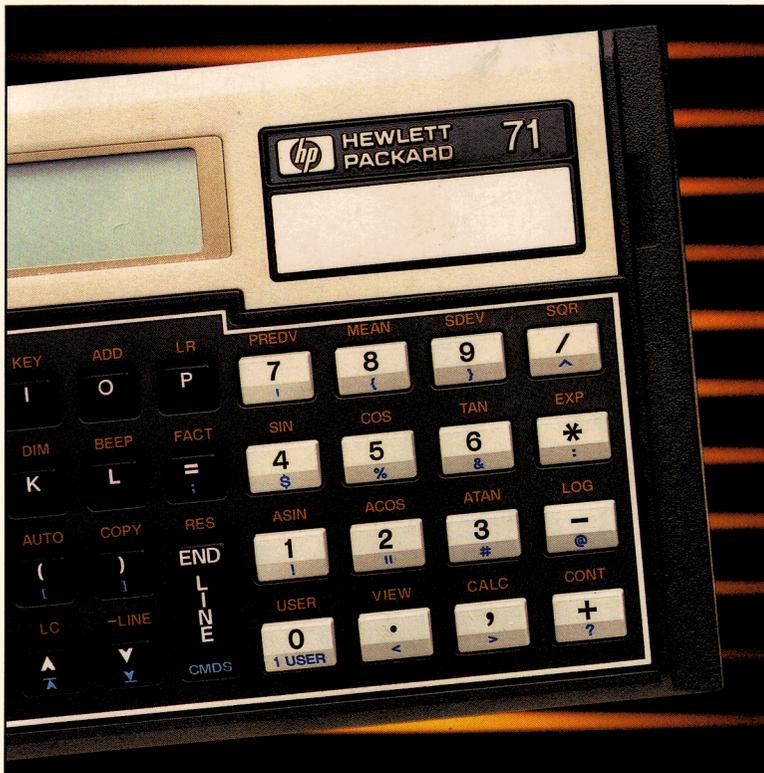

HP-71

Manuel d'utilisation



Notice

La société Hewlett-Packard n'offre aucune garantie en ce qui concerne les procédures décrites, le contenu de ce manuel ou l'application des programmes à une situation particulière. Les procédures et programmes sont proposés uniquement tels quels et l'utilisateur assume entièrement le risque de leur qualité et performance. Si les procédures ou programmes s'avèrent défectueux, l'utilisateur (et non la société Hewlett-Packard ni toute autre partie) est responsable de l'ensemble des coûts de correction et de tout dommages en résultant. La société Hewlett-Packard n'est pas responsable des dommages résultants ou connectés à l'utilisation de ces procédures et programmes.



HP-71

Manuel d'utilisation

December 1984

© Hewlett-Packard France, 1984
Texte protégé par la législation
en vigueur en matière de propriété
littéraire et dans tous les pays

Introduction à l'ordinateur portable HP-71

En achetant l'ordinateur portable HP-71, vous devenez propriétaire d'un instrument de calcul très performant, conçu comme organe central d'un système de calcul parfaitement portable. La solide conception et les hautes performances de cet ordinateur peuvent accroître votre productivité de façon importante.

Le HP-71 vous offre les caractéristiques suivantes:

- Petite taille pour permettre une portabilité maximale.
- Mode calcul pour effectuer des calculs complexes tout en contrôlant les résultats intermédiaires.
- Puissant jeu de fonctions, d'ordres et d'opérateurs BASIC (plus de 230 en tout). De nombreux ordinateurs de taille supérieure ne disposent pas d'un jeu aussi complet.
- Fonctions statistiques très performantes vous permettant d'effectuer des calculs sur 15 variables indépendantes.
- Sous-programmes et fonctions définies par l'utilisateur récursifs. Très appréciées dans d'autres langages de programmation, ces techniques augmentent la puissance de traitement du HP-71.
- Système perfectionné de fichiers internes pour le stockage de vos données et programmes.
- Mémoire permanente. Lorsque vous éteignez le HP-71, il conserve en mémoire les programmes et les données.
- Clavier personnalisable pour vos applications particulières.

Les extensions optionnelles du HP-71 comprennent des modules d'applications contenant des programmes pré-enregistrés prêts à fonctionner, un lecteur de cartes magnétiques pour le stockage et le rappel de données de façon économique et une interface HP-IL qui vous permet de connecter une imprimante, un lecteur de cassette magnétique, une interface vidéo, un modem et d'autres interfaces.

Table des matières

Utilisation de ce manuel	6
--------------------------------	---

Première partie: Opérations de base

Chapitre 1: Généralités	10
•Présentation •Clavier •Réinitialisation de la mémoire et mode BASIC	
•Fenêtre d'affichage •Mise à l'heure et date •Calculs au clavier	
•Entrée et exécution de programmes existants •Personnalisation du clavier	
•Contraste et inclinaison de l'affichage •Témoins d'état	
•Rappel des cinq dernières instructions-Pile d'instructions •Signaux sonores	
•Et ensuite! •Conventions de syntaxe	
Chapitre 2: Calculs avec le HP-71	36
•Présentation •Utilisation du mode CALCUL •Opérateurs arithmétiques	
•Fonctions numériques •Format des nombres •Précision numérique	
•Précision des variables numériques •Exceptions mathématiques	
•Intervalle des nombres •Opérateurs de relation •Opérateurs logiques	
•Hiérarchie des opérateurs.	
Chapitre 3: Variables simples et en tableaux	66
•Présentation •Caractéristiques des variables et des tableaux	
•Variables numériques: simples et en tableaux •Chaînes alphanumériques	
Chapitre 4: Fonctions statistiques	78
•Présentation •Déclarations de tableaux statistiques	
•Utilisation des opérations statistiques •Ajustement de valeurs à des courbes	
Chapitre 5: Horloge et calendrier	90
•Présentation •Calendrier du HP-71 •Horloge du HP-71	
Chapitre 6: Opérations sur les fichiers	98
•Présentation •Fichier courant •Fichier de travail: <code>workfile</code>	
•Introduction à l'utilisation des fichiers •Structure de la mémoire du HP-71	
•Noms de fichier •Noms d'unité •Copie de fichier	
•Changement de nom de fichier •Suppression de fichier •Protection de fichier	
•Catalogues de fichiers	

Chapitre 7: Personnalisation du HP-71	120
• Présentation • Redéfinition du clavier • Interactions programme/clavier	
• Jeu de caractères secondaire • Champs d'affichage protégés	
• Lecture de caractères à l'affichage • Affichage graphique	
• Restriction de l'utilisation du HP-71 • Exécution automatique d'instructions	
• Contrôle de l'affichage	

Deuxième partie: Programmation du HP-71

Chapitre 8: Rédaction et exécution de programmes	142
• Présentation • Entrée d'un programme • Exécution d'un programme	
• Arrêt d'un programme • Modifications d'un programme	
• Utilisation des fichiers BIN et LEX • Transformation de fichiers	
Chapitre 9: Conditions d'erreur	162
• Présentation • Types d'erreurs • Messages d'erreurs	
• Opérations de traitement d'erreurs • Contrôle des erreurs par programme	
• Messages d'avertissement • Exceptions mathématiques	
Chapitre 10: Branchement, boucle et exécution conditionnelle	178
• Présentation • Branchements inconditionnels • Branchements calculés	
• Branchements par l'horloge • Boucles • Exécution conditionnelle	
Chapitre 11: Indicateurs d'état	190
• Présentation • Introduction aux indicateurs • Tests des indicateurs	
• Armement et effacement des indicateurs • Indicateurs utilisateur	
• Indicateurs système	
Chapitre 12: Sous-programmes et fonctions définies par l'utilisateur ..	202
• Présentation • Sous-programmes • Fonctions définies par l'utilisateur	
Chapitre 13: Formats d'impression et d'affichage	224
• Présentation • Formatage simple • Formatage par chaîne de format	
• Contrôle de l'affichage et de l'imprimante •	
Chapitre 14: Stockage et rappel de données	240
• Présentation • Entrée de données au clavier • Données de programme	
• Fichiers de données • Accès séquentiel • Accès sélectif • Accès aux tableaux	
• Transfert de canaux à un sous-programme	

Annexes et index

Annexe A: Informations générales	266
Annexe B: Accessoires	282
Annexe C: Utilisation du lecteur de cartes magnétiques HP 82400A ..	284
Index des sujets	294
Index des mots-clés	Dos de couverture

Utilisation de ce manuel

Le HP-71 est un ordinateur portable possédant plus de fonctions, ordres et opérateurs que de nombreux ordinateurs de taille supérieure. De ce fait, sa documentation est importante et a été conçue de façon à vous permettre d'utiliser rapidement le HP-71 pour des applications scientifiques et commerciales.

La documentation du HP-71 comprend:

- Le *Manuel d'utilisation*.

Ce manuel présente l'utilisation du HP-71. Sa lecture suppose une connaissance de base de la programmation. Tous les utilisateurs doivent lire certaines parties de ce manuel, en particulier le chapitre 1, «Généralités».

- Le *Manuel de référence*.

Le *manuel de référence* contient la description complète de la syntaxe de chaque ordre, fonction et opérateur du HP-71. Lorsque vous saurez bien utiliser le HP-71, le *manuel de référence* sera votre source principale d'informations.

- L'*Aide-mémoire*.

Ce petit *aide-mémoire* que vous pouvez conserver dans l'étui de l'ordinateur résume toutes les informations importantes concernant l'ordinateur.

