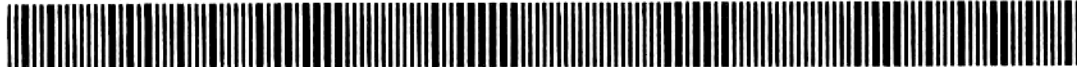
RANGÉE 36 (246 : 251)



RANGÉE 37 (252 : 259)



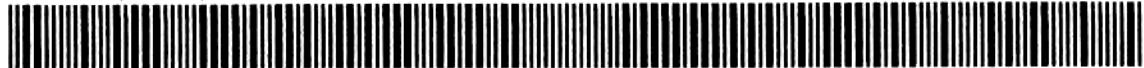
RANGÉE 38 (259 : 267)



RANGÉE 39 (267 : 274)



RANGÉE 40 (275 : 279)



(continue)

SIGMA (suite)

RANGÉE 41 (279 : 288)



RANGÉE 42 (288 : 296)



RANGÉE 43 (296 : 302)



RANGÉE 44 (303 : 304)



RANGÉE 45 (304 : 310)



RANGÉE 46 (310 : 312)



RANGÉE 47 (313 : 321)



RANGÉE 48 (321 : 330)



RANGÉE 49 (331 : 333)



RANGÉE 50 (333 : 340)



RANGÉE 51 (340 : 344)



RANGÉE 52 (345 : 353)



RANGÉE 53 (353 : 361)



RANGÉE 54 (362 : 367)



Tableaux de fonctions

Table des matières

Introduction	414
Localisation d'une fonction	414
Signification des en-têtes de colonnes des tableaux	414
Fonctions de système et de format	416
Fonctions d'effacement	418
Fonctions des registres de données et de la pile	420
Fonctions numériques	423
Fonctions de la mémoire annexe	425
Fonctions d'horloge	429
Fonctions de mise au point	431
Fonctions dirigeant l'exécution de programmes	432
Fonctions alpha	436
Fonctions interactives	438

Introduction



Les 10 tableaux suivants décrivent les fonctions du HP-41CX. Chacun d'eux décrit les fonctions ayant des caractéristiques communes et certaines de ces fonctions apparaissent dans plus d'un tableau. Sur la plupart des tableaux figurent des entrées (voir paragraphes suivants); sur celui des fonctions de la mémoire annexe figurent des entrées bien spécifiques décrites en page 425.


Localisation d'une fonction

- Pour trouver une fonction exécutant une opération spécifique, reportez-vous au tableau de fonctions dont le titre décrit le type d'opération désiré.
- Pour trouver ce que fait une fonction alors que vous n'avez que son nom, reportez-vous à l'index des fonctions (intérieur de couverture). La dernière référence de page listée est celle qui correspond au tableau proprement dit.

Signification des en-têtes de colonnes des tableaux

Nom à l'affichage. Représente le nom d'une fonction figurant dans un listage de programme des catalogues 2 et 3 et le nom d'affichage d'une fonction lorsqu'une touche est maintenue enfoncée. Ce nom est par ailleurs celui que vous devez spécifier pour affecter la fonction au clavier personnel; si la fonction ne figure pas dans cette colonne, vous ne pouvez l'affecter au clavier personnel.

Nom au clavier. Est le nom d'une fonction indiqué sur le clavier Alpha. (Si le nom est imprimé en jaune, vous devez appuyer sur  préalablement). Si la fonction ne figure pas dans cette colonne, vous devez utiliser  et le nom d'affichage ou bien affecter la fonction au clavier personnel.

IND. Un « I » dans cette colonne indique que vous pouvez spécifier indirectement le paramètre de cette fonction. Pour cela, entrez la fonction et appuyez sur ; IND s'affiche après le nom de la fonction. Spécifiez ensuite le registre détenant l'adresse du registre à accéder.





Pile. Montre comment la fonction affecte la pile.

L = *LAST X*. Le contenu précédent du registre X est copié dans le registre LAST X.

↓ = *La pile descend*. Le contenu du registre Z est copié dans le registre Y et celui du registre T est copié dans le registre Z.

↑ = *La pile monte*. Le contenu des registres X, Y et Z est copié dans les registres Y, Z et T respectivement; le contenu précédent du registre T est perdu. (Ce qui suppose que ce mouvement vers le haut était déjà activé).

A = *Mouvement vers le haut activé*. Si la prochaine fonction exécutée montre « ↑ » dans la colonne « Pile » ou si vous entrez un nombre, la pile montera. (La plupart des fonctions activent le mouvement vers le haut).

D = *Mouvement vers le haut désactivé*. Si la prochaine fonction exécutée montre « ↑ » dans la colonne « Pile » ou si vous entrez un nombre, le nouveau nombre du registre X remplace le contenu précédent et la pile ne monte pas. (Seules les touches , ,  et  désactivent le mouvement vers le haut).

N = *Neutre*. Le mouvement vers le haut n'est ni activé, ni désactivé; la pile reste immobile.

Indicateurs. Ces indicateurs affectent ou sont affectés par la fonction.

Octets. Nombre d'octets de mémoire nécessaires lorsque la fonction est utilisée dans un programme. Si la fonction ne figure pas dans cette colonne, c'est qu'elle n'est pas programmable.

Page. Numéro de page de référence pour ce volume. Pour les références se rapportant au volume 1, voir l'index des fonctions en fin de manuel.

Fonctions de système et de format

Beaucoup de ces fonctions comprennent des options qui sont validées définitivement : formats d'affichage, unités d'angles, allocation mémoire, affectations du clavier personnel, etc. Certaines fonctions de système en font partie, par exemple les touches en bascule et les catalogues.

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
	ALPHA	Active/désactive le clavier Alpha.			48		155
AOFF		Désactive le clavier Alpha.		A	48	1	159
AON		Active le clavier Alpha.		A	48	1	159
ASN	ASN	Affecte la fonction ou le label global à la touche spécifiée du clavier personnel.		A			166
CAT <i>n</i>	CATMOG <i>n</i>	Exécute le catalogue spécifié (1 à 6).					
		Catalogues 1, 2, 3, 6.		N			170
		Catalogue 4 (EMDIR).		A		(2)	206
		Catalogue 5 (ALMCAT).		A	31	(2)	255
CF <i>nn</i>	CF <i>nn</i>	Efface l'indicateur binaire <i>nn</i> $00 \leq nn \leq 29$.	I	A	<i>nn</i>	2	288
CLK12		Choisit l'affichage sur 12 heures.		A		2	239
CLK24		Choisit l'affichage sur 24 heures.		A		2	239
CLKEYS		Efface toutes les affectations au clavier personnel.		A		2	168
CLKT		Choisit l'affichage de l'heure uniquement.		A		2	239
CLKTD		Choisit l'affichage de l'heure et la date.		A		2	239
DEG		Choisit le degré comme unité d'angle.		A	42-43	1	186
DMY		Choisit jour-mois-année comme format de date.		A	31	2	242
ENG <i>n</i>	ENG <i>n</i>	Choisit la notation ingénieur avec <i>n</i> décimales.	I	A	36-41	2	161

Fonctions de système et de format (suite)

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
FIX n	FIX n	Choisit la notation virgule fixe avec n décimales.	I	A	36-41	2	160
GRAD		Choisit le grade comme unité d'angle.		A	42-43	1	186
MDY		Choisit mois-jour-année comme format de date.		A	31	2	242
	ON	Éteint/allume le HP-41.			11-26 45-55		155
ON		Allume le HP-41 et supprime l'extinction automatique.			44		155
PASW		Affecte la fonction ou le label spécifié dans le registre Alpha au clavier Alpha en utilisant le code de touche spécifié dans le registre X .		A		2	166
	PRGM	Valide/invalidé le mode programme.		N			155
PRIZE		Affecte au stockage des données le nombre de registres spécifiés par le contenu du registre X .		A		2	199
RAD		Choisit le radian comme unité d'angle.		A	42-43	1	186
RCUFLAG		Place dans le registre X l'état des indicateurs 00 à 43.		†, A		2	296
SCI n	SCI n	Choisit la notation scientifique avec n décimales.	I	A	36-41	2	161
SF nn	SF nn	Arme l'indicateur nn , $00 \leq nn \leq 29$.	I	A	nn	2	288
ΣREG nn		Affecte les registres statistiques aux positions R_{nn} à R_{nn+5} .	I	A		2	190
ΣREG?		Place dans le registre X l'adresse du premier registre statistique courant.		†, A		2	190
SIZE nnn		Affecte au stockage des données nnn registres de la mémoire principale.		A			199

Fonctions de système et de format (suite)

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
SIZE?		Place dans le registre X le nombre de registres affectés au stockage des données.		†, A		2	199
STOFLAG		Restitue les états des indicateurs binaires 00 à 43 en utilisant la donnée d'état du registre X .		A	00-43	2	296
		Restitue les états des indicateurs binaires <i>dd</i> à <i>ff</i> spécifiés par la valeur <i>dd.ff</i> dans le registre X en utilisant la donnée d'état dans le registre X .		A	<i>dd.ff</i>	2	296
	USER	Active/désactive le clavier personnel.		N	27		155

Fonctions d'effacement

Pour mieux comprendre ce tableau, reportez-vous à la page 414, « Signification des en-têtes de colonnes des tableaux ».

Nom à l'affichage	Nom au clavier	Description	Pile	Indicateurs	Octets	Page
	↵	Lorsque l'indicateur de saisie (—) est affiché, efface le dernier symbole entré.		*		158
		Lorsque la saisie, numérique ou alphanumérique, est terminée, efface le contenu du registre X ou du registre Alpha en mode exécution ; en mode programme, supprime la ligne de programme affichée.				159
		Lorsque le message est affiché, efface le message.		N	50	285
	↵ enfoncée, ON, ↵ relâchée	Efface les mémoires de l'ordinateur, principale et annexe (l'heure et la date sont inchangées).			00-55	155
CLA	CLA	Efface le registre Alpha.		A		159

* Lorsque la pression sur ↵ efface le registre X, la montée de la pile est désactivée (D). Sinon, ↵ est neutre (N).