La documentation du HP-71 suppose que vous possédez déjà une connaissance de base du langage BASIC et que vous savez rédiger des programmes utilisant:

- Des variables et des tableaux.
- Des sous-routines.
- Des branchements, des boucles et des tests de conditions.
- Des instructions DATA.
- Des imprimantes.
- Des instructions d'entrée de données.
- Des commentaires sur les lignes de programme.

Si vous n'avez aucune expérience de programmation en BASIC et désirez néanmoins programmer votre HP-71, nous vous recommandons d'étudier d'abord le langage BASIC. Si vous voulez uniquement utiliser le HP-71 pour exécuter des programmes existants, vous n'avez pas besoin d'apprendre le BASIC. Le HP-71 est conçu de façon à vous permettre d'effectuer des calculs et d'exécuter des programmes sans vous inquiéter de la programmation.

Tous les utilisateurs doivent lire le chapitre 1, «*Généralités*», pour se familiariser avec le fonctionnement de l'ordinateur. Les autres chapitres présentent des sujets particuliers et leur lecture est donc optionnelle.

Si vous voulez apprendre à...	lisez les chapitres...
Exécuter un programme pré-enregistré.	1, 6, 8*
Effectuer des calculs au clavier.	1, 2
Effectuer des analyses statistiques.	1, 2, 4
Utiliser l'horloge et le calendrier interne.	1, 5
Personnaliser votre HP-71.	1, 7
Ecrire et exécuter des programmes.	1, 2, 3, 6, 8, 9, 10, 11
Utiliser des structures de programmations complexes.	12, 13, 14
Utiliser le lecteur de carte magnétique HP 82400A.	Annexe C
* Le chapitre 8 décrit comment rédiger et exécuter des programmes. Si vous voulez simplement exécuter des programmes existants, ne lisez que la partie du chapitre 8 traitant de l'exécution.	

Vous trouverez à la fin de ce manuel un index des sujets suivi d'un index des mots-clés.

Première Partie
Opérations de base

Généralités

Contents

Présentation	11
Clavier	11
Touches à exécution immédiate	12
Aides à la frappe	12
Conventions de représentation des séquences de touches	13
Mise sous et hors tension (ON, f OFF)	13
Touche END LINE	13
Ré-initialisation de la mémoire et mode BASIC	13
Fenêtre d'affichage	14
Déplacement de la fenêtre d'affichage (◀, ▶, g◀, g▶)	15
Effacement de l'affichage (ATTN)	16
Correction des erreurs de frappe (ATTN, f BACK)	16
Mise à l'heure et date	17
Calculs au clavier	18
Calculs en mode BASIC	19
Calculs en mode CALC	19
Entrée et exécution de programmes existants	21
Affichage d'une ligne de programme (▲, ▼, g▼, g▲)	21
Modification d'une ligne (f BACK, f CHAR, f I/R, f -LINE)	21
Nom de fichier de programme (EDIT)	21
Entrée, modification et exécution du programme DEPASSE	22
Contrôle de la vitesse d'affichage (DELAY)	26
Stockage du programme fichier DEPASSE (EDIT, NAME)	27
Exécution d'un programme quelconque en mémoire	27
Personnalisation du clavier (f USER, g 1USER)	28
Contraste et inclinaison de l'affichage (CONTRAST)	29
Témoins d'état	30
Rappel des cinq dernières instructions-Pile d'instructions (g CMDS)	31
Signaux sonores (BEEP)	32
Et ensuite!	33
Conventions de syntaxe	34

Présentation

Ce chapitre vous présente:

- Le clavier.
- L'affichage.
- L'effacement de la mémoire.
- L'horloge.
- Les calculs au clavier.
- L'entrée et l'exécution de programmes.
- La modification de programmes.
- La création de touches définies par l'utilisateur.
- L'utilisation de la pile d'instructions.
- L'utilisation des touches à exécution directe
- L'indicateur sonore.
- Le reste du manuel.

Clavier

La plupart des touches de votre HP-71 ont une fonction primaire et deux fonctions secondaires. La fonction primaire est indiquée par la mnémonique blanche ou noire imprimée sur le dessus de la touche. Les fonctions secondaires sont indiquées par les mnémoniques jaunes imprimées au dessus des touches et par les mnémoniques bleues imprimées sur la face inclinée des touches.

- Pour choisir le caractère ou l'opération imprimé sur le dessus d'une touche, appuyez simplement sur cette touche. Exemple: .
- Pour choisir le caractère secondaire ou la fonction imprimée en jaune et en bleu, appuyez sur la touche préfixe de même couleur ( ou ) et sur la touche. Par exemple:  ,  . Vous pouvez relâcher la touche préfixe avant d'appuyer sur la touche de fonction ou vous pouvez la maintenir enfoncée.
- Vous pouvez choisir d'affecter les majuscules ou les minuscules aux positions primaires des touches alphanumériques. la séquence   vous permet de passer d'une option à l'autre.
- Lorsque les majuscules sont en positions primaires des touches, la pression de  suivie d'une touche alphanumérique affiche la lettre minuscule correspondante. De même lorsque les minuscules sont en positions primaires, la pression de  suivi d'une touche alphanumérique affiche la lettre majuscule correspondante.

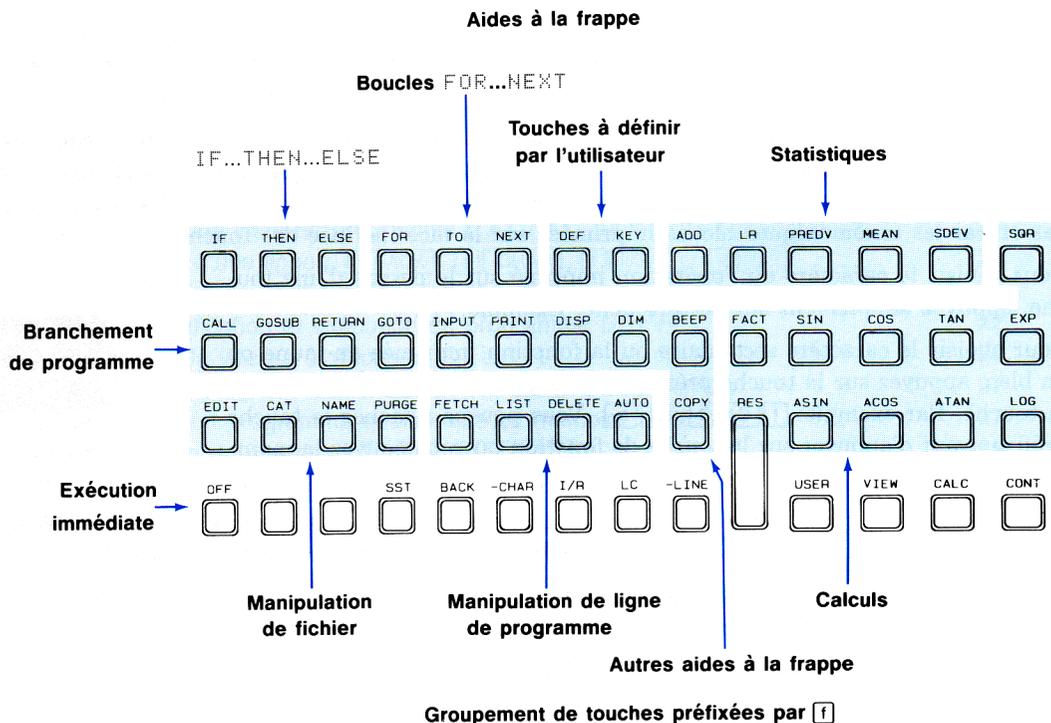


Touches à exécution immédiate

La plupart des touches du HP-71 affichent simplement des caractères lorsque vous les utilisez. Cependant, la plupart des touches du dernier rang sont des *touches à exécution immédiate*; elles exécutent une opération dès que vous appuyez sur la touche. Exemple: la pression de **f** **◀** exécute immédiatement un recul d'un caractère avec effacement. Cette opération vous permet de corriger facilement les erreurs de frappe. Vous trouverez une liste de toutes les touches (positions primaires et secondaires) dans l'index du manuel.

Aides à la frappe

Toutes les positions de touches précédées de **f** dans les trois premiers rangs sont des *aides à la frappe*. Elles affichent simplement des groupes de caractères couramment utilisés et permettent ainsi une frappe plus rapide. Il vous suffit alors d'appuyer sur une touche pour afficher le nom d'une fonction et vous êtes sûr de ne pas faire de faute de frappe. Exemple: pour frapper GOSUB avec un espace, vous pouvez appuyer sur **G** **O** **S** **U** **B** **SPC** ou bien plus simplement sur **f** **S**. Les mnémoniques imprimées en jaune au dessus des touches d'aide à la frappe représentent les caractères que ces touches affichent. Le diagramme ci-après illustre le groupement logique de ces aides à la frappe sur le clavier.



Conventions de représentation des séquences de touches

Tout au long de ce manuel, les séquences de touches sont représentées d'une des quatre façons suivantes:

1. Les séquences avec ou sans préfixe affichant des caractères sont représentées par ces caractères. Exemple: * signifie «appuyez sur la touche *» et ‡ signifie «appuyez sur [9][4]».
2. Les touches sans préfixe qui n'affichent pas de caractère sont représentées par les mnémoniques imprimées sur les touches. Exemple: ◀ signifie «appuyez sur la touche ◀».
3. Les séquences préfixées par [f] qui n'affichent pas de caractère sont représentées par [f] suivi des mnémoniques jaunes des touches. Exemple: [f][BACK] signifie «appuyez sur [f] et sur ◀».
4. Les séquences préfixées par [9] qui n'affichent pas de caractère sont représentées par [9] suivi des mnémoniques bleues des touches. Exemple: [9]▶ signifie «appuyez sur [9] et sur ▶».

Mises sous ou hors tension ([ON], [f][OFF])

La pression de [ON] allume le HP-71 et [f][OFF] l'éteint. Pour économiser la batterie, le HP-71 s'éteint automatiquement après 10 minutes d'inactivité.

Touche [END LINE]

La touche [END LINE] est similaire à la touche RETURN de nombreux autres ordinateurs. Lorsque vous appuyez sur [END LINE], l'ordinateur effectue l'une des opérations suivantes:

- Exécute l'instruction ou effectue le calcul que vous venez d'entrer.
- Stocke en mémoire les caractères que vous venez de frapper. Par exemple, lorsque vous entrez un programme en mémoire, vous devez appuyer sur [END LINE] après la frappe de chaque ligne à l'affichage.
- Détecte une erreur. Dans ce cas, l'ordinateur émet un bip et affiche un message.

Ré-initialisation de la mémoire et mode BASIC

Nous vous conseillons de ré-initialiser maintenant la mémoire de votre HP-71 de façon que les affichages des exemples de ce chapitre correspondent aux illustrations. Le HP-71 dispose de trois types de ré-initialisation: INIT: 1, INIT: 2 et INIT: 3. Vous pouvez utiliser maintenant le dernier car il efface la mémoire utilisateur (RAM). Vous trouverez une description complète des deux autres types de ré-initialisation dans l'annexe A. L'exemple suivant illustre la ré-initialisation de la mémoire.

Nota: Nous utiliserons souvent dans ce manuel le format de procédure utilisé ci-dessous pour indiquer en détail une série de séquences de touches et les affichages résultants. Les séquences suivent les conventions décrites à la page précédente. Les affichages résultants sont illustrés sous forme de caractères matriciés dans un cadre.

Entrée/Résultat

ON **/**

Appuyez et relâchez les deux touches en même temps.

```
INIT: 1
```

L'affichage montre l'ordre pour le premier type de ré-initialisation. Toutes les touches sont inactives sauf **1**, **2**, **3** et **END LINE**.

3 **END LINE**

Choisissez une ré-initialisation de type **3**.

```
Memory Lost
```

L'ordinateur vous indique que la mémoire est effacée.

ON

Efface l'affichage.

```
> █
```

Le symbole > est le symbole de mode BASIC. Vous effectuerez la plus grande partie de votre travail avec le HP-71 dans ce mode. Vous pouvez aussi utiliser le HP-71 dans un autre mode, CALCUL, que nous présenterons dans quelques pages. Le symbole █ clignotant est le curseur de remplacement, indiquant la position à laquelle le prochain caractère entré remplacera l'espace ou le caractère présent.

Fenêtre d'affichage

L'affichage de 22 caractères est une fenêtre ouverte sur une ligne de 96 caractères. Les séquences de touches suivantes illustrent la longueur de cette ligne et vous montrent les caractères et espaces affichés par certaines des touches d'aide à la frappe.

Entrée/Résultat

Appuyez sur **f** et, en maintenant **f** enfoncée, appuyez dans l'ordre sur **Q**, **W**, **E** et **R**.

```
>IF THEN ELSE FOR █
```

Le symbole BASIC, █, occupe la première position de la ligne de 96 caractères.

Appuyez et maintenez [f] puis appuyez sur les autres touches du premier rang, de gauche à droite, suivies de celles du second rang en vous arrêtant à [J]. Séquence: [f] et [T], [Y], [U], [I], [O], [P], [7], [8], [9], [/], [A], [S], [D], [F], [G], [H] et [J].

```
← ETURN GOTO INPUT PRIN█
```

Remarquez que la pression de [J] n'a pas semblé modifier l'affichage mais a produit un bip, indiquant que la ligne est pleine. La flèche gauche à l'extrémité gauche de la fenêtre indique qu'une partie de la ligne est en dehors de la fenêtre. Le curseur est maintenant en position 96.

Pourquoi la 96^{me} position est-elle vierge dans l'affichage ci-dessus? Qu'est-il advenu du T de PRINT? Lorsque vous essayez d'entrer plus de 96 caractères dans une ligne, le 97^{me} caractère et les suivants apparaissent en position 96, se remplaçant successivement tant que vous entrez des caractères. Dans l'exemple ci-dessus, le dernier caractère est l'espace final de DISP , aide à la frappe produite par [f][J].

Déplacement de la fenêtre d'affichage ([←], [→], [g][←], [g][→])

Les touches [←] et [→] vous permettent de déplacer la fenêtre le long de la ligne.

- [←] déplace le curseur vers la gauche d'un espace à la fois sans effacer de caractère. Si vous maintenez cette touche enfoncée pendant plus d'une demi-seconde, son action se répète.
- [→] déplace le curseur vers la droite. Son fonctionnement est similaire à celui de [←].
- [g][←] déplace immédiatement le curseur au début de la ligne.
- [g][→] déplace immédiatement le curseur sur l'espace suivant le dernier caractère de la ligne ou sur la position 96.

Entrée/Résultat

```
←←←←
```

Déplace le curseur de quatre positions vers la gauche.

```
← ETURN GOTO INPUT PRIN
```



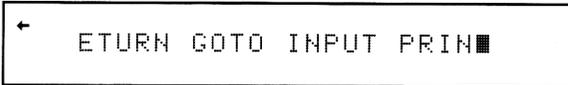
Déplace le curseur au premier caractère de la ligne.



La flèche à l'extrémité droite de l'affichage indique que la ligne continue à droite.



Déplace le curseur à l'extrémité droite de la ligne.



Effacement de l'affichage (**ATTN**)

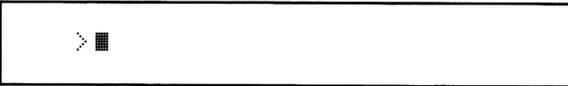
Lorsque le HP-71 est allumé, la touche **ON** devient la touche **ATTN** (ATTENTION). Cette touche a deux effets:

- S'il n'y a pas de programme en cours d'exécution, la touche **ATTN** efface l'affichage.
- Si un programme est en cours d'exécution, **ATTN** l'interrompt et affiche l'indicateur **SUSP**.

Entrée/Résultat



Efface l'affichage.



Lorsque l'affichage est effacé ou lorsque le curseur n'est pas affiché, vous pouvez toujours frapper une instruction, un calcul ou une ligne de programme et l'entrer dans le HP-71 (en appuyant sur **END LINE**). Si l'affichage contient des caractères mais pas de curseur, la première pression de touche efface l'affichage avant d'exécuter sa fonction (sauf avec les touches **←**, **→**, **9←** et **9→** qui n'ont aucun effet dans cette situation). Vous trouverez des exemples de ces effets tout au long de ce chapitre.

Correction des erreurs de frappe (**ATTN** , **f BACK**)

Vous disposez de deux touches pour corriger les erreurs de frappe:

- **ATTN** efface l'affichage si aucun programme n'est en cours d'exécution.
- **f BACK** recule le curseur d'une position en effaçant le caractère à la nouvelle position.

Mise à l'heure et date

Le HP-71 contient une horloge à quartz très précise et un calendrier couvrant plusieurs milliers d'années. L'horloge fonctionne en permanence dès que vous installez la batterie, que le HP-71 soit allumé ou éteint. Cette section du chapitre vous indique comment initialiser l'horloge à l'heure et à la date correcte.

L'exemple ci-dessous suppose que la date d'aujourd'hui est le 20 mai 1984 et qu'il est 16 heures 13 minutes et 10 secondes.

Le HP-71 utilise le format de date *année/mois/jour* (AA/MM/JJ) et requiert que vous entriez six chiffres, y compris les éventuels zéros d'en-tête.

Exemple: Spécifiez la date, 20 mai 1984.

Entrée/Résultat

SETDATE"84/05/20"

Définit la date.

SETDATE"84/05/20"■

END LINE

Entre la date.

>■

DATE# END LINE

Affiche la date.

84/05/20

Nous vous indiquerons plus loin la procédure permettant de mettre l'horloge à l'heure avec une précision de 1 seconde ou mieux. Lisez les instructions suivantes et mettez votre horloge à l'heure.

Entrez l'heure plus environ 30 secondes puis appuyez sur **END LINE** lorsque l'heure vraie correspond à celle affichée.

Il n'est pas nécessaire d'effacer l'affichage avant d'appuyer sur les touches du groupe suivant. Lorsque le curseur n'est pas affiché, la pression de touche suivante efface l'écran et affiche le caractère de cette touche.

Entrée/Résultat

```
SETTIME "16:14:00" END LINE
```

Cette instruction définit l'heure. L'horloge du HP-71 utilise le format d'heure sur 24 heures et requiert que vous entriez six chiffres y compris les éventuels zéros d'en-tête. Les deux zéros suivant le second symbole : correspondent aux secondes.



Supposons que la frappe de cette instruction `SETTIME` soit terminée à 16:13:30. Le HP-71 égrène maintenant les secondes en indiquant en-outre les demi-secondes. Appuyez sur END LINE lorsque l'horloge arrive à une demi-seconde avant 16:14:00 pour mettre le HP-71 à l'heure avec une précision de quelques dixièmes de secondes. Le chapitre 5 vous indique comment ajuster la mise à l'heure et la vitesse de l'horloge.