Fonctions d'effacement (suite)

Nom à l'affichage	Nom au clavier	Description	Pile	Indicateurs	Octets	Page
<code>CLALMA</code>		Efface la 1 ^{re} alarme dont le message correspond au contenu du registre Alpha.	A		2	258
<code>CLALMX</code>		Efface l'alarme dont la position dans le listage <code>ALMCAT</code> est spécifiée dans le registre X.	A		2	259
<code>CLD</code>		Efface le message de l'affichage.	A	50	1	318
<code>CLFL</code>		Efface le fichier dont le nom se trouve dans le registre Alpha.	A		2	213
<code>CLKEYS</code>		Efface toutes les affectations sur le clavier personnel.	A		2	168
<code>CLP</code> <i>label</i>		Efface le dernier programme en mémoire contenant le label global spécifié.	A			286
<code>CLRALMS</code>		Efface toutes les alarmes.	A		2	258
<code>CLRG</code>		Efface tous les registres de données de la mémoire principale.	A		1	202
<code>CLRGX</code>		Efface chaque <i>iii</i> ème registre entre R_{ddd} et R_{fff} en mémoire principale tels que spécifiés par <i>ddd.fff</i> dans le registre X.	A		2	202
<code>CLS</code>	<code>CLP</code>	Efface les registres statistiques.	A		1	190
<code>CLST</code>		Efface la pile opérationnelle.	A		1	183
<code>CLX</code>	<code>CLX</code>	Efface le registre X.	D		1	159
<code>DEL</code> <i>nnn</i>		Supprime <i>nnn</i> lignes de programme en commençant à la ligne affichée.	N			286
<code>DELCHR</code>		Supprime le nombre de caractères spécifiés dans le registre X de l'enregistrement d'un fichier ASCII en commençant à la valeur du pointeur.	A		2	225
<code>DELREC</code>		Supprime l'enregistrement courant d'un fichier ASCII.	A		2	224

Fonctions d'effacement (suite)

Nom à l'affichage	Nom au clavier	Description	Pile	Indicateurs	Octets	Page
<code>PCLPS</code>		Efface le dernier programme de la mémoire principale contenant le label global spécifié dans le registre Alpha et tous ceux le suivant.	A		2	286
<code>PURFL</code>		Supprime le fichier dont le nom se trouve dans le registre Alpha .	A		2	208

Fonctions des registres de données et de la pile

Ces fonctions actionnent la pile ou les registres de données, ou prennent un de ces registres comme paramètre. (Pour les fonctions permettant le transfert des données entre des fichiers de données en mémoire annexe et la pile ou des registres de données en mémoire principale, reportez-vous en page 425, « Fonctions de la mémoire annexe ». Pour mieux comprendre ce tableau, voir page 414, « Signification des en-têtes de colonnes des tableaux ».

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
<code>ASTO nn</code>	<code>ASTO nn</code>	Copie les six caractères de gauche du registre Alpha dans R_{nn} .	I	A		2	200
<code>ARCL nn</code>	<code>ARCL nn</code>	Ajoute le contenu de R_{nn} au registre Alpha.	I	A	28, 29 36-41	2	200
<code>CLRG</code>		Efface tous les registres de stockage de données.		A		1	202
<code>CLRGX</code>		Efface chaque <i>iii</i> ème registre entre R_{ddd} et R_{fff} tels que définis par <i>ppp.dddii</i> dans le registre X .		A		2	202
<code>CLV</code>	<code>CLV</code>	Efface les registres statistiques.		A		1	190
<code>CLST</code>		Efface la pile opérationnelle.		A		1	183
<code>CLX</code>	<code>CLX</code>	Efface le registre X.		D		1	159
<code>DSE nn</code>		Décrémente <i>iiiiiffcc</i> dans R_{nn} de la valeur <i>cc</i> et saute l'instruction de programme suivante si <i>iiii - cc ≤ fff</i> .	I	A		2	306

Fonctions des registres des données et de la pile (suite)

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
ENTER ↑	ENTER ↑	Copie le contenu du registre X dans le registre Y et fait monter les contenus des autres registres de la pile.		↑, D		1	175
ISG nn	ISG nn	Incrémente <i>iiii.fffcc</i> dans R_{nn} de la valeur <i>cc</i> et saute l'instruction de programme suivante si $iiii + cc > fff$.	I	A		2	306
LASTX	LAST>	Copie le contenu du registre LAST X dans le registre X et fait monter les contenus des autres registres de la pile.		↑, A		1	179
PSIZE		Affecte au stockage des données le nombre de registres spécifiés par le registre X .		A		2	199
R ↑		Fait monter les contenus des registres de la pile.		A		1	181
RCL nn	RCL nn	Place dans le registre X le contenu du registre R_{nn} .	I	↑, A		*	200
RDN	R ↓	Fait descendre les contenus des registres de la pile.		A		1	181
REGMOVE		Copie les contenus des registres R_{sss} à $R_{sss+nnn}$ dans les registres R_{ddd} à $R_{ddd+nnn}$ tels que spécifiés par <i>sss.dddnnn</i> dans le registre X .		A		2	201
REGSWAP		Échange les contenus des registres R_{sss} à $R_{sss+nnn}$ avec ceux des registres R_{ddd} à $R_{ddd+nnn}$ tels que spécifiés par <i>sss.dddnnn</i> dans le registre X .		A		2	201
Σ+	Σ+	Cumuls statistiques.		L, D		1	191
Σ-	Σ-	Correction des cumuls statistiques.		L, D		1	191

* Si $00 \leq nn \leq 15$, nécessite 1 octet; sinon, nécessite 2 octets.

Fonctions des registres des données et de la pile (suite)

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
Σ REG <i>nn</i>		Affecte les registres statistiques aux positions R_{nn} à R_{nn+5} .	I	A		2	190
Σ REG?		Place dans le registre X l'adresse du premier registre statistique courant.		t, A		2	190
SIZE <i>nnn</i>		Affecte au stockage des données <i>nnn</i> registres de la mémoire principale.		A			199
SIZE?		Place dans le registre X le nombre de registres affectés au stockage des données.		t, A		2	199
ST+ <i>nn</i>	STO + <i>nn</i>	Ajoute le contenu du registre X au contenu de R_{nn} et place la somme dans R_{nn} .	I	A		2	201
ST- <i>nn</i>	STO - <i>nn</i>	Soustrait le contenu du registre X de celui de R_{nn} et place la différence dans R_{nn} .	I	A		2	201
ST* <i>nn</i>	STO * <i>nn</i>	Multiplie le contenu du registre X par celui de R_{nn} et place le produit dans R_{nn} .	I	A		2	201
ST/ <i>nn</i>	STO / <i>nn</i>	Divise le contenu du registre R_{nn} par celui du registre X et place le quotient dans R_{nn} .	I	A		2	201
STO <i>nn</i>	STO <i>nn</i>	Copie le contenu du registre X dans le registre R_{nn} .	I	A		*	200
VIEW <i>nn</i>	VIEW <i>nn</i>	Affiche le contenu du registre R_{nn} .	I	A	21,50 55	2	319
X \leftrightarrow Y <i>nn</i>		Echange les contenus des registres X et R_{nn} .	I	A		2	201
X \leftrightarrow F		Echange les contenus du registre X et les indicateurs 00 à 07.		A	00-07	2	295
X \leftrightarrow Y	X \leftrightarrow Y	Echange les contenus des registres X et Y.		A		1	181

* Si $00 \leq nn \leq 15$, nécessite 1 octet; sinon, nécessite 2 octets.

Fonctions numériques

Toutes les fonctions numériques sont programmables; elles utilisent toutes un octet de mémoire programme. Les opérations de fonctions trigonométriques et de conversions de coordonnées rectangulaires/polaires dépendent de l'unité d'angle (indicateurs binaires 42 et 43). Pour mieux comprendre ce tableau, reportez-vous à la page 414, « Signification des en-têtes de colonnes des tableaux ».

Nom à l'affichage	Nom au clavier	Description	Pile	Page
$+$	$+$	Addition.	L, ↓, A	188
$-$	$-$	Soustraction.	L, ↓, A	188
$*$	\times	Multiplication.	L, ↓, A	188
$/$	\div	Division.	L, ↓, A	188
$1/x$	$1/x$	Inverse.	L, A	185
$10 \uparrow X$	$\wedge 10$	Exponentielle en base 10.	L, A	187
ABS		Valeur absolue.	L, A	186
$ACOS$	\cos^{-1}	Arc cosinus.	L, A	186
$ASIN$	\sin^{-1}	Arc sinus.	L, A	186
$ATAN$	\tan^{-1}	Arc tangente.	L, A	186
CHS	\overline{CHS}	Changement de signe.	A	185
COS	\overline{COS}	Cosinus.	L, A	186
$D \rightarrow R$		Conversion degrés en radians.	L, A	187
DEC		Conversion octal en décimal.	L, A	187
$E \uparrow X$	e^x	Exponentielle en base e .	L, A	187
$E \uparrow X - 1$		Exponentielle en base e pour des arguments proches de zéro.	L, A	187
$FACT$		Factorielle.	L, A	185
FRC		Partie fractionnaire.	L, A	186
HMS		Conversion d'heures décimales en heures, minutes et secondes.	L, A	187
$HMS \uparrow$		Addition d'heures, minutes et secondes.	L, ↓, A	188
$HMS -$		Soustraction d'heures, minutes et secondes.	L, ↓, A	188
HR		Conversion d'heures, minutes et secondes en heures décimales.	L, A	187
INT		Partie entière.	L, A	186
LN	\overline{LN}	Logarithme népérien.	L, A	187
$LN \uparrow X$		Logarithme népérien pour arguments proches de 1.	L, A	187

Fonctions numériques (suite)

Nom à l'affichage	Nom au clavier	Description	Pile	Page
LOG	LOG	Logarithme en base 10.	L,A	187
MEAN		Moyenne.	L,A	192
MOD		Modulo (reste)	L, ↓ ,A	190
OCT		Conversion décimale en octal.	L,A	187
P→R	P→R	Conversion polaire en rectangulaire.	L,A	189
%	%	Pourcentage.	L,A	188
%CH		Différence en pourcents.	L,A	188
PI	PI	Pi (3,141592654)	↑ ,A	159
R→D		Conversions radians en degrés.	L,A	187
R→P	R→P	Conversion rectangulaire en polaire.	L,A	189
RND		Arrondi.	L,A	186
SDEV		Ecart-type.	L,A	192
Σ+	Σ+	Accumulations pour statistiques.	L,D	191
Σ-	Σ-	Correction d'accumulations.	L,D	191
SIN	SIN	Sinus.	L,A	186
SIGN		Signe de x.	L,A	186
SQRT	√x	Racine carrée.	L,A	185
TAN	TAN	Tangente.	L,A	186
X↑2	x²	Carré.	L,A	185
Y↑X	y^x	Puissance.	L, ↓ ,A	189

Fonctions de la mémoire annexe

Toutes les fonctions de la mémoire annexe sont programmables, elles utilisent toutes 2 octets de mémoire programme. Pour mieux comprendre ce tableau, reportez-vous à la page 414, signification des en-têtes de colonnes des tableaux et à la description suivante des en-têtes bien spécifiques de ce tableau.

Type de fichier. Les fonctions dotées d'une indication (P, D ou A) dans cette colonne agissent uniquement sur les fichiers du type indiqué.

P = Fichier **P**rogramme

D = Fichier de **D**onnées

A = Fichier Texte (**A**SCII)

Nom de fichier. Les fonctions dotées d'une indication (oui, OK ou non) dans cette colonne agissent sur les fichiers spécifiés comme suit.

Oui = Vous devez entrer le nom du fichier désiré dans le registre Alpha.

OK = Vous pouvez entrer le nom du fichier désiré dans le registre Alpha ou effacer le registre Alpha pour spécifier le fichier courant.

Non = Agit sur le fichier courant uniquement.

Pointeur. Les fonctions dotées d'une indication dans cette colonne agissent suivant la valeur en cours du pointeur.