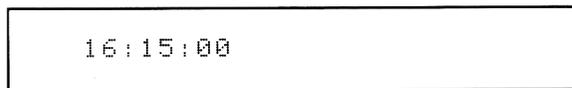
La fonction `TIME#` donne l'heure à l'instant de son exécution. Pour afficher l'heure avec l'égrènement des secondes, vous devez rédiger un petit programme tel que celui de la page 131 au chapitre 7.

Supposons que vous exécutiez `TIME#` exactement une minute après avoir exécuté `SETTIME`:

Entrée/Résultat

```
TIME# END LINE
```

Affiche l'heure sous forme d'une chaîne alphanumérique et non pas comme une valeur numérique.



Voici l'heure à laquelle `TIME#` a été exécuté.

Calculs au clavier

Vous pouvez effectuer des calculs au clavier de deux façons:

- *En mode BASIC*, le HP-71 effectue les calculs de la même façon que la plupart des ordinateurs à langage BASIC. Vous introduisez d'abord l'expression entière puis vous appuyez sur END LINE pour obtenir le résultat.
- *En mode CALCUL*, vous frappez aussi l'expression entière comme en mode BASIC. Néanmoins, dès que la portion déjà entrée de l'expression peut être évaluée, le HP-71 affiche automatiquement ce résultat intermédiaire. La pression de END LINE donne le résultat complet.

La possibilité de surveiller le déroulement d'un calcul en vérifiant les résultats intermédiaires est un avantage important:

- Vous pouvez vérifier que le déroulement a bien lieu comme vous le désirez. Ceci vous permet dans certains cas de déceler des erreurs qui seraient autrement très difficiles à trouver.

- Il est plus facile de comprendre ainsi comment se comporte une expression, ce qui est souvent plus utile que le résultat final.

Evaluons l'expression: $7 + 4 - 9 \times (15 - 7 \div 3)$. De façon que les résultats des calculs sur votre HP-71 apparaissent tels que ceux de ce manuel, exécutez l'instruction suivante:

Entrée/Résultat

FIX2 [END LINE]



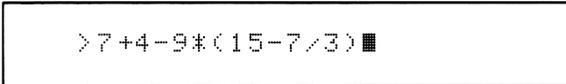
Le HP-71 affiche désormais les résultats arrondis à deux décimales.

Calculs en mode BASIC

Exemple: Évaluez une expression en mode BASIC comme suit.

Entrée/Résultat

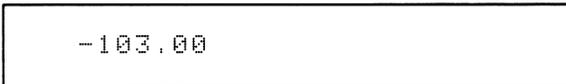
7+4-9*(15-7/3)



L'expression est entrée prête à être évaluée.

[END LINE]

Évalue l'expression.



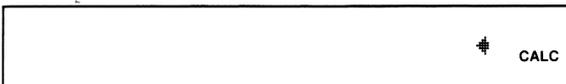
Le résultat.

Calculs en mode CALC

Exemple: Évaluez la même expression en mode CALCUL. Si vous faites une erreur, appuyez sur [f] [BACK] pour effacer tous les caractères de l'erreur et complétez l'expression correctement. Placez tout d'abord le HP-71 en mode CALCUL.

Entrée/Résultat

[f] [CALC]



Le témoin vous indique que vous êtes en mode CALCUL et le *curseur d'insertion* clignotant (page 21) indique que les caractères seront insérés à partir du bord droit de l'écran.

7+



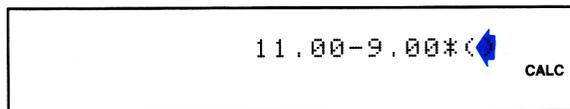
Dès que vous entrez un opérateur en mode CALCUL, le HP-71 accepte l'opérande le plus récent et l'affiche comme un résultat.

4-



Lorsque vous entrez -, le HP-71 non seulement entre le 4 mais évalue le début de l'expression et affiche le résultat intermédiaire.

9*(

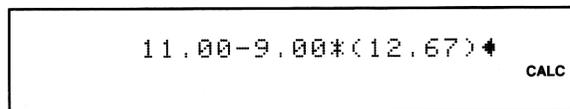


Cet affichage illustre deux caractéristiques du mode CALCUL:

- Le HP-71 n'évalue pas $11.00-9.00$ puisqu'il doit d'abord connaître le contenu des parenthèses suivantes de façon à respecter la hiérarchie des opérateurs (chapitre 2, page 64).
- Le curseur d'insertion partage sa position avec une parenthèse droite, pour vous rappeler que l'expression requiert une parenthèse droite pour toute parenthèse gauche ouverte.

Entrée/Résultat

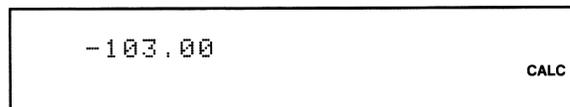
15-7/3)



Lorsque vous entrez la parenthèse droite, le symbole $)$ clignotant disparaît. Si une expression comporte plusieurs niveaux de parenthèses emboîtés, le HP-71 laisse la parenthèse clignotante jusqu'à ce que tous les niveaux soient fermés.

END LINE

Évalue et affiche le résultat final.



Entrée et exécution de programmes existants

Les quelques pages suivantes illustrent l'entrée dans l'ordinateur et l'exécution d'un programme dont vous possédez le listage. Nous décrivons d'abord quelques outils de correction d'erreur qui vous permettront d'afficher et de modifier des lignes de programme si nécessaire.

Affichage d'une ligne de programme (▲, ▼, ⓐ▲, ⓐ▼)

Lorsque vous entrez ou exécutez un programme, ces quatre séquences vous permettent d'afficher toute ligne du programme pour la modifier. Toute ligne affichée grâce à ces touches devient la *ligne courante*. L'effet de ces touches est illustré lors de la section sur l'entrée du programme.

- ▲ rappelle à l'affichage la ligne précédant la ligne courante. Si vous maintenez cette touche enfoncée pendant plus d'une demi-seconde, le HP-71 répète son action.
- ▼ rappelle à l'affichage la ligne suivant la ligne courante. Si vous maintenez cette touche enfoncée pendant plus d'une demi-seconde, le HP-71 répète son action.
- ⓐ▲ rappelle à l'affichage la ligne de plus petit numéro.
- ⓐ▼ rappelle à l'affichage la ligne de plus grand numéro.

Modification d'une ligne (f BACK, f -CHAR, f I/R, f -LINE)

Ces quatre séquences, utilisées en mode BASIC, vous permettent de modifier toute ligne affichée:

- f BACK recule le curseur d'une position en effaçant le caractère à la nouvelle position. Si vous maintenez cette touche enfoncée pendant plus d'une demi-seconde, le HP-71 répète son action.
- f -CHAR efface le caractère à la position du curseur et déplace les caractères suivants d'une position vers la gauche. Si vous maintenez la touche enfoncée pendant plus d'une demi-seconde, le HP-71 répète son action.
- f I/R passe du curseur de remplacement (■) au curseur d'insertion (♣) et inversement. Lorsque le curseur de remplacement est à l'affichage, tout caractère entré remplace l'espace ou le caractère à la position du curseur. Lorsque le curseur d'insertion est à l'affichage, tout caractère entré est inséré entre le caractère sur lequel se trouve le curseur et le précédent.
- f -LINE efface tous les caractères de la position du curseur à la fin de la ligne (y compris éventuellement ceux se trouvant entre la fin de la fenêtre d'affichage et la fin de la ligne).

Nom de fichier de programme (EDIT)

Nota: Pour vous aider à trouver plus facilement les touches d'aide à la frappe, les séquences de touches des pages suivantes seront représentées à la fois par leurs mnémoniques et par les symboles des touches permettant de les afficher. Par exemple, EDIT sera représenté par:

(f Z)
EDIT

Le HP-71 peut contenir de nombreux programmes en mémoire. Chaque programme est stocké dans un fichier que vous devez identifier par un nom. Les noms se composent de un à huit caractères: le premier devant être une lettre et les suivants des lettres ou des chiffres.

Nous allons bientôt entrer un programme en mémoire mais nous devons d'abord utiliser l'ordre EDIT pour créer et nommer le fichier dans lequel ce programme sera stocké.

Entrée/Résultat

```
(f Z)
EDIT DEPASSE END LINE
```

```
DEPASSE BASIC 0 →
```

Le HP-71 affiche l'étiquette du fichier DEPASSE qui est maintenant prêt à recevoir le programme.

La flèche droite dans l'affichage indique que l'étiquette continue vers la droite. L'étiquette indique que DEPASSE est un fichier BASIC vide. Les catalogues et les étiquettes sont décrits au chapitre 6.

Entrée, modification et exécution du programme DEPASSE

Lors de l'entrée d'un programme en mémoire les erreurs que vous faites peuvent être de deux types:

1. Les erreurs dont vous vous apercevez avant d'appuyer sur `END LINE` ou celles que le HP-71 repère dès que vous appuyez sur `END LINE`.
2. Les erreurs que ni vous ni le HP-71 ne repérez avant l'exécution du programme.

Il est possible que vous fassiez des erreurs de chaque type lors de l'entrée de ce programme ou de tout autre. Pour vous aider à apprendre à traiter ces erreurs, nous introduirons volontairement une erreur de chaque type et nous vous montrerons comment les corriger (les erreurs et le traitement d'erreur sont présentés plus en détails au chapitre 9).

Après le listage suivant du programme DEPASSE, nous vous indiquerons comment entrer chaque ligne (y compris les deux erreurs). Remarquez que les lignes de programmes commençant par ! sont des lignes de *commentaire*, ignorées par l'ordinateur lors de l'exécution du programme. Le HP-71 affiche ou imprime néanmoins ces lignes dans les listages. Le symbole `␣` permet de concaténer (joindre) plusieurs instructions sur une même ligne.

```
10 ! PROGRAMME DEPASSE
20 REAL X,Y @ STD
30 FOR X=1 TO 20
40 Y=(X^2)^(X^2) @
   DISP Y @ IF Y=MAXREAL THEN 60
50 NEXT X
60 "Le plus grand nombre positif fini que peut
   afficher le HP-17 est";Y
```

Dans la séquence suivante, frappez les erreurs telles qu'elles sont indiquées, nous vous montrerons comment les corriger.

Entrée/Résultat

10 ! PROFEAMME PASSE

```
>10 ! PROFEAMME PASSE■
```

Nous corrigerons cette ligne de façon que le texte soit PROGRAMME DEPASSE.

Appuyez 12 fois sur .

```
>10 ! PROFEAMME PASSE
```

GR

```
>10 ! PROGRAMME PASSE
```

      I/R

```
>10 ! PROGRAMME ■ASSE
```

Le curseur d'insertion indique la position où le prochain caractère doit être inséré.

DE

```
>10 ! PROGRAMME DE■ASS
```



Entre la ligne 10 dans le fichier. La touche  agit de la même façon quelle que soit la position du curseur et son type.

```
>■
```

Le HP-71 est prêt pour la ligne suivante.

20 REAL X,Y @ STD 

REAL X,Y (page 57) déclare les variables X et Y en pleine précision et STD (page 55) définit un format d'affichage pour des nombres en pleine précision.

```
>■
```

(f[R]) (f[T])
 30 FOR X=1 TO 20 [END LINE]

> █

40 Y=(X^2)^(X^2) @ Y

La variable Y à la fin de la ligne de programme (après le symbole @) constitue une instruction DISP implicite. Ceci est équivalent à la frappe de DISP Y. L'instruction DISP implicite est présentée plus en détails à la page 67 (le HP-71 permet aussi l'utilisation d'instruction LET implicite).

>40 Y=(X^2)^(X^2) @ Y █

(f[Q]) (f[W])
 @ IF Y=MAXREAL THEN 60

Remarquez les espaces apportés par les aides à la frappe. MAXREAL est le nom dans le HP-71 du plus grand nombre positif fini que l'ordinateur peut représenter.

← IF Y=MAXREAL THEN 60 █

La flèche gauche indique que le reste de la ligne est à gauche en dehors de la fenêtre d'affichage.

[END LINE]

> █

(f[Y])
 50 NEXT X [END LINE]

> █

60 "L (f[LC]) e plus grand no

(f[LC]) passe le clavier de mode majuscule en mode minuscule et réciproquement.

>60 "Le plus grand no █

mbre positif ni que

Continue la ligne 60.

```
← mbre positif ni que■
```

Lors de la frappe de la partie suivante de cette ligne, utilisez la touche **[9]** pour frapper HP. Nous vous expliqueront comment corriger le 17 après avoir exécuté le programme.

Entrée/Résultat

peut acheter le HP-

```
← peut acheter le HP-■
```

17 est"; **[f][LC]** Y

[f][LC] remet le HP-71 en mode majuscule.

```
← 17 est"; Y■
```

[END LINE]

Entrez la ligne 60 dans le fichier. Le programme est maintenant complet.

```
>■
```

Appuyez sur **[RUN]** pour exécuter ce programme.

Le témoin **PRGM** à l'extrémité gauche de l'affichage s'allume et le HP-71 affiche une série de nombres de plus en plus grands. Parmi ceux-ci, vous pouvez remarquer le nombre 1.E200. Le E signifie «exposant»; ce nombre représente donc 1×10^{200} . Juste après l'affichage de 1.91509351599E449, le HP-71 affiche un message, WRN L40:Overflow pour vous indiquer que le nombre suivant de la série est plus grand que le plus grand nombre affichable par le HP-71, 9.999999999999E499.

Lorsque la ligne 60 affiche le message programme, vous vous apercevez que le numéro de modèle de votre ordinateur est 71 et non pas 17. Vous pouvez maintenant corriger cette erreur en suivant la procédure ci-après.

Entrée/Résultat

(f) (B)
 FETCH 60 END LINE

```
>60■DISP "Le plus gran →
```

Rappelle la ligne 60 à l'affichage.

Lorsque vous rappelez une ligne, le HP-71 positionne le curseur immédiatement après le numéro de ligne de façon à faciliter les corrections. La flèche droite indique que la ligne continue au delà de l'affichage.

9 →

```
← icher le HP-17 est");Y■
```

Déplace le curseur à l'extrémité droite de la ligne.

Appuyez neuf fois sur ← pour placer le curseur sur le 1 de 17.

```
← icher le HP-17 est");Y
```

71 END LINE

Le programme est corrigé.

```
>■
```

Contrôle de la vitesse d'affichage (DELAY)

```
DELAY vitesse de ligne [, vitesse de caractère]
```

Vous pouvez contrôler le nombre de secondes pendant lequel une ligne doit être affichée avant d'être remplacée par la suivante (*vitesse de ligne*) et aussi la vitesse de déroulement des lignes contenant plus de 22 caractères (paramètre *vitesse de caractère* optionnel). Vous pouvez choisir une vitesse de ligne et une vitesse de caractère entre 0 et 8 secondes. Les vitesses supérieures à huit secondes sont considérées infinies; il n'y a ni remplacement de ligne ni déroulement de caractères. Ces deux paramètres sont indépendants l'un de l'autre.

Stockage du programme `DEPASSE (EDIT, NAME)`

Supposons que vous vouliez maintenant entrer un autre programme et que vous oubliez de créer un nouveau fichier avec `EDIT` avant d'entrer les lignes de programme. Où donc le HP-71 placera-t-il ces nouvelles lignes? Dans le même fichier que les lignes du programme `DEPASSE`. En fait elles remplaceraient les lignes du programme `DEPASSE` ayant les mêmes numéros.

Vous pouvez éviter ce type de problème en créant toujours un nouveau fichier avant de frapper des lignes de programme. Ou bien vous pouvez utiliser un brouillon appelé `workfile` (*fichier de travail*) dans lequel vous pouvez entrer toutes vos lignes. Pour ouvrir ce fichier, il suffit d'exécuter `EDIT` sans nom de fichier.

Entrée/Résultat

`EDIT` `ENDLINE`

```
workfile BASIC 0 →
```

Cet affichage contient la première partie de l'étiquette du fichier de travail et indique qu'il s'agit d'un fichier `BASIC` vide. Les catalogues et les étiquettes sont présentés plus en détails au chapitre 6.

Si un *fichier de travail* existe déjà, l'instruction `EDIT` `ENDLINE` vous y positionne. Le HP-71 ne peut contenir qu'un seul fichier de travail.

Vous pouvez créer un nouveau fichier de travail vide si vous changez d'abord le nom du fichier de travail courant en utilisant l'ordre `NAME`. Cet ordre est présenté plus en détails au chapitre 6, pages 100 à 102.

En résumé, lors de l'entrée d'un programme dans la mémoire du HP-71, il est recommandé de:

1. Créer un nouveau fichier en exécutant `EDIT nom de fichier`.
2. Entrer les lignes de programmes, tester le programme et le modifier si nécessaire.
3. Ouvrir le fichier de travail en exécutant `EDIT`.

Exécution d'un programme quelconque en mémoire

Vous disposez de deux méthodes pour exécuter un programme:

- Exécutez `RUN nom de fichier`. Ceci est valable pour tous les programmes en mémoire.
- Appuyez sur `RUN` (ou exécutez `RUN`). Ceci n'est valable que pour le programme stocké dans le fichier courant.

Exécution du programme courant. Pour l'instant, le fichier de travail est le fichier courant. Le HP-71 contient toujours un seul fichier courant. Lorsque vous entrez un programme au clavier, le HP-71 le place automatiquement dans le fichier courant et vous pouvez l'exécuter en appuyant simplement sur `[RUN]`. Vous pouvez aussi le modifier immédiatement tel que nous l'avons fait avec le programme `DEPASSE`.

Exécution d'autres programmes. Vous pouvez exécuter tout programme en mémoire en exécutant l'instruction `RUN nom de fichier`. Ceci est même la seule façon d'exécuter un programme ne se trouvant pas dans le fichier courant. Lorsque vous exécutez ainsi un programme, il devient le programme courant et vous pouvez le modifier et utiliser `[RUN]` pour recommencer son exécution.

Personnalisation du clavier (`[f] USER`), (`[g] 1 USER`)

Le HP-71 comporte en fait deux claviers complets et indépendants qui se partagent les mêmes touches. Le clavier Normal, celui que vous avez utilisé jusqu'à maintenant, correspond aux mnémoniques imprimées sur et au-dessus des touches. Le clavier Utilisateur possède les mêmes fonctions sauf pour les touches ayant été redéfinies par l'utilisateur. Il y a deux façons de passer d'un clavier à l'autre:

- `[f] USER` passe définitivement d'un clavier à l'autre. Il suffit d'appuyer à nouveau sur `[f] USER` pour revenir au clavier précédent.
- `[g] 1 USER` passe d'un clavier à l'autre pour une seule touche avec ou sans préfixe. Le HP-71 revient automatiquement au clavier précédent.