RRR = pointeur de registre (pour fichiers de données).

rrr.ccc = pointeur d'enregistrement/caractère (pour fichiers texte).

rrr. = pointeur d'enregistrement uniquement (pour fichiers texte).

Nom à l'affichage	Type de fichier	Description	Nom de fichier	Pointeur	Pile	Indicateurs binaires	Page
APPCHR	A	Ajoute le contenu du registre Alpha à la fin de l'enregistrement courant.	Non	<i>rrr.</i>	A		224
APPREC	A	Ajoute le contenu du registre Alpha à la fin du fichier courant comme nouvel enregistrement.	Non		A		222
ARCLREC	A	Ajoute au registre Alpha l'enregistrement courant en commençant à la valeur du pointeur.	Non	<i>rrr.ccc</i>	A	17	226
ASROOM	A	Donne dans le registre X le nombre d'octets disponibles dans le fichier courant.	Non		†,A		222

Fonctions de la mémoire annexe (suite)

Nom à l'affichage	Type de fichier	Description	Nom de fichier	Pointeur	Pile	Indicateurs binaires	Page
<input type="checkbox"/> CLFL	A,D	Efface le fichier nommé dans le registre Alpha .	Oui		A		213
<input type="checkbox"/> CHFLAS	A	Crée le fichier ASCII nommé dans le registre Alpha avec le nombre de registres spécifiés dans le registre X.	Oui		A		212
<input type="checkbox"/> CHFLD	D	Crée le fichier données nommé dans le registre Alpha avec le nombre de registres spécifiés dans le registre X .	Oui		A		211
<input type="checkbox"/> DELCHR	A	Supprime de l'enregistrement le nombre de caractères spécifiés dans le registre X , en commençant à la valeur du pointeur.	Non	<i>rrr.ccc</i>	A		225
<input type="checkbox"/> DELREC	A	Supprime l'enregistrement courant.	Non	<i>rrr.</i>	A		224
<input type="checkbox"/> ED	A	Active le clavier et l'affichage pour la mise au point de texte.	OK	<i>rrr.ccc</i>	A	26, 28, 48	228
<input type="checkbox"/> EMDIR		Liste le catalogue de la mémoire annexe.			A		206
<input type="checkbox"/> EMDIRX		Cherche un fichier selon son rang dans le listage <input type="checkbox"/> EMDIR spécifié par le contenu du registre X . Place le nom du fichier dans le registre Alpha et le type dans le registre X.			L,A		207
<input type="checkbox"/> EMROOM		Place dans le registre X le nombre maximal d'enregistrements pour un nouveau fichier.			†,A		208
<input type="checkbox"/> FLSIZE	A,D,P	Place dans le registre X le nombre d'enregistrements du fichier.	OK		†,A		208
<input type="checkbox"/> GETAS	A	Copie un fichier de stockage de masse dans un fichier de la mémoire annexe nommé dans le registre Alpha .	Oui		A		227

Fonctions de la mémoire annexe (suite)

Nom à l'affichage	Type de fichier	Description	Nom de fichier	Pointeur	Pile	Indicateurs binaires	Page
GETP	P	Remplace le dernier programme en mémoire par le fichier nommé dans le registre Alpha .	Oui		A	27	209
GETR	D	Copie les registres correspondants du fichier en mémoire principale.	OK		A		217
GETRES	A	Copie l'enregistrement dans le registre Alpha, en commençant au pointeur.	Non	<i>rrr.ccc</i>	A	17	226
GETRX	D	Copie les registres de fichier dans les registres R_{ppp} à R_{ddd} spécifiés par <i>ppp.ddd</i> dans le registre X , en commençant à la valeur du pointeur.	Non	<i>RRR</i>	A		218
GETSUI	P	Copie en fin de mémoire programme le fichier nommé dans le registre Alpha .	Oui		A	27	209
GETX	D	Copie le registre de fichier courant dans le registre X.	Non	<i>RRR</i>	†,A		220
INSCHR	A	Insère le contenu du registre Alpha en commençant à la valeur du pointeur.	Non	<i>rrr.ccc</i>	A		224
INSREC	A	Insère le contenu du registre Alpha comme nouvel enregistrement à la valeur du pointeur.	Non	<i>rrr.</i>	A		222
POSFL	A	Cherche dans le fichier la chaîne spécifiée dans le registre Alpha et place la position dans le registre X (-1 s'il ne trouve pas la chaîne).	Non	<i>rrr.ccc</i>	†,A		226
PURFL	A,D,P	Supprime le fichier nommé dans le registre Alpha .	Oui		A		208
RCLPT	A,D	Place dans le registre X la valeur du pointeur.	Non	<i>rrr.ccc</i>	†,A		216
	P	Place le nombre d'octets dans le fichier en cours.	Non		†,A		211

Fonctions de la mémoire annexe (suite)

Nom à l'affichage	Type de fichier	Description	Nom de fichier	Pointeur	Pile	Indicateurs binaires	Page
RCLPTA	A,D	Place dans le registre X la valeur du pointeur.	OK	<i>rrr.ccc</i> <i>RRR</i>	↑,A		216
	P	Place dans le registre X le nombre d'octets du programme.	OK		↑,A		211
RESZFL	A,D	Change la taille du fichier courant selon le nombre <i>n</i> de registres spécifiés dans le registre X.	Non		A		213
SAVEAS	A	Copie le fichier de la mémoire annexe nommé dans le registre Alpha dans le fichier de stockage de masse nommé dans le registre Alpha.	Oui		A		227
SAVEP	P	Copie le programme nommé dans le registre Alpha dans le fichier nommé dans le registre Alpha.	Oui		A		208
SAVER	D	Copie tous les registres de la mémoire principale dans le fichier.	OK		A		217
SAVERX	D	Copie le contenu des registres R_{ddd} à R_{fff} de la mémoire principale, spécifiés par <i>ddd.fff</i> dans le registre X , dans le fichier en commençant au registre courant.	Non	<i>RRR</i>	A		218
SAVEK	D	Copie le contenu du registre X dans le registre courant.	Non	<i>RRR</i>	A		220
SEENPT	A,D	Place le pointeur du fichier courant à la valeur du registre X.	Non		A		215
SEENPTA	A,D	Place le pointeur à la valeur du registre X.	OK		A		215

Fonctions d'horloge

Pour mieux comprendre ce tableau, reportez-vous à la page 414, « Signification des en-têtes de colonnes des tableaux ».

Nom à l'affichage	Description	Pile	Indicateurs	Octets	Page
ADATE	Ajoute le contenu du registre X au contenu du registre Alpha dans le format de date courant (DMY ou MDY).	A	31,36-39	2	243
ALMCAT	Liste toutes les alarmes, en commençant par celle qui arrive à échéance en premier. Peut être aussi exécuté par CATALOG 5.	A	31	2	255
ALMNOW	Active les alarmes de commande périmées.	A		2	261
ATIME	Ajoute le contenu du registre X au registre Alpha dans le format d'heure courant (CLK12 ou CLK24).	A	36-39	2	240
ATIME24	Ajoute le contenu du registre X au registre Alpha dans le format CLK24.	A	36-39	2	241
CLALMA	Efface la 1 ^{re} alarme dont le message ou label correspond au contenu du registre Alpha.	A		2	258
CLALMX	Efface l'alarme dont la position dans le listage ALMCAT est spécifiée par le contenu du registre X .	A		2	259
CLK12	Choisit le format 12 heures pour l'affichage de l'heure.	A		2	239
CLK24	Choisit le format 24 heures pour l'affichage de l'heure.	A		2	239
CLKT	Choisit l'affichage de l'heure uniquement pour l'horloge.	A		2	239
CLKYD	Choisit l'affichage heure et date pour l'horloge.	A		2	239
CLOCK	Affiche l'horloge.	A	31	2	238
CLRALMS	Efface toutes les alarmes.	A		2	258
CORRECT	Met l'horloge à l'heure et ajuste le facteur d'exactitude.	A		2	238
DATE	Place la date courante dans le registre X.	↑,A	31	2	242
DATE+	Calcule une nouvelle date à partir de celle dans le registre Y et du nombre de jours dans le registre X. La nouvelle date peut être antérieure ou postérieure selon le signe du contenu du registre X.	L,↑,A	31	2	244

Fonctions d'horloge (suite)

Nom à l'affichage	Description	Pile	Indicateurs	Octets	Page
DDAYS	Calcule le nombre de jours entre la date dans le registre Y et celle dans le registre X. Ce nombre est positif si la date dans Y précède celle dans X, sinon il est négatif.	L, ↓, A	31	2	244
DMY	Choisit le format jour-mois-année.	A	31	2	242
DOW	Remplace la date dans le registre X par le numéro du jour de la semaine correspondant.	L, A	31	2	244
HMS	Convertir le contenu du registre X d'heure décimale en heure, minutes, secondes.	L, A		1	187
HMS+	Ajoute le contenu du registre X au contenu du registre Y en format heure, minutes, secondes.	L, ↓, A		1	188
HMS-	Soustrait le contenu du registre X de celui du registre Y en format heure, minutes, secondes.	L, ↓, A		1	188
HFI	Convertit le contenu du registre Y d'heure, minutes, secondes en heure décimale.	L, A		1	187
MDY	Choisit le format mois-jour-année.	A	31	2	242
RCLAF	Rappelle le facteur d'exactitude de l'horloge dans le registre X.	↑, A		2	376
RCLALM	Rappelle dans X l'alarme dont la position dans le listage ALMCA1 est spécifiée par le contenu du registre X .	L, ↑, A *	31	2	252
RCLSW	Place dans le registre X la valeur du chronomètre.	↑, A		2	273
RUNSW	Déclenche le chronomètre.	A		2	273
SETAF	Définit le facteur d'exactitude de l'horloge.	A		2	376
SETDATE	Met l'horloge à la date spécifiée dans le registre X .	A	31	2	242
SETIME	Met l'horloge à l'heure spécifiée dans le registre X .	A		2	237
SETSW	Définit l'heure de départ du chronomètre à la valeur spécifiée dans le registre X .	A		2	273
STOPSW	Arrête le chronomètre.	A		2	273
SW	Active le clavier et l'affichage du chronomètre.	A	26	2	266

* Copie le contenu du registre Y dans le registre T, sans se soucier si le mouvement vers le haut est activé.

Fonctions d'horloge (suite)

Nom à l'affichage	Description	Pile	Indicateurs	Octets	Page
[SWPT]	Définit les pointeurs de stockage et du rappel du chronomètre par la valeur <i>sss.rrr</i> dans le registre X.	L,A	26	2	274
[T+X]	Ajuste l'heure avec le contenu du registre X.	A		2	238
[TIME]	Place l'heure courante dans le registre X.	↑,A		2	240
[XYZALM]	Définit une alarme avec l'heure du registre X, la date du registre Y, l'intervalle de répétition du registre Z et le message ou label du registre Alpha.	A	31	2	250

Fonctions de mise au point

Ces fonctions ne sont pas programmables et sont exécutées en mode programme. Elles permettent d'écrire ou de mettre au point vos programmes. Comme les touches [ON], [USER] et [ALPHA], ces fonctions ne nécessitent pas de revenir en mode exécution pour une exécution. Pour mieux comprendre ce tableau, reportez-vous en page 414, « Signification des en-têtes de colonnes des tableaux ».

Nom à l'affichage	Nom au clavier	Description	Indicateurs	Page
	[←]	Quand l'indicateur de saisie est affiché, efface le dernier chiffre ou le dernier caractère entré ; sinon, efface la ligne de programme affichée.		285
[ASN]	[ASN]	Affecte la fonction ou le label spécifié à la touche spécifiée du clavier personnel.		166
[BST]	[BST]	Affiche la ligne de programme précédente.		284
[CAT] <i>n</i>	[CATALOG] <i>n</i>	Exécute le catalogue spécifié (1 à 6).		170
[CLP] <i>label</i>		Efface le dernier programme en mémoire principale contenant le label global spécifié.		286
[COPY] <i>label</i>		Copie en mémoire, le programme ROM contenant le label global spécifié.		281

Fonctions de mise au point (suite)

Nom à l'affichage	Nom au clavier	Description	Indicateurs	Page
DEL <i>nnn</i>		Supprime <i>nnn</i> lignes de programme en commençant à la ligne affichée.		286
	GTU <input type="checkbox"/>	Affiche le numéro de ligne ou le label global spécifié.		283
	GTD <input type="checkbox"/> <input type="checkbox"/>	Compacte la mémoire programme et crée un programme nul en fin de mémoire programme.		281
ON		Allume le HP-41 et supprime l'extinction automatique.	44	155
PACK		Compacte la mémoire programme.		198
SIZE <i>nnn</i>		Affecte au stockage des données <i>nnn</i> registres de la mémoire principale.		199
SST	SST	Affiche la ligne de programme suivante.		284

Fonctions dirigeant l'exécution de programmes

Ces fonctions permettent de stopper l'exécution d'un programme ou de provoquer l'exécution directe de certaines lignes de programme. Pour mieux comprendre ce tableau, reportez-vous à la page 414, « Signification des en-têtes de colonnes des tableaux ».

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
AVIEW	AVIEW	Affiche le contenu du registre Alpha; si l'indicateur 21 est armé et l'indicateur 55 désarmé, arrête l'exécution.		A	21,50 55	1	318
CLOCK		Arrête l'exécution et affiche l'horloge.		A	31	2	238
OSE <i>nn</i>		Décrémente <i>iiii.ffff</i> dans R_{nn} de <i>cc</i> et saute l'instruction de programme suivante si <i>iiii-cc ≤ fff</i>	I	A		2	306
END		Marque la fin d'un programme.		A		3	301
FC? <i>nn</i>		Teste l'indicateur <i>nn</i> ($00 ≤ nn ≤ 55$) et saute l'instruction suivante si l'indicateur est armé.	I	A	<i>nn</i>	2	304

Fonctions qui dirigent l'exécution de programme (suite)

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
PC?C <i>nn</i>		Teste et désarme l'indicateur <i>nn</i> ($00 \leq nn \leq 29$) et saute l'instruction suivante sauf si l'indicateur était désarmé.	I	A	<i>nn</i>	2	304
PS? <i>nn</i>	PS? <i>nn</i>	Teste l'indicateur <i>nn</i> ($00 \leq nn \leq 55$) et saute l'instruction suivante sauf si l'indicateur est armé.	I	A	<i>nn</i>	2	304
PS?C <i>nn</i>		Teste et efface l'indicateur <i>nn</i> ($00 \leq nn \leq 29$) et saute l'instruction suivante sauf si l'indicateur était armé.	I	A	<i>nn</i>	2	304
GETP		Remplace le dernier programme en mémoire par le fichier programme nommé dans le registre Alpha . Si le dernier programme appelle un nouveau (et est remplacé), l'exécution revient à la 1 ^{re} ligne du nouveau programme.		A	27	2	209
GTO <i>label</i>	GTO <i>label</i>	Transfère l'exécution au label spécifié.	I	A		*	300
ISG <i>nn</i>	ISG <i>nn</i>	Incrémente <i>iiii.fcc</i> dans R_{nn} de <i>cc</i> et saute l'instruction suivante si $iiii + ccc > fff$.	I	A		2	306
LBI	LBI	Label global, numérique local ou Alpha local.		A		‡‡	299
OFF		Met l'ordinateur hors tension.		N	11-26 44-55	1	292
PROMPT		Affiche le contenu du registre Alpha et arrête l'exécution.		A	50	1	314

* Si $00 \leq nn \leq 14$ ou si un paramètre est indirectement spécifié, nécessite 2 octets; si le paramètre est un label global de *m* caractères, nécessite $2 + m$ octets; sinon nécessite 3 octets.

‡‡ Si $00 \leq nn \leq 14$, nécessite 1 octet; si le paramètre est un label global de *m* caractères, nécessite $4 + m$ octets; sinon, nécessite 2 octets.

Fonctions qui dirigent l'exécution de programme (suite)

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
RTM	RTM	Renvoie l'exécution à la ligne suivant l'instruction XEQ ayant appelé le sous-programme.		A		1	301
STOP	R/S	Stoppe l'exécution.		A		1	302
VIEW nn	VIEW nn	Affiche les contenus de R_{nn} et, si l'indicateur 21 est armé et l'indicateur 55 désarmé, arrête l'exécution.	1	A	21,50 55	2	319
X = 0?	X = 0?	Saute l'instruction suivante sauf si le contenu du registre X = 0.		A		1	304
X ≠ 0?		Saute l'instruction suivante sauf si le contenu du registre X ≠ 0.		A		1	304
X < 0?		Saute l'instruction suivante sauf si le contenu du registre X < 0.		A		1	304
X <= 0?		Saute l'instruction suivante sauf si le contenu du registre X ≤ 0.		A		1	304
X > 0?		Saute l'instruction suivante sauf si le contenu du registre X > 0.		A		1	304
X = Y?	X = Y?	Saute l'instruction suivante sauf si le contenu du registre X = contenu du registre Y.		A		1	304
X ≠ Y?		Saute l'instruction suivante sauf si le contenu du registre X ≠ contenu du registre Y.		A		1	304
X < Y?		Saute l'instruction suivante sauf si le contenu du registre X < contenu du registre Y.		A		1	304
X <= Y?	X ≤ Y?	Saute l'instruction suivante sauf si le contenu du registre X ≤ contenu du registre Y.		A		1	304
X > Y?	X > Y?	Saute l'instruction suivante sauf si le contenu du registre X > contenu du registre Y.		A		1	304

Fonctions qui dirigent l'exécution de programme (suite)

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
<code>X = NN?</code>		Saute l'instruction suivante sauf si le contenu du registre X = contenu de R_{nn} , nn spécifié dans le registre Y.		A		2	305
<code>X ≠ NN?</code>		Saute l'instruction suivante sauf si le contenu du registre X ≠ contenu de R_{nn} , nn spécifié dans le registre Y.		A		2	305
<code>X < NN?</code>		Saute l'instruction suivante sauf si le contenu du registre X < contenu de R_{nn} , nn spécifié dans le registre Y.		A		2	305
<code>X ≤ NN?</code>		Saute l'instruction suivante sauf si le contenu du registre X ≤ contenu de R_{nn} , nn spécifié dans le registre Y.		A		2	305
<code>X > NN?</code>		Saute l'instruction suivante sauf si le contenu du registre X > contenu de R_{nn} , nn spécifié dans le registre Y.		A		2	305
<code>X ≥ NN?</code>		Saute l'instruction suivante sauf si le contenu du registre X ≥ contenu de R_{nn} , nn spécifié dans le registre Y.		A		2	305
<code>REQ label</code>	<code>XEQ label</code>	Appelle le label spécifié comme sous-programme.	I	A		*	301

* Si le label est spécifié indirectement, nécessite 2 octets ; si un label local est spécifié, nécessite 3 octets ; si un label global de m caractères est spécifié, nécessite $2 + m$ octets.

Fonctions Alpha

Ces fonctions permettent d'entrer et de sortir des données du registre Alpha et de manipuler les données dans ce même registre. Ne sont pas incluses les fonctions qui utilisent le registre Alpha pour un nom de fichier. Pour mieux comprendre ce tableau, reportez-vous à la page 414, « Signification des en-têtes de colonnes des tableaux ».

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
	[F]	Annexe les caractères suivants au registre Alpha.		N			159
[ADATE]		Annexe le contenu de X au registre Alpha dans le format de date courant ([DMY] ou [MDY]).		A	31, 36-39	2	243
[ALENG]		Donne le nombre de caractères du registre Alpha dans le registre X.		↑,A		2	313
[ANUM]		Donne la première chaîne numérique du registre Alpha dans le registre X.		↑,A	22,28 29	2	311
[AOFF]		Désactive le clavier Alpha.		A	48	1	159
[AON]		Active le clavier Alpha.		A	48	1	159
[ARCL] nn	[ARCL] nn	Annexe le contenu de R _{nn} au registre Alpha.	I	A	28,29 36-41	2	200
[ARCLREC]		Annexe un enregistrement ASCII, en commençant à la valeur du pointeur, au registre Alpha.		A	17	2	226
[AROT]		Effectue une rotation circulaire du contenu du registre Alpha du nombre de positions spécifié dans le registre X . Rotation vers la gauche pour un nombre positif et vers la droite pour un nombre négatif.		A		2	313
[ASHF]		Décale les six caractères de gauche hors du registre Alpha.		A		1	200
[ASTO] nn	[ASTO] nn	Copie les six caractères de gauche du registre Alpha dans R _{nn} .	I	A		2	200

Fonctions Alpha (suite)

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
ATIME		Ajoute le contenu du registre X au registre Alpha dans le format de temps courant (<code>CLK12</code> ou <code>CLK24</code>).		A	36-39	2	240
ATIME24		Annexe le contenu du registre X au registre Alpha en format <code>CLK24</code> .		A	36-39	2	241
ATOX		Décale l'octet de gauche hors du registre Alpha et place sa valeur décimale dans le registre X.		↑, A		2	311
AVIEW	AVISW	Affiche le contenu du registre Alpha.		A	21,50 55	1	318
CLA	CLA	Efface le registre Alpha.		A		1	159
GETREC		Copie un enregistrement ASCII, en commençant à valeur du pointeur, dans le registre Alpha.		A	17	2	226
POSA		Cherche dans le registre Alpha la chaîne spécifiée dans le registre X et place la position dans le registre X (ou -1 si la chaîne ne s'y trouve pas).		L, A		2	312
PROMPT		Affiche le contenu du registre Alpha et arrête l'exécution du programme.		A	21,50 55	1	314
XTOA		Convertit le contenu du registre X en l'octet équivalent et l'annexe au registre Alpha.		A		2	309

TABLEAU 10 - TONALITÉS ET INDICATEURS

Pour mieux comprendre ce tableau, reportez-vous à la page 414, « Signification des en-têtes de colonnes des tableaux ».