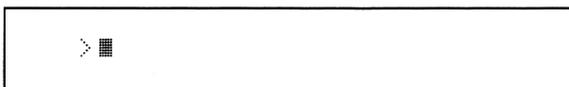
Vous pouvez redéfinir les fonctions des positions primaire et secondaires de toutes les touches sauf deux: `[f]` et `[g]`. Vous pouvez redéfinir les touches comme aides à la frappe, pour exécuter des instructions quelconques ou des combinaisons d'instructions reliées par `⌘`. Vous pouvez aussi utiliser les touches redéfinies comme aides lors de l'entrée de données en réponse à un message de programme. Chaque définition de touches est limitée aux 96 caractères d'une ligne d'affichage. Nous allons maintenant effectuer une redéfinition simple.

Entrez la définition de touche suivante:

Entrée/Résultat

```
([f][U])([f][I])
DEF KEY "<", "TIME#" [END LINE]
```

Redéfinit la séquence `[g]<` de façon à afficher l'heure lorsque vous appuyez sur ces touches sur le clavier Utilisateur.

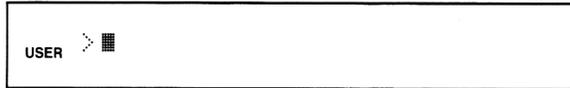


Vous pouvez maintenant utiliser la séquence **[g][<]** de deux façons:

Entrée/Résultat

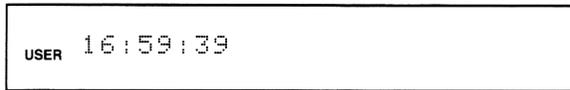
[f][USER]

Active le clavier Utilisateur.



Le témoin **USER** vous indique que le clavier Utilisateur est actif.

[g][<]



Il est presque 17 heures.

[f][USER]



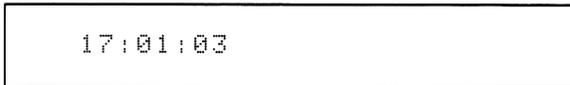
Le témoin **USER** est éteint et le clavier Normal est à nouveau actif.

Voici une autre façon d'utiliser la même touche redéfinie:

Entrée/Résultat

[g][1][USER][g][<]

Maintenez la touche **[g]** enfoncée et appuyez successivement sur **[0]** et sur **[<]**.

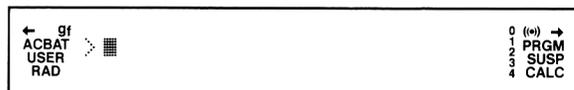


Remarquez que le témoin **USER** n'est pas allumé. La séquence **[g][1][USER]** n'active le clavier Utilisateur que pour la séquence suivante et retourne immédiatement au clavier Normal.

Contraste et inclinaison de l'affichage (CONTRAST)



L'ordre **CONTRAST** vous permet de choisir différentes valeurs de contraste entre 0 et 15 pour contrôler l'intensité de l'affichage et l'angle d'inclinaison. **CONTRAST 0** donne le contraste minimum et l'inclinaison maximum alors que **CONTRAST 15** donne le contraste maximum et une inclinaison minimum. **CONTRAST 15** est en outre le mode dans lequel les témoins sont les plus visibles. Après une ré-initialisation, la valeur de contraste vaut par défaut 9. Vous pouvez essayer diverses valeurs et choisir celle qui vous convient le mieux.

Entrée/RésultatCONTRAST 15 END LINE

Si vous regardez l'affichage de dessus, vous pouvez voir le symbole BASIC et le curseur affiché avec un très fort contraste. Si vous inclinez le HP-71, vous pouvez voir les témoins et tous les points servant à former les caractères.

Témoins d'état

Le tableau suivant résume les témoins d'état du HP-71.

Témoin	Signification
←	La ligne courante s'étend au-delà de la gauche de l'affichage
g	Vous avez appuyé sur g mais pas encore sur la deuxième touche de la séquence.
f	Vous avez appuyé sur f mais pas encore sur la deuxième touche de la séquence.
AC	Réservé pour utilisation ultérieure.
BAT	Batterie faible.
USER	Le clavier Utilisateur est actif.
RAD	L'unité d'angle est le radian.
0	L'indicateur* 0 est armé.
1	L'indicateur 1 est armé.
2	L'indicateur 2 est armé.
3	L'indicateur 3 est armé.
4	L'indicateur 4 est armé.
(•)	Réservé pour utilisation ultérieure
→	La ligne s'étend au delà de la droite de l'affichage.
PRGM	Un programme est en cours d'exécution.
SUSP	L'exécution d'un programme est suspendue.
CALC	Le HP-71 est en mode CALCUL.

* Les indicateurs sont présentés au chapitre 11.

Vous pouvez revenir au contraste par défaut en exécutant `CONTRAST 9`.

Rappel des cinq dernières instructions—Pile d'instructions (**9** **CMS**)

Le HP-71 conserve en permanence une liste des cinq dernières instructions exécutées dans une partie de la mémoire appelée *pile d'instructions*. Ces instructions comprennent tout ce qui a été exécuté (suivi de **END LINE**) au clavier tel que les évaluations d'expressions, entrées de lignes de programme, les ordres et les fonctions. Vous pouvez afficher toute instruction de la pile et la ré-exécuter (en appuyant à nouveau sur **END LINE**) ou la modifier. Cette pile d'instructions s'avère particulièrement utile lors de l'exécution répétitive d'une série d'instructions identiques ou très similaires. Par exemple, l'évaluation d'une série d'expressions de 10 termes lorsqu'un seul des termes diffère est beaucoup plus simple avec la pile d'instructions: il suffit de modifier un terme à chaque fois au lieu de ré-entrer l'expression complète.

Si plusieurs des cinq dernières instructions sont identiques, la pile n'en conserve qu'une.

Procédure d'utilisation de la pile d'instructions:

- Activez la pile d'instructions en appuyant sur **9** **CMS**.
- Désactivez la pile d'instructions en appuyant à nouveau sur **9** **CMS** ou en appuyant sur **ATTN**.
- Désactivez la pile d'instructions et exécutez l'instruction affichée en appuyant sur **END LINE**.

Affichez maintenant quelques unes des instructions que vous venez d'entrer.

Entrée/Résultat

9 **CMS**

```
\CONTRAST 9
```

Active la pile d'instructions.

L'affichage contient l'instruction la plus récente avec le curseur de remplacement. Le symbole `\` identifie la pile d'instructions. Toutes les touches de déplacement du curseur et de mise au point sont actives.

▲

```
\CONTRAST 9
```

Utilisez les touches **▲** et **▼** pour vous déplacer dans la pile. Vous pouvez afficher les instructions antérieures en appuyant sur **▲** et les plus récentes en appuyant sur **▼**.

```
\CONTRAST 15
```

L'instruction précédente.

END LINE

```

← gf
ACBAT  > █
USER
RAD
0 (0) →
1 PRGM
2 SUSP
3
4 CALC

```

9) CMDS

```

← gf
ACBAT  \ CONTRAST 15
USER
RAD
0 (0) →
1 PRGM
2 SUSP
3
4 CALC

```

▲

```

← gf
ACBAT  \ CONTRAST 9
USER
RAD
0 (0) →
1 PRGM
2 SUSP
3
4 CALC

```

END LINE

```

> █

```

Vous pouvez exécuter toute expression ou instruction de la pile en appuyant sur `END LINE`.

La pression de `END LINE` exécute l'instruction affichée (`CONTRAST 15`) et désactive la pile d'instructions. `CONTRAST 15` devient alors l'instruction la plus récente.

Réactive la pile d'instructions et affiche l'instruction la plus récente.

Affiche l'instruction précédente dans la pile.

Exécute l'instruction affichée et désactive la pile.

Le symbole BASIC indique que la pile n'est plus active.

Signaux sonores (BEEP)

L'ordre `BEEP` produit un son dont vous pouvez contrôler la fréquence et la durée. Vous pouvez en outre éteindre ce signal et choisir entre deux niveaux d'intensité. L'application principale de ce signal sonore est l'émission d'indicateur sonore d'erreur ou d'état pour les programmes.

Cet ordre peut prendre cinq formes:

```

BEEP
BEEP fréquence en hertz
BEEP fréquence en hertz , durée en secondes
BEEP ON
BEEP OFF

```

Quelques faits concernant BEEP:

- Lorsque vous exécutez BEEP sans spécifier de fréquence ou de durée, le HP-71 émet un son à 500 hertz pendant 0,25 seconde.
- Vous pouvez spécifier des fréquences entre 0 et 4900 hertz, néanmoins seules les fréquences supérieures à 150 hertz produisent des sons reconnaissables.
- Vous pouvez spécifier des durées entre 0 et 1000 secondes.
- Vous pouvez spécifier les fréquences et les durées sous la forme d'expressions numériques. Le HP-71 évalue ces expressions lors de l'exécution de BEEP.
- BEEP ON autorise l'émission de bips.
- BEEP OFF désactive l'émission de bips.
- Après une ré-initialisation de la mémoire, BEEP ON est actif.
- Vous pouvez augmenter l'intensité des sons en exécutant SFLAG -25 (arme l'indicateur -25). Les indicateurs sont présentés plus en détails au chapitre 11.
- Vous pouvez redéfinir l'intensité du niveau par défaut en exécutant CFLAG -25 (efface l'indicateur -25).

Et ensuite!

Ce chapitre vous a présenté succinctement quelques unes des caractéristiques du HP-71. Cet ordinateur a beaucoup d'autres possibilités mais il n'est peut-être pas nécessaire que vous les étudiez toutes maintenant. Si vous désirez simplement exécuter des programmes existants, il vous suffit de lire les chapitres 6 et 8 pour vous familiariser avec la structure de fichier utilisée par le HP-71 et apprendre à lancer l'exécution. Lisez le chapitre 2 si vous voulez utiliser le HP-71 pour faire des calculs au clavier et le chapitre 4 si ces calculs concernent la solution de problèmes de statistiques.