Nom à l'affichage	Nom au clavier	Description	IND	Pile	Indicateurs	Octets	Page
[Fwd]		Avance le papier (s'il y a une imprimante).		A	21,55	1	368
[F4]		Emet quatre tonalités.		A	26	1	319
[Pause]		Arrête l'exécution pendant 10 secondes ou jusqu'à ce que vous appuyiez sur une touche ; place le code de touche dans le registre X (ou 0 si vous n'appuyez sur aucune touche).		↑, A		2	317
[F5]		Arrête l'exécution pendant l' intervalle spécifié dans le registre X ou jusqu'à ce que vous appuyiez sur une touche ; place le code de touche dans le registre X et le code ASCII dans le registre Y.		L, ↑, A *	28,48	2	317
[F6]		Affiche le contenu du registre Alpha et arrête l'exécution.		A	50	1	314
[F7]		Interrompt l'exécution pendant une seconde environ.		A		1	315
[F8] n		Emet la tonalité n, 0 ≤ n ≤ 9.	I	A	26	2	319

* Copie les contenus des registres Y et Z dans les registres Z et T respectivement, sauf si le mouvement vers le haut de la pile est validé.

Index

L'index ci-dessous fait référence aussi bien aux pages du volume 1 (pages 1 à 145) qu'à celles du volume 2 (pages 145 à 440). Pour simplifier vos recherches, nous avons fait figurer cet index dans les deux volumes.

A

- Addition, 51
- Adressage, 37, 162
 - indirect, 162, 200
- Affectation d'une fonction à une touche, 46, 156, 166
- Affichage. Cf. aussi Effacement, Message, Programme,
 - Défilement, Aff. fnct. param. standard, 161
 - alpha. de caractères nuls, 366
 - alphabétique, 27, 161
 - alphabétique, défilement, 27
 - d'indic. binaires de ponctuation, 290
 - de caractères, 310
 - de caractères nuls, 366
 - de la fonction des touches, 169
 - de résultats de programmes, 93
 - des messages, 161
 - du contenu d'un registre, 39, 319
 - du registre ALPHA, 318
 - en notation fixe, 31, 160
 - en notation ingénieur, 33, 161
 - effacement de, 19, 20
 - formats d', 31
 - indic. binaires de format d', 291
- Alarme
 - activation, 253
 - affichage, 253
 - annulation, 69, 247-248, 253-254
 - armement, 250-251
 - armement par programme, 261-263
 - arrêt, 69, 258
 - capacité mémoire nécessaire, 198
 - Cf. Catalogue 5
 - clavier du catalogue de, 71-73, 256-257
 - condition de, 361
 - date, 68, 250
 - dépassé, 260
 - effacement de, 69, 258-259, 261
 - exemple, 69
 - heure, 68, 251
 - inactivée, 260
 - intervalle de répétition, 68, 250
 - message, 68, 247-251
 - niveaux de, 363
 - numéro de, 252, 255, 258
 - période de répétition, 68, 250
 - rappel de, 252
 - tonalités, 253
 - types de, 247-249
 - témoin de, 260
- Alarms de commande avec interruption, 248, 251, 259-260-261
- Alarms de commande sans interruption, 248, 251, 260-261
- Alarme-message, 247-251
- Alarme périmée
 - activation automatique, 260, 361-363
 - effacement, 261
 - exécution, 361
- Alarme périmée de commande, activation, 261
- Alarme périmée dépassée, 360
- Alarme périmée et modes du HP-41, 362
- Alarms simultanées, 255
- Alimentation électrique faible, 383
- Alimentation électrique, effets d'une interruption de, 382
- ALMREL, programme, 263
- ALPHA, 15, 230
- Alpha(bétique)
 - affichage, 27, 161
 - clavier, 14, 15, 24, 155-156, 157, 159, 230
 - exécution, 44-45, 282
 - registre, 27, 158, 206, 222, 226, 312
 - saisie, 159
 - spécification de paramètre, 162
 - chaînes, 26, 86, 117, 159-160, 197, 213, 309, 366
 - chiffres, 26, 200, 309
 - jeu de caractères, 14, 24, 232
 - label, 169
 - nom, 44
- Annexion
 - touche d', 27, 159
 - de caractères, 159, 367
- Annulation d'une fonction, 169

Arc cosinus, 54
 Arc sinus, 54
 Arc tangente, 54
 Arithmétique. *Cf. aussi* Calculs, Opérations non commutatives.
 directe dans les registres, 40-42, 201
 sur valeurs horaires, 65, 188
 vectorielle, 59
 vectorielle (exemple), 109
 Armement/Désarmement d'un indic. binaire, 288
 Arrondi, fonction d'arrondi d'un nombre, 186
 ASCII, fichier (*cf.* fichiers de texte), 113
 Assistance technique, 390

B

BAT, 34, 160, 230, 383
 Batteries, 381, 382
 recommandées, 384
 durée de vie, 381
 installation, 384
 jeu de, 382
 puissance, 34, 160
 Boucle, 298, 305-306
 Branchement, 88
 à un label, 298, 301
 dans des boucles, 305
 fonctions pour boucles, 306
 multiplets requis pour, 300
 saut de ligne, 298, 304

C

Caractères non standard de fichiers de texte, 229
 Calculs. (*Cf. aussi* Constantes, calculs avec), 16, 21, 175, 176
 avec constantes, 24, 177-178, 180
 dans la pile opérationnelle, 180, 182
 dépassant les limites sup. et inf. *Cf.* limites sup. et inférieures non commutatifs, 22, 176, 180, 181
 avec parenthèses imbriquées, 176
 sur des dates, 66
 sur valeurs horaires, 64
 Caractère. *Cf. aussi* caractères alpha; caractères des fichiers de texte code de, 310, 317
 effacement de, 20
 pointeur de. *Cf.* Pointeur d'enregistrement/caractère saisie.
 Cf. Saisie de caractères alphabétiques
 de texte, 212
 fictif, 229
 nul, 311

 nul dans nom de fichier, 367
 nul dans registre ALPHA, 309, 366
 nul dans une chaîne, 366-367
 nul et caractères annexés, 367
 nul (affichage de), 366
 nul (effacement de), 367
 Caractères alpha(bétiques)
 affichables, 309
 convertibles en nombres, 309, 311
 copie, 200
 dans les programmes, 94
 effacement, 26
 non standard, 309
 saisie, 26, 27
 Caractères ASCII, 310
 Caractères des fichiers de texte, 113, 114, 120, 212, 229
 addition, 120, 232
 annexion, 224
 insertion, 224
 recherche, 226
 suppression, 225
 suppression (Editeur de texte), 120, 232
 Carré, fonction, 51, 185
 Catalogue 1, 98, 99, 100, 171, 196, 284
 recherche, 299, 303
 Catalogue 2, 171, 248, 394, 399
 alarmes, 251
 fonctions horaires, 401
 fonctions étendues, 401
 recherche, 299, 303
 Catalogue 3, 171, 399
 recherche, 303
 Catalogue 4, 125, 171, 206, 400
 Catalogue 5, 71-73, 171, 196, 255, 400
 exécution pas à pas, 71
 Catalogue 6, 48, 168, 196, 400
 Catalogue des fonctions standard. *Cf.* Catalogue, 3
 Catalogues, 170
 comparaison CX et C/CV, 400
 consommation d'énergie, 170, 400
 fonctionnement, 400
 CERCLE (programme), 96-97, 99
 Chaînes alphanumériques, 26, 86, 117, 159-160, 197, 213, 309, 311, 366
 dans des programmes, 93, 282
 recherche des, 226
 saisie, 27
 Changement de signe, 18, 159, 185
 Chargement de programmes. *Cf.* Programmes, introduction
 Chiffres alphabétiques, 26, 200, 309
 calculs sur, 200
 effacement, 20
 groupes de, 35, 161

- séparation de, 35, 161
 - touches de saisie, 18, 159
 - Chronomètre
 - affichage, 61, 238, 267, 270, 271
 - chargement de l'heure, 273
 - clavier, 75, 266-268-269
 - correction, 238
 - différences de temps, 78-79
 - déclenchement et arrêt, 75, 268, 273
 - effacement, 267
 - erreurs, 80
 - exemples, 77-78
 - format, 239
 - heure, 63, 237
 - heure, 63, 237
 - mode, 362
 - modes, 76
 - pointeurs, 270-274
 - programmation, 273
 - programme, 275
 - précision, 238, 374, 378
 - précision, 378
 - rappel de l'heure, 273
 - registres, 270, 272
 - registres-pointeurs, 77, 79, 80
 - remise à zéro, 75, 267
 - réglage, 64, 238
 - temps intermédiaires, 76-77, 271
 - Chronomètre en décompteur, 274
 - Chronomètre et capacité mémoire, 78
 - Chronomètre sous alimentation faible, 383
 - Circuit (exemple), 55
 - Clavier
 - conventions d'utilisation, 15, 156
 - fonctionnement, 44
 - fonctions ne figurant pas sur le, 44, 156
 - mode, 362
 - Clavier alphabétique, 14, 15, 24, 155-156, 157, 159, 230
 - en mode exécution, 159
 - en mode programme, 159
 - indicateurs binaires, 291
 - Clavier normal, 14, 15, 155, 156
 - Clavier par défaut, 156
 - Clavier personnel, 14, 15, 46-47, 88, 155, 156, 166, 198
 - affectations, 166, 168
 - annulation des affectations, 168, 171
 - catalogue. Cf. Catalogue 6
 - indicateur binaire, 290
 - labels globaux, 211
 - priorités, 169
 - touches, 156
 - Claviers du HP-41 CX, 158
 - Compactage de la mémoire, 85, 89, 196, 198, 281
 - Comparaison de X à Y, 304
 - Comparaison de X à Y, indirecte, 305
 - Comparaison de X à zéro, 304
 - Comparaison de données alphabétiques, 304-305
 - Comparaison (fonctions de), 303-304
 - Compatibilité HP-41CX et HP-41C/CV, 398
 - Configurations pour les modules de mémoire annexe, 370
 - Consommation d'énergie électrique, 381, 382
 - Constantes, 180
 - calculs avec 24, 177-178, 180
 - Conversion d'unités d'angles, 53, 187
 - Conversion de base, 187
 - Conversion des coordonnées, 54, 189
 - Conversion décimal-octal, 187
 - Conversion octal-décimal, 187
 - Coordonnées rectangulaires, 54, 189
 - Copie de programmes des modules d'application, 107-108, 281
 - Correction d'erreurs, 16
 - dans les calculs, 179
 - sur l'affichage, 159
 - Correction du chronomètre, 238, 374, 378
 - Cosinus, 54
 - Croissance cumulée (calcul de la), 177
 - Cube (élévation de x au), 183
 - Curseur, 118, 229, 230, 232
 - commande du, 120, 232
-
- D**

 - Date
 - format, 242
 - indicateurs binaires du format, 291
 - Dates
 - addition, 66, 244
 - différence, 67, 244
 - mise, 62, 242
 - rappel, 66, 242
 - valides, 242, 243, 245
 - DEG
 - conversion, 53, 187
 - mode, 53, 186
 - Degrés décimaux, 53, 186
 - Degrés-minutes-secondes, 53, 65, 187
 - Différence entre jours, 67, 244
 - Division, 51
 - Données. Cf. aussi Fichiers de données, Registres
 - entrée de (Cf. aussi saisie), 95
 - indicateur de saisie, 290
 - manipulation, 402

- registres d'un fichier de, 213
 - saisie, 92
 - sortie de, 92, 95, 96
 - transfert entre des registres et un fichier, 218
 - Défilement de l'affichage, 28, 162
 - Dépassement de capacité, 24, 42, 56
 - Dépassement de capacité, 24, 42
-
- E**
- Ecart-type, 58, 192
 - Eclipse solaire (exemple), 67
 - Editeur de texte, 117 et suiv.
 - affichage, 118, 229, 230
 - clavier, 119-120, 231
 - délai d'invalidation, 230
 - invalidation, 117, 230
 - invalidation automatique, 230
 - témoins, 117, 230
 - validation, 117, 230
 - Effacement. *Cf. aussi* Suppression
 - d'alarmes, 258-259, 261
 - d'indicateur binaire, 288
 - de l'affichage, 19, 20, 93, 159, 160, 319
 - de l'affichage alphabétique, 26
 - de la pile, 183
 - de programmes, 102-103, 286-287
 - de temps au chronomètre, 75, 267
 - des alarmes, 258, 259, 261
 - des fonctions affectées au clavier personnel, 47, 168
 - des registres de données, 38, 202
 - des registres statistiques, 56, 190
 - du registre alphabétique, 160
 - En-tête (fichier de texte), 113, 205, 208, 212., 214
 - END (instruction), 89, 98, 281, 284
 - Enregistrement, 113-114, 121, 212, 214, 222, 229
 - annexion, 222
 - insertion, 223
 - insertion (éditeur de texte), 121, 232
 - longueur maximale, 230, 233
 - nombre d', 229
 - pointeur d', 113-114, 115, 213-216, 214, 228-230
 - rappel d', 226
 - recherche (éditeur de texte), 233
 - suppression, 224
 - suppression (éditeur de texte), 121, 233
 - Entière, partie entière d'un nombre, 186
 - Erreur(s)
 - affichage des, 347
 - avec fonctions numériques, 185
 - avec éditeur de texte, 233
 - correction en calcul, 179
 - d'horloge, 245
 - effacement de messages, 20
 - fichier, 127
 - indicateur binaire de non-arrêt, 290, 306
 - messages, 171, 354
 - non prise en compte, 290
 - programme, 98
 - situations, 171
 - Exponentielle, 51
 - Exposants, 159, 161
 - dans lignes de programme et listages, 19, 32
 - utilisation des, 18
 - Exécution alphabétique, 44-45, 282
 - Exécution automatique, 289
 - Exécution de programme. *Cf. aussi* Programme, Sous-programme, Fonctions
 - après sous-programme, 301
 - automatique, 282
 - avec clavier utilisateur, 282
 - avec imprimante, 368
 - pas à pas, 91, 282
 - arrêt, 368
 - répétition, 90
 - témoin, 90, 161, 282
 - Exécution des fonctions, 17, 44-45
 - Exécution, mode, 15, 83, 155, 282
-
- F**
- Facteur d'exactitude
 - ajustement, 376
 - formule, 377
 - rappel, 376
 - Factorielle, 51, 185
 - Fichier. *Cf. aussi* Fichiers, Données
 - catalogue. *Cf.* Catalogue 4
 - définition du nom, 207
 - déplacement du pointeur, 115
 - détermination de l'emplacement du pointeur, 216
 - détermination de la taille, 208
 - détermination du type, 207
 - en-tête. *Cf.* En-tête
 - erreurs, 127
 - mémoire. *Cf.* Mémoire, fichiers en nom, 116, 205, 206
 - modification de la taille, 213
 - pointeur, 115, 126, 206, 213, 214, 217
 - positionnement du pointeur, 215
 - taille, 206
 - Fichier de données, 123-124, 205, 215, 216
 - copie dans un, 123, 217, 218, 219, 220

création, 211
 effacement, 213
 nom, 211
 rappel de données d'un, 123-124, 218, 219, 220
 Fichier de programme, 124-125, 205, 208-211
 création, 124, 208
 nom, 208
 rappel, 124, 209
 recherche, 124, 209
 sauvegarde, 124, 208
 Fichier de texte, 113 et suiv., 205, 215, 216, 222, 228
 caractères, 113, 114, 120, 212, 229
 création, 116, 212
 effacement, 117, 213
 enregistrement. *Cf.* Enregistrements
 modification de la taille, 116
 nom, 212
 pointeur. *Cf.* Fichier, Enregistrement.
 rappel à partir d'un, 226
 sauvegarde en mémoire de masse, 227
 élimination, 117
 Fichiers. *Cf. aussi* Fichier, 205-206
 affectation de registres à des, 212
 besoins en mémoire des, 205, 212
 création, 211
 effacement, 213
 modification de l'affectation des registres, 213
 modification de la taille, 213
 rappel depuis la mémoire de masse, 227
 recherche d'une chaîne alphabétique, 226
 registres dans les, 206, 208
 sauvegarde en mémoire de masse, 227
 spécification, 206
 types de, 113, 205
 suppression, 208
 Fin d'une saisie, 17, 18
 Fonctions à deux nombres (diadiques), 51, 187
 Fonctions à un nombre (monadique), 17, 30, 184
 Fonctions conditionnelles, 303
 Fonctions conditionnelles étendues, 402
 Fonctions de commande de boucles, 306
 Fonctions de comparaison, 303-304
 Fonctions du clavier personnel, 46-47
 Fonctions du clavier personnel, affectation, 46, 156
 Fonctions du clavier personnel, annulation, 47
 Fonctions du clavier personnel, catalogue. *Cf.* Catalogue 6
 Fonctions du clavier personnel, exécution, 47
 Fonctions du clavier personnel, visualisation, 48
 Fonctions exponentielles, 187
 Fonctions externes et lignes de programmes, 396
 Fonctions externes, catalogue. *Cf.* Catalogue 2
 Fonctions externes, mémoire de programmation pour, 397
 Fonctions externes, temps d'exécution, 397

Fonctions ne figurant pas sur le clavier, 44, 156
 Fonctions non programmables, 283
 Fonctions paramétrables, 20, 30, 38
 Fonctions paramétrables, affichage des, 31
 Fonctions personnalisées, 109
 Fonctions primaires, 14, 15, 16
 Fonctions secondaires, 14, 15
 Fonctions temporelles, catalogue, 394
 Format de la date, 62, 242, 291
 Formats angulaires, 53, 186
 Formats de l'affichage, 31, 160, 291
 Formats de l'heure, 61, 239
 Formule
 de l'écart-type, 192
 de la moyenne, 192

G

GRAD, 53, 160, 186
 Grades (mode), 53, 186

H

HP-IL (Boucle d'interface Hewlett-Packard), 394
 HP-41CX et HP-41C/CV, 400-401
 addition et soustraction, 65, 188
 affichage, 238
 conventions, 61, 239
 conversion, 187
 correction, 238
 dérive, 375
 erreurs, 245
 fonctions, 399
 mise à, 63, 237
 module, 399
 précision, 238, 374
 précision, 377
 rappel, 64, 240
 remise à, 64, 238
 valeurs, 237
 validité, 245
 Horner (méthode de), 177

I

- Indicateur binaire (i.b.), 289
 - armement, désarmement, 35, 288
 - d'alimentation électrique faible, 292
 - de commande, 288-289
 - de commande de dispositif externe, 289
 - de commande de programme, 288
 - de ponctuation (affichage), 290
 - du système, 291
 - utilisateur, 288-289
 - conversion en un nombre, 292, 295
 - copie multiple de, 295
 - groupe de, 295
 - manipulation de, 402
 - restitution de l'état, 292, 296
 - sauvegarde de l'état de, 295
 - sauvegarde de l'état de, 292, 295-296
 - test de, 303-304
 - test et désarmement, 304
 - types, 288
 - état à réinitialisation et mise en marche, 293
 - états représentés par un multiplet, 294
 - test, 288, 298, 304
- Imprimante et programme, 368
- Imprimante
 - avance du papier, 368
 - indicateur binaire d'autorisation, 289
 - indicateur binaire de présence, 292
 - listage avec heure et date, 369
- Indicateur de saisie, 18, 30, 38, 95, 158, 314
- Indicateurs binaires (i.b.), 289
- Indicateurs de saisie, 95, 314
- Initialisation des programmes, 108
- Insertion de lignes de programme, 102, 286
- Insertion, mode (Editeur de texte), 121, 230, 232
- Instruction END, 89, 98, 281
- Intermédiaires, résultats, 20
- Intermédiaires, statistiques, 191
- Inverse
 - fonction, 51, 179, 185
 - fonction inverse de cosinus, 54
 - fonction inverse de sinus, 54
 - fonction inverse de tangente, 54
 - notation polonaise, 16, 174

J

- Jeu de caractères alphabétiques, 14, 24, 232
- Jour de la semaine, 67, 244
- Jour de la semaine, 67, 244

L

- Label, 297-303
- Label global, 86-88, 248, 282, 299
 - affecté automatiquement au clavier utilisateur, 211
 - affecté (déplacement vers), 285
 - et alarmes, 251
 - multiple, 284
 - affichage, 98, 284
 - branchement, 299
 - déplacement vers, 100, 283-284-285
 - insertion, 284
 - manquant, 100, 284
 - recherche, 299, 303
- Label numérique, 88, 299, 317
 - branchement à, 299
 - courte-distance, 299
 - déplacement à, 285
 - longue-distance, 299
- Labels alphabétiques, 169
- Labels alphabétiques locaux, 88, 169, 300
- Labels locaux, 87, 88, 299, 300
- Labels locaux, recherche, 300, 301
- LAST X, registre, 23, 175, 179, 180, 181, 185
- Ligne de programme, 84, 85, 197, 282
 - besoins en mémoire, 197
 - insertion, 102, 286
 - recherche, 100, 283
 - suppression, 100-101, 285
 - numéro de, 282
- Limites, 24, 42
- Limites (résultat hors-limites), 24, 90
- Logarithme naturel, 51, 187
- Logarithme népérien, 51, 187
- Longueur d'une chaîne alphabétique, 313

M

- Manuel, organisation du, 9
- Message, 158
- Message dans un programme, 94, 282, 308, 314, 318-319
 - affichage, 30, 34, 161
 - création, effacement, 94, 95
 - indicateur binaire, 292
- Messages d'état, 354
- Mise au point d'un programme, 91
- Mise sous tension et hors tension, 14, 155
- Mode programme, 155
- Mode rappel, 272
- Mode remplacement (éditeur de texte), 121, 230, 232
- Modes angulaires, 53, 186, 291

Modes de fonctionnement, 15
 Modules ROM, 393, 397
 Modules d'application, 83, 393
 Modules de mémoire annexe, 370-371
 Modules de mémoire annexe, mise en place et retrait, 371
 Modules manquants, 395
 Modulo, 190
 Moyenne, 58, 192
 Multiplets