Pour la programmation, lisez les chapitres 2, 3, 6 et 8 à 14. Le chapitre 5 décrit le fonctionnement de l'horloge et le chapitre 7 celui du clavier Utilisateur.

Nous vous conseillons en-outré de parcourir le *Manuel de référence* et l'*Aide mémoire* pour vous familiariser avec leurs contenus. Le *Manuel de référence* contient par exemple un glossaire des termes couramment utilisés dans ces manuels.

Conventions de syntaxe

La syntaxe est la façon dont les instructions doivent être frappées pour être comprises par l'ordinateur. Nous utiliserons les conventions suivantes tout au long de ce manuel.

CHARACTERES MATRICES	Tous les éléments en caractères matricés doivent être introduits tels qu'ils apparaissent (les mots-clés peuvent être introduits indifféremment en majuscules ou en minuscules).
<i>italiques</i>	Les éléments en italiques sont les paramètres des ordres ou les arguments des fonctions.
" " , ' '	Les chaînes de caractères peuvent être spécifiées entre guillemets ou apostrophes (les exemples de ce manuel utilisent les guillemets). Les noms de fichiers peuvent être en général spécifiés avec ou sans guillemets ou apostrophes (les exemples du manuel n'utilisent ni guillemets ni apostrophes). Le HP-71 convertit automatiquement tous les noms de fichier en majuscules.
[]	Les éléments entre crochets sont optionnels.
<i>éléments empilés</i>	Lorsque plusieurs éléments sont empilés, vous devez choisir l'un d'entre-eux uniquement.
...	Trois points suivant un élément indiquent que ce dernier peut être répété.

Les descriptions des mots-clés (ordres, fonctions et opérateurs) qui apparaissent dans ce manuel vous permettent de les utiliser correctement. Néanmoins, certaines d'entre-elles n'incluent pas tous les détails. Elles sont indiquées par la mention «Syntaxe simplifiée». Pour obtenir une description complète et détaillée des mots-clés, consultez le «*Dictionnaire des mots-clés*» dans le *Manuel de référence*.

Calculs avec le HP-71

Table des matières

Présentation	37
Utilisation du mode CALCUL	37
Caractéristiques du mode CALCUL	38
Corrections des erreurs de frappe	45
Opérations non-autorisées	46
Messages d'avertissement en mode CALCUL	46
Opérateurs arithmétiques (+, -, *, /, ^, DIV, %)	47
Fonctions numériques	47
Fonctions d'altération des nombres (ABS, IP, FP, INT, FLOOR, CEIL) ..	48
Conversions décimale et hexadécimale (DTH\$, HTD)	48
Fonctions générales (PI, SQR, FACT, MAX, MIN, MOD, RMD, RED, RES, SGN)	49
Fonctions logarithmiques (LGT, LOG, EXP, EXPONENT, LOGP1, ESPM1)	50
Unités d'angles (RADIANS, DEGREES)	50
Fonctions trigonométriques (SIN, COS, TAN, ASIN, ACOS, ATAN, DEG, RAD, ANGLE)	51
Nombres aléatoires (RND, RANDOMIZE)	52
Format des nombres	54
Notation exponentielle (E)	54
Format d'affichage standard (STD)	55
Notation à décimale fixe (FIX)	55
Notation scientifique (SCI)	55
Notation ingénieur (ENG)	56
Précision numérique (OPTION ROUND)	56
Précision des variables numériques (REAL, SHORT, INTEGER)	57
Exceptions mathématiques (IVL, DVZ, OVZ, UNF, INX)	57
Traitement des exceptions (DEFAULT ON, DEFAULT OFF, DEFAULT EXTEND)	58
Proposition IEEE pour le traitement des exceptions mathématiques (+0, -0, +Inf, -Inf, INF, NAN, NaN, TRAP, ?)	59
Catégories de nombres (CLASS)	60

Intervalle des nombres (MINREAL, EPS, MAXREAL)	61
Opérateurs de relation (Combinaisons de <, =, >, #, ?)	62
Opérateurs logiques (AND, OR, EXOR, NOT)	62
Hierarchie des opérateurs	64

Présentation

Ce chapitre vous présente:

- Une nouvelle façon de calculer avec un ordinateur: mode CALCUL.
- Tous les opérateurs mathématiques: arithmétiques, relationnels et logiques.
- Toutes les fonctions mathématiques.
- Les nombres aléatoires et leur utilisation.
- Trois façons de formater les nombres à l'affichage.
- La précision des nombres affichés et stockés.
- Les exceptions mathématiques: opérations invalides, division par zéro, dépassements supérieur et inférieur, résultat inexact.
- Les trois réponses à chaque exception.
- La proposition IEEE pour le traitement des exceptions mathématiques.

Utilisation du mode CALCUL

Le HP-71 vous permet d'évaluer toute expression numérique dans deux modes qui utilisent la hiérarchie algébrique des opérateurs (page 64).

- Le mode BASIC, utilisé par la plupart des ordinateurs programmables dans ce langage. Vous devez introduire l'expression entière avant que l'évaluation ne soit lancée par la pression de la touche **END LINE**.
- Le mode CALCUL facilite l'évaluation d'expressions au clavier en affichant les résultats intermédiaires des calculs pendant l'introduction des expressions.

ATTENTION

Ne connectez ou retirez pas de module lorsque le mode CALCUL est actif. Ceci provoquerait une ré-initialisation de la mémoire (perte des informations en mémoire). Consultez le chapitre 6 pour plus d'informations sur l'utilisation des modules enfichables.

Caractéristiques du mode CALCUL

La liste de caractéristiques suivante (certaines sont accompagnées d'exemple) vous montre comment utiliser au mieux le mode CALCUL pour évaluer des expressions numériques.

Jeu complet de fonctions numériques. Vous pouvez utiliser toutes les fonctions et tous les opérateurs numériques du HP-71 en mode CALCUL, y compris les fonctions numériques unilignes définies par l'utilisateur présentes dans le fichier courant.

Jeu de variables communes. Les modes CALCUL et BASIC utilisent le même jeu de variables. Une variable définie en mode BASIC conserve sa valeur en mode CALCUL et vice versa.

Résultat implicite. Lors de l'entrée d'une expression, vous pouvez utiliser le résultat de l'expression précédente en spécifiant simplement une paire de parenthèses vides.

Affectations de touches par l'utilisateur. Vous pouvez utiliser en mode CALCUL les affectations de touches du clavier Utilisateur sauf les définitions de touches à exécution seule (les définitions de touches sont présentées au chapitre 7).

Complexité sans bornes des expressions. Vous pouvez évaluer en mode CALCUL toute expression numérique qui peut être entrée et exécutée en BASIC.

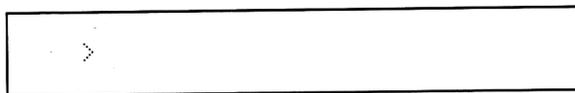
Calcul sur 12 chiffres. Le HP-71 calcule tous les résultats avec 12 chiffres après la virgule.

Instructions d'affectation. Vous pouvez affecter des valeurs à des variables puis utiliser ces dernières dans des expressions comme le montre l'exemple suivant.

Nota: Pour cet exemple et les suivants, nous vous recommandons de placer l'affichage de votre HP-71 en format `FIX 2` de façon qu'il corresponde aux illustrations de ce manuel. Vous ne pouvez pas exécuter `FIX 2` en mode CALCUL, vérifiez donc que le HP-71 est en mode BASIC (si le témoin **CALC** est allumé à l'extrémité droite de l'affichage, appuyez sur `f` `CALC` pour passer en mode BASIC). Vous devez voir le symbole `>` à l'extrémité gauche de l'affichage. Si l'affichage n'est pas vierge, appuyez sur `ATTN`.

Entrée/Résultat

`FIX 2` `END LINE`



`f` `CALC`



Tous les nombres entrés et tous les résultats seront affichés arrondis à deux chiffres après la virgule.

Passe en mode CALCUL.

$A=3/2$

A=3.00/2# CALC

-5

A=1.50-5# CALC

END LINE

-3.50 CALC

A

A# CALC

*2

-3.50*2# CALC

END LINE

-7.00 CALC

Commence l'exemple d'instruction d'affectation.

Il n'y a pas encore eu d'évaluation.

La pression de \square provoque l'évaluation du début de l'expression.

Termine l'expression et affecte sa valeur à A.

Utilisez A dans une expression pour confirmer que cette variable représente bien 3.50.

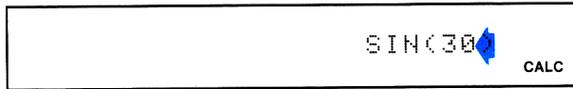
Dès l'entrée d'un opérateur (*), le HP-71 remplace A par sa valeur.

Affiche le résultat.

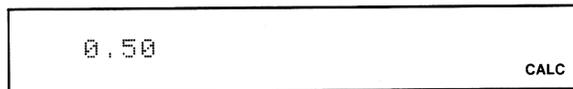
Correspondance automatique des parenthèses. Pour chaque parenthèse gauche que vous entrez, le HP-71 place automatiquement une parenthèse droite. Vous n'avez donc pas besoin de frapper la dernière parenthèse à la fin d'une ligne. Vous pouvez néanmoins la frapper, le HP-71 interprète correctement l'expression.