- caractères alphabétiques, 309
- compte, 98
- dans registre ALPHA, 309
- dans un programme, 211
- disponibles dans fichier de texte, 222
- état d'indicateur binaire, 294
- fin de fichier, 212
- nuls, 309
- nuls dans un programme, 197, 198
- représentant des nombres, 294, 309
- utilisés dans sous-programmes, 303
- utilisés en branchement, 300
- utilisés pour labels, 299
- utilisés pour lignes de programme, 197

 Multiplication, 51
 Mémoire annexe

- disponible, 125
- fichier en. *Cf. aussi* Fichiers, 205
- fonctions comparées du HP-41CX et du HP-41C/CV, 401-402
- registres disponibles en, 205-208, 211
- répertoire. *Cf. aussi* Catalogue 4, 125

 Mémoire de masse

- copie de fichiers de la, 227
- sauvegarde de fichiers dans la, 227

 Mémoire disponible

- pour les fichiers, 125
- pour les programmes, 89, 281

 Mémoire morte (ROM), 281
 Mémoire permanente, 14, 28-29, 155
 Mémoire permanente, réinitialisation, 29, 382
 Mémoire principale, 36, 194

- disponible pour les registres de données, 199
- affectation de la, 194-196, 199, 281
- alarmes dans, 196, 198
- configuration par défaut, 196
- programmes dans la, 196, 197
- redéfinition des touches dans la, 196, 198

 Mémoire

- affectation de la, 36
- compactage, 85, 89, 196, 198, 281
- comparaison HP-41CX et HP-41C/CV, 399
- fichiers de texte en, 114
- fichiers en, 113

pile opérationnelle. *Cf.* Pile opérationnelle
 programmes en, 84, 85, 99
 répartition, 399

 N

Nombres, 159

- alphabétiques (recherche), 309, 311
- négatifs, 18, 159
- saisie de, 159

 Noms alphabétiques, 44

- de programmes, 282

 Notation ingénieur, affichage, 33, 161
 Notation polonaise inverse, 16, 174
 Notation scientifique, affichage, 32, 161
 NULL, 48, 169
 Nuls, 311, 366
 Numérique

- clavier (Editeur de texte), 118, 230, 232
- saisie, 159
- affichages, 31, 160
- erreurs dans les fonctions, 185
- spécification des paramètres, 162, 165-166

 Numéros de lignes de programme, spécification, 166

 O

Octal-décimal (conversion), 187
 On, indicateur binaire, 291
 Opérations non cummutatives, 22, 176, 180, 181
 Ordinateur, vérification du fonctionnement, 385

 P

Paramètres

- fonctions avec indirects, 164
- spécification, 30, 158, 162, 200
- spécification indirecte, 162

 Paramètres, spécification par touches spéciales, 165-166
 Partielle, séquence partielle de touches, 31

- exécution des programmes, 91, 282
- visualisation des programmes, 91, 100, 284

 Permanente (instruction END), 86, 89, 98, 196, 281
 Permanentes (valeurs), 155
 Permutation de x et y, 181
 Permutation des registres de la pile opérationnelle, 181

- Perturbations, radio et télévision, 391
- Pi, 19, 159
- Pile opérationnelle
- adressage des registres, 166
 - arithmétique directe dans les registres, 182
 - autorisation des montées, 176
 - calculs dans, 180
 - descente dans la, 175
 - effacement, 183
 - fonctions neutres, 176
 - inhibition des montées, 176
 - montée dans la, 175
 - opérations avec fonctions numériques, 184
 - permutation circulaire, 181
 - permutation des registres, 181
 - permutation des registres, 181, 182
 - registres, 20, 21, 175, 180
 - remplissage, 177
- Piles. *Cf.* Batteries
- Pointeur. *Cf.* Fichier, Programme, Enregistrement/caractère
- Polaires, coordonnées, 54, 189
- Polynomiales, calcul des expressions, 177
- Ponctuation du texte, 229
- Position d'un programme en mémoire, 283-285
- Position d'une chaîne dans fichier de texte, 226
- Position d'une chaîne dans registre ALPHA, 312-313
- Pourcentage, 51, 52, 188
- différence relative en, 51, 52, 188
 - rapport de deux nombres en, 189
- PRGM, 83, 90, 155, 282, 316
- Programme SURFACE, 86, 91, 93, 96, 99
- Programme CERCLE, 96-97, 99
- Programme de saisie de données, 92
- Programme(s). *Cf. aussi* Exécution, Fichier, Ligne
- affichage de tous les
 - affichage des résultats, 93
 - affichage des résultats d'un, 369
 - arrêt, 93
 - branchement, 298
 - catalogue. *Cf.* Catalogue 1, 2, 3, 4, 5, 6
 - comment le relancer, 93
 - compatibilité avec le HP-41C/CV, 398
 - copie à partir de modules d'application, 107-108, 281
 - copie dans un fichier, 124
 - correction de, 91, 98
 - dans un module, 395
 - délimitation, 87
 - déplacement au début d'un, 285
 - déplacement sur un, 100, 283-285
 - détermination de la taille, 211
 - effacement de, 102-103, 286-287
 - erreurs de, 98
 - exécution de, 90, 282
 - exécution automatique de, 289
 - exécution d'un programme rappelé, 209, 210
 - interaction de l'utilisateur, 317
 - interruption, 92, 93, 319
 - introduction, 89, 280
 - introduction de données, 95
 - messages, 94, 282, 308, 314, 318, 319
 - messages de demande de saisie, 95
 - mise au point, 91, 98
 - mode, 15, 83, 87, 285
 - modification, 91, 98, 285-286
 - mémoire. *Cf.* Mémoire, programmes en nom de, 86
 - nom manquant, 100, 284
 - opérations sur les fichiers de texte dans un, 222
 - pause, 93, 316, 319
 - pointeur de, 85, 100, 283-285
 - prévisualisation de, 170
 - sauvegarde dans un fichier, 124, 208
 - sortie de données de, 92, 95, 96
 - stockage de, 89, 280
 - suppression de, 102-103, 286-287
 - SURFACE (programme), 86, 91, 93, 96, 99
 - utilisation de l'éditeur de texte, 233
 - visualisation pas à pas de, 91, 100, 284
- Programmes d'application, 393
- exécution, 83
- Programmes des modules d'application, 107-108, 281
- Prévisualisation des fonctions, 48, 169
- Puissance de 10, 51
- Puissance, fonction, 51, 52, 189
- Périphériques, description des, 392

R

- ROM externes (XROMS), 394
- RAC, programme, 53, 160, 186
- Racine carrée
- fonction, 51, 185
 - calcul des, 190
- Radians, 53, 186
- conversion de, 53, 187
 - mode, 53, 186
- Rappel d'alarme, 252
- Rappel de caractères alphabétiques, 200
- Rappel de nombres, 37, 123, 182, 200, 218, 219, 220
- Rappel (mode), 272
- Rectangulaires (coordonnées), 54, 189
- Redéfinition de touches, 46, 156, 166
- Registre LAST X, 23, 175, 179, 180, 181, 185
- Registre T, 20, 175, 179, 180
- Registre X, 20, 30, 40, 158-159, 175, 179, 180

Registre X, rappel dans, 179, 182
 Registre X, stockage à partir de, 182
 Registre X, échange avec Y, 39
 Registre X, échange avec état indicateur binaire, 292, 295
 Registre X, échange du contenu, 201
 Registre Y, 20, 175, 179-180
 Registre Y, échange avec X, 39
 Registre Z, 20, 175, 179-180
 Registre alphabétique, 27, 158, 206, 222, 226, 312
 adjonction d'heure, 240, 241
 adjonction de date, 243
 affichage, 318
 affichage dans un programme, 94
 annexion à, 226
 capacité, 27, 159
 caractères nuls, 366
 copie dans, 226
 copie dans X, 309
 décalage dans, 200
 effacement, 160
 manipulation de données, 308
 messages d'alarme et, 68
 permutation dans, 200, 313
 rappel, 96
 recherche de chaîne, 312-313
 recherche de chiffres, 309, 311
 rotation, 200, 313
 Registre alphabétique et alarmes, 250, 251
 Registre(s). *Cf. aussi*. Pile, LAST X, alpha(bétique)
 accès à des blocs de, 218, 219
 affectation, 199, 281
 affichage du contenu, 30, 319
 arithmétique directe, 40-42, 201
 copie dans un fichier, 123, 217, 220
 copie des données dans un fichier, 218
 copie du contenu de blocs, 201
 dans un fichier, 206, 208, 213
 effacement des données, 38, 202
 fichier, 212, 214
 modification de l'affectation, 199
 permutation du contenu, 39-40, 201
 permutation du contenu de blocs, 201
 pointeur, 213, 215, 216
 rappel de données d'un, 220
 spécification, 164
 spécification de l'adresse, 200
 de stockage des données, 36, 194
 disponibles pour des données, 199
 disponibles pour des fichiers, 125, 206-208, 211
 disponibles pour les programmes, 89, 281
 non affectés, 36, 194, 196
 statistiques. *Cf.* Statistiques
 supérieurs à R99, 199
 vérification de l'affectation, 199

Remise à zéro du chronomètre, 75, 267
 Remplacement, mode (éditeur de texte), 121, 230, 232
 Répertoire
 des fichiers. *Cf.* Catalogue 4.
 de la mémoire annexe. *Cf.* Catalogue 4.
 des affectations du clavier personnel. *Cf.* Catalogue 6.
 des alarmes. *Cf.* Catalogue 5.
 des fonctions externes. *Cf.* Catalogue 2.
 des fonctions standard. *Cf.* Catalogue 3.
 des programmes. *Cf.* Catalogue 1.
 Reste, 190
 Répétitive, alarme, 255, 258, 259
 Résultat hors-limites, 24, 290
 Résultats intermédiaires, 20
 Rigil Centaurus (exemple), 24
 ROM (read-only memory), 281

5

Sous-programme(s), 301-302
 et codes des touches, 317
 appel de, 301, 317
 fin de, 301
 multiplats requis, 303
 pile de retour, 302
 rappel de programme en tant que, 209, 210
 retour de, 301-302
 Saisie (cf. aussi entrée), 95
 alphabétique, 159
 inversée, 17
 indicateurs de, 95, 314
 Sauvegarde de données dans un fichier, 123, 217, 218, 220
 Scientifique, affichage en notation, 32, 161
 Secondaires (fonctions), 14, 15, 156
 SECRETS (fichier), 114, 116, 121
 SETALM (programme), 261
 SHIFT (touche), 15-16, 156, 230
 Signe d'un nombre, 186
 Sinus, 54
 Sommation de données, 55-56, 191
 correction, 57, 191
 Soustraction, 51
 SPLITS, programme, 276
 Statistiques intermédiaires, 191
 Statistiques
 affectation des registres, 56, 190
 correction des données, 57, 191
 effacement des registres, 56, 190
 emplacement des registres, 56, 190
 limites des registres, 56-57, 192
 registres, 55, 190
 somme de données, 55-56, 191

Stockage de caractères alphabétiques, 200
 Stockage de nombres, 37, 123, 182, 200, 217, 218, 220
 Suppression. *Cf. aussi*. Effacement
 de caractères après annexion, 367
 de lignes de programme, 100-101, 285
 d'un fichier, 208
 Séparateur décimal, 35, 160, 161, 290

T

T, en exposant, 93, 118, 121, 159, 171, 229, 282, 395
 Taille de la mémoire principale, 199
 Tangente, 54
 Temps intermédiaire(s), 78-79, 270-272
 erreur, 273
 impression, 275
 exemple, 79
 mode, 270-272
 négatif, 273
 stockage et rappel, 272
 Températures, spécifications, 391
 Terminologie, comparaison HP-41CX/HP-41C/CV, 403
 Test conditionnel, 104
 Tonalité(s), 94, 260, 319
 indicateurs binaires de, 290
 Touche(s)
 de commande (Editeur de texte), 120
 fonctionnant en bascule, 14, 155
 code numérique des, 46, 166
 prise en compte enchaînée, 159
 précision des pressions sur, 375
 préfixe. *Cf. aussi* Fonctions paramétrables, 20
 programme exécuté par pression sur, 317
 redéfinition de, 46, 156, 166
 représentation des, 2, 16
 Trigonométrie, 53-54
 Témoins, 34, 160

U

Unité d'angle, indicateur binaire, 291
 USER, 47, 155

V

Valeur absolue, 186
 Valeurs permanentes, 155
 VECTEUR, programme, 109 et suiv.

Z

Zéros de queue, 366
 Zéros de tête, 366

Index des fonctions

Les fonctions ont chacune un nom alphanumérique (en bleu) et un nom au clavier (en noir ou en jaune); certaines n'ont pas les deux à la fois. (Voir les conventions à l'intérieur de la page de couverture.)

Chaque fonction a au plus trois références de pages. La première en *italique>* concerne le volume 1. La seconde concerne le volume 2. La troisième en **gras** concerne les tableaux de fonctions situés en fin du volume 2.

Fonction	Pages	Fonction	Pages	Fonction	Pages
	19, 158, 418	ATIME24	241, 429	D-R	53, 187, 423
	27, 159, 436	AYOK	311, 437	DATE	66, 242, 429
	51, 188, 423	AVIEW ((VIEW))	94, 318, 437	DATE+	66, 244, 429
	51, 188, 423	BEEP ((BEEP))	94, 319, 438	DDAYS	67, 244, 430
	51, 188, 423	BST ((BST))	91, 284, 431	DEC	187, 423
	51, 188, 423	CAT ((CATALOG)) <i>n</i>	170, 416	DEG	53, 186, 416
	51, 185, 423	CF ((CF)) <i>nn</i>	35, 288, 416	DEL	101, 286, 432
	51, 187, 423	CHS ((CHS))	18, 185, 423	DELCHR	225, 426
ABS	186, 423	CLA ((CLA))	26, 159, 437	DELREC	224, 426
ACOS ((COS))	54, 186, 423	CLALMA	258, 429	DIRY	62, 242, 430
ADATE	243, 429	CLALMY	259, 429	DOW	67, 244, 430
ADV	368, 438	CLD	94, 318, 419	DSE <i>nn</i>	306, 432
ALENG	313, 436	CLFL	117, 213, 426	ED	117, 228, 426
ALMCAT	71, 255, 429	CLK12	61, 239, 429	EEX	18, 159,
ALMNOW	261, 429	CLK24	61, 239, 429	EMDIR	125, 206, 426
ALPHA	24, 155, 416	CLKEYS	47, 168, 416	EMDIRX	207, 426
ANUM	311, 436	CLKT	61, 239, 429	EMPRODM	208, 426
AOFF	159, 436	GLKTD	61, 239, 429	END	89, 301, 432
AOK	159, 436	CLOCK	61, 238, 429	ENG ((ENG)) <i>n</i>	33, 161, 416
APPCR	224, 425	CLP	102, 286, 431	ENTER# ((ENTER#))	17, 175, 420
APPREC	222, 425	CLRALMS	69, 258, 429	EX ((EX))	51, 187, 423
ARCL ((ARC)) <i>nn</i>	96, 200, 436	CLRG	38, 202, 420	EX-T	187, 423
ARCLREC	226, 425	CLRGX	38, 202, 420	FACT	51, 185, 423
AROT	313, 436	CLS ((LS))	56, 190, 420	FACT <i>nn</i>	304, 432
ASHF	200, 436	CLST	183, 420	FACT <i>nn</i>	304, 433
ASIN ((SIN))	54, 186, 423	CLX ((CX))	19, 159, 420	FIX ((FX)) <i>n</i>	31, 160, 417
ASN ((ANV)) <i>nom, touche</i>	46, 166, 416	COPY	107, 281, 431	FLSIZE	208, 426
ASROOM	222, 425	CORRECT	238, 429	FRG	186, 423
ASTD ((STD)) <i>nn</i>	200, 436	COS ((COS))	54, 186, 423	FRG	304, 433
ATAN ((TAN))	54, 186, 423	CRFLAS	116, 212, 426	FST ((ST)) <i>nn</i>	304, 433
ATIME	240, 429	CRFLD	123, 211, 426	FSTC <i>nn</i>	304, 433
				GETAG	227, 426

Fonction	Pages	Fonction	Pages	Fonction	Pages
GETREV	317, 438	PSEZ	199, 417	SST ((SST))	91, 284, 432
GETREYX	317, 438	PURWL	117, 208, 427	ST+ ((STO +)) nn	40, 201, 421
GETP	124, 209, 427	RI	181, 421	ST- ((STO -)) nn	40, 201, 422
GETR	123, 217, 427	R-R-0	53, 187, 424	STW ((STO X)) nn	40, 201, 422
GETREC	226, 427	R-P ((R+P))	54, 189, 424	STJ ((STO -)) nn	40, 201, 422
GETRKA	218, 427	RS	93, 166, 434	STO ((STO)) nn	37, 200, 422
GETSUM	209, 427	RAD	53, 186, 417	STOFLAG	296, 418
GETX	124, 220, 427	RCL ((RCL)) nn	37, 200, 421	STOP ((RS))	93, 302, 434
GRAD	53, 186, 417	RCLAP	376, 430	STOPSW	273, 430
GTO ((GTO)) label	100, 300, 433	RCLAL#	252, 430	SW	75, 266, 430
GTO □ nnn ou label	283, 432	RCLFLAG	296, 417	SWPT	274, 431
GTO □ □	89, 281, 432	RCLPT	216, 427	T+X	64, 238, 431
HMS	53, 187, 430	RCLPTA	216, 428	TAN ((TAN))	54, 186, 424
HMS+	53, 188, 430	RCLPW	273, 430	TIME	64, 240, 431
HMS-	53, 188, 430	RDN ((R))	181, 421	YONE n	319, 438
HR	53, 187, 430	REDMOVE	201, 421	USER	47, 155, 418
INSCHR	224, 427	REDSWAP	201, 421	VIEW ((VIEW)) nn	39, 319, 422
INSREC	222, 427	RESZFL	116, 213, 428	X+2 ((x ²))	51, 185, 424
INT	186, 423	RND	186, 424	X=0? ((x=0?))	304, 434
ISG ((ISG)) nn	306, 433	RTN ((RTN))	90, 301, 434	X≠0?	304, 434
LAST ((LAST*))	23, 179, 420	RUNSW	273, 430	X<0?	304, 434
LSL ((LBL)) label	88, 299, 433	SAVEAS	227, 428	X<-0?	304, 434
LN ((LN))	51, 187, 423	SAVEP	124, 208, 428	X<0?	304, 434
LN+X	187, 423	SAVER	123, 217, 428	X=Y? ((x=y?))	304, 434
LOG ((LOG))	51, 187, 424	SAVERX	218, 428	X≠Y?	304, 434
MDY	62, 242, 430	SAVERX	123, 220, 428	X<Y?	304, 434
MEAN	58, 192, 424	SCI ((SCI)) n	32, 161, 417	X<=Y? ((x≤y?))	304, 434
MOD	190, 424	SDIV	58, 192, 424	X>Y? ((x>y?))	304, 434
OC	187, 424	SEKPT	215, 428	X=NM?	305, 435
OFF	292, 433	SEKPTA	115, 215, 428	X≠NM?	305, 435
OH	155, 417	SETAF	376, 430	X<NM?	305, 435
ON	14, 155, 417	SETDATE	62, 242, 430	X<-NM?	305, 435
P-R ((P+R))	54, 189, 424	SETIME	63, 237, 430	X>NM?	305, 435
PRCA	198, 432	SETSW	273, 430	X>=NM?	305, 435
PASK	166, 417	SF ((SF)) nn	35, 288, 417	X>	40, 201, 422
PCLP	103, 286, 419	S+ ((S+))	55, 191, 424	X<>F	295, 422
% ((%))	51, 188, 424	S- ((S-))	57, 191, 424	X<>Y ((x≠y))	39, 181, 422
ΣCH	51, 188, 424	ΣREG nn	56, 190, 417	XEO ((XEO)) label	45, 301, 435
IN ((I))	19, 159, 424	ΣREG>	56, 190, 417	XTOA	309, 437
POS	312, 437	SIN ((SIN))	54, 186, 424	XVZALM	67, 250, 431
POSPL	226, 427	SIGN	186, 424	Y+X ((Y ^X))	51, 189, 424
PRGM	87, 155, 417	SIZE nnn	199, 417		
PROMPT	95, 314, 438	SIZE?	199, 418		
PSE	93, 315, 438	SOFT ((V))	51, 185, 424		

CHAPITRES

- 9: **Le clavier et l'affichage (page 154)**
 - 10: **La pile opérationnelle (page 174)**
 - 11: **Fonctions numériques (page 184)**
 - 12: **Mémoire principale (page 194)**
 - 13: **Mémoire annexe (page 204)**
 - 14: **L'éditeur de texte (page 228)**
 - 15: **Les fonctions d'horloge**
 - 16: **Alarmes**
 - 17: **Fonctionnement du chronomètre (page 266)**
 - 18: **Bases de programmation (page 280)**
 - 19: **Indicateurs binaires (page 288)**
 - 20: **Branchements (page 298)**
 - 21: **Opérations alpha et opérations interactives (page 308)**
 - 22: **Programmes d'enregistrement de temps (page 320)**
-
- A: Messages d'erreur et d'état (page 354)**
 - B: Pour en savoir davantage sur les alarmes périmées (page 360)**
 - C: Caractères nuls (page 366)**
 - D: Opérations d'impression (page 368)**
 - E: Modules d'extension mémoire (page 370)**
 - F: Spécifications de l'horloge (page 374)**
 - G: Piles, garantie et maintenance (page 380)**
 - H: Périphériques, extensions et HP-IL (page 392)**
 - I: Comparaison avec le HP-41C/CV (page 398)**
 - J: Codes-barres (page 404)**



Siège social
Parc d'activité du Bois Briard
Avenue du Lac
91040 Evry Cedex
Tél. (6) 077.83.83

Direction pour l'Europe
150, route du Nant d'Avril
1217 Meyrin 2
Genève
Tél. (022) 83.81.11