Entrée/Résultat

SIN(30


 A rectangular display box containing the text "SIN(30" followed by a blue cursor arrow pointing to the right. To the right of the display is a small icon of a calculator with a blue arrow pointing to the closing parenthesis key. Below the display is the word "CALC".

END LINE


 A rectangular display box containing the text "0.50". To the right of the display is a small icon of a calculator with a blue arrow pointing to the closing parenthesis key. Below the display is the word "CALC".

L'aide à la frappe au dessus de la touche $\boxed{4}$ vous permet d'entrer SIN avec la parenthèse gauche en une seule pression de touche.

La parenthèse droite clignotante à la position du curseur représente un nombre de parenthèses droites égale au nombre de parenthèses gauches ouvertes. Dans ce cas, une seule.

Il n'a pas été nécessaire de frapper $\boxed{[$ ou $\boxed{]}$ pour entrer cette expression.

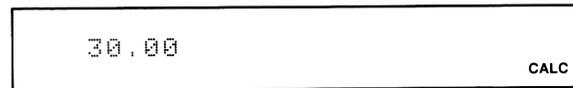
Résultat implicite (). Une paire de parenthèses vides entrée dans une expression représente la valeur de la dernière expression évaluée. L'expression courante utilise cette valeur pour le calcul. Une paire de parenthèses vides peut représenter soit un membre d'une expression soit l'argument d'une fonction.

Entrée/Résultat

ASIN(


 A rectangular display box containing the text "ASIN(" followed by a blue cursor arrow pointing to the right. To the right of the display is a small icon of a calculator with a blue arrow pointing to the closing parenthesis key. Below the display is the word "CALC".

END LINE


 A rectangular display box containing the text "30.00". To the right of the display is a small icon of a calculator with a blue arrow pointing to the closing parenthesis key. Below the display is the word "CALC".

A= END LINE


 A rectangular display box containing the text "30.00". To the right of the display is a small icon of a calculator with a blue arrow pointing to the closing parenthesis key. Below the display is the word "CALC".

La pression de $\boxed{\text{END LINE}}$ entre la parenthèse droite, rappelle le résultat précédent représenté par la paire de parenthèses et évalue $\text{ASIN}(0.50)$.

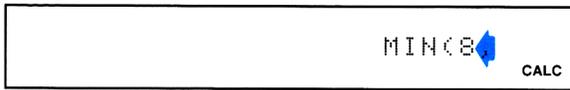
Ceci illustre une autre application importante des résultats implicites.

Le résultat précédent, 30.00, est maintenant affecté à la variable A.

Liste de virgules pour les arguments de fonction. Pour les fonctions et tableaux nécessitant plusieurs arguments, l'affichage indique le nombre minimum de virgules.

Entrée/Résultat

MIN(8

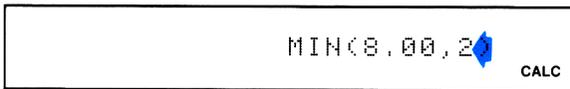


Entrez le premier argument d'une fonction MIN.

La virgule partageant la position du curseur indique que vous devez entrer au moins un autre argument.

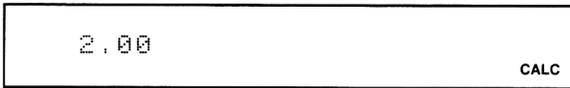
,2

Entrez la virgule et le second argument. Le caractère suivant le 8 pouvant être une virgule ou la suite de la première expression, vous devez frapper la virgule au clavier.



La parenthèse clignotante vous indique qu'aucun autre argument n'est nécessaire.

END LINE



Cette fois encore, vous n'avez pas eu à entrer la dernière parenthèse avant de terminer l'expression.

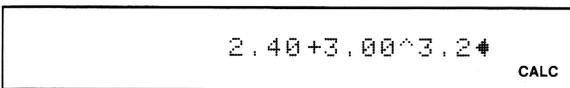
Evaluation pas-à-pas (f[SST] ou [RUN]). Lorsque vous entrez une expression en mode CALCUL, il peut arriver que certains termes soient évalués simultanément. Néanmoins, il peut être intéressant pour vous de connaître chaque résultat intermédiaire sans perturber l'ordre dans lequel les opérateurs doivent agir (hiérarchie).

Exemple: Supposons que vous vouliez calculer $\log(11) + 3^{3.2} - 4$ et que vous vouliez en-outre connaître le résultat de $3^{3.2}$. Remarquez que vous disposez sur le clavier de l'aide à la frappe (f[-] pour LOG.

Entrée/Résultat

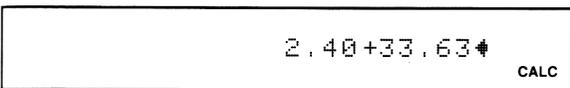
LOG(11)+3^3.2

Puisque vous voulez connaître la valeur de $3^{3.2}$, n'entrez pas -4 sinon le HP-71 évaluerait immédiatement $\log(11) + 3^{3.2}$.

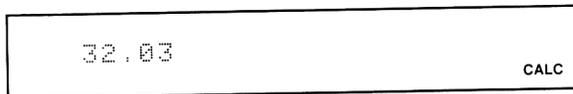


[RUN]

Les touches [RUN] et (f[SST] ont la même fonction lorsque le mode CALCUL est actif.

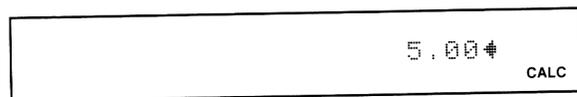


L'exécution pas-à-pas affiche la valeur de $3^{3.2}$.

-4 **END LINE**

 A rectangular display box containing the number 32.03 on the left and the word CALC on the right.

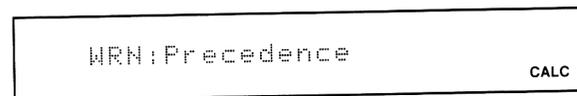
Résultat final.

Exemple: Le mode CALCUL ne vous permet pas d'enfreindre la hiérarchie des opérateurs lorsque vous évaluez une expression pas-à-pas. Vérifiez ceci en utilisant l'expression $2 + 3 \times 4$.

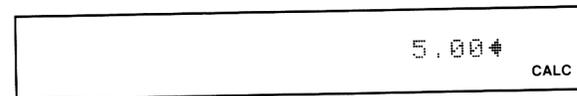
Entrée/Résultat2+3 **RUN**

 A rectangular display box containing the number 5.00 followed by a small symbol (a vertical line with a horizontal bar) on the left and the word CALC on the right.

Affiche le résultat intermédiaire.

Vous essayez d'entrer * mais le HP-71 n'effectue pas de multiplication. Il affiche:


 A rectangular display box containing the text WRN:Precedence on the left and the word CALC on the right.

Le HP-71 vous indique que la multiplication doit être effectuée avant l'addition.


 A rectangular display box containing the number 5.00 followed by a small symbol (a vertical line with a horizontal bar) on the left and the word CALC on the right.

L'affichage précédent remplace le message après une courte pause.

Nous vous indiquerons ci-après comment corriger cette erreur en utilisant la pile d'instructions.

Rappel de l'expression complète (**▲**). La pression de **▲** active la pile d'instructions uniquement lorsque le mode CALCUL est actif. L'affichage résultant peut être de deux types:

- L'affichage contient tous les termes de la dernière expression évaluée plus le symbole \downarrow indiquant que l'expression vient d'être évaluée.
- L'affichage contient tous les termes de l'expression entrée dont le résultat final n'est pas encore évalué. Cet affichage de la pile d'instructions rappelle les opérateurs et les opérandes que vous avez introduits dans le HP-71 à l'origine.

Dans les deux cas, vous pouvez modifier l'expression affichée en utilisant le curseur mobile de la pile d'instructions.

Lorsque la pile d'instructions est active, **END LINE** donne différents résultats selon la présence ou l'absence de \downarrow à l'affichage:

- \downarrow non affiché: **END LINE** désactive la pile d'instructions et affiche l'expression en partie évaluée, y compris les effets des éventuelles mises au point dans la pile d'instructions.
- \downarrow affiché: **END LINE** évalue l'expression affichée (y compris les résultats de toute mise au point), désactive la pile d'instructions et affiche le résultat final.

Entrée/Résultat

2+3 CALC

▶▶▶ *4 END LINE

2.00+3.00*4⚡ CALC

END LINE

14.00 CALC

Active la pile d'instructions et rappelle la forme initiale de votre expression de façon que vous puissiez corriger l'erreur de hiérarchie.

Le clavier Utilisateur est prêt pour les mises au point.

Complète l'expression et désactive la pile d'instructions.

Dans ce cas, END LINE affiche la même expression non évaluée.

Affiche le résultat.

Supposons que vous vouliez évaluer l'expression $6-3^7$ mais que vous entriez $6+3^7$ et que vous ne remarquiez votre erreur (+ au lieu de -) qu'après l'évaluation. Il n'est pas nécessaire de ré-entrer toute l'expression. Vous pouvez activer la pile d'instructions, modifier l'expression et la ré-évaluer.

Entrée/Résultat

6+3^7 END LINE

2193.00 CALC



6+3^7⚡ CALC

Affiche un résultat incorrect.

Vous vous apercevez que vous avez entré un + au lieu du -.

Active la pile d'instructions

Le symbole d'exécution ⚡ indique que vous avez appuyé sur END LINE juste avant d'activer la pile d'instructions.



6-3^7↓ CALC

END LINE

-2181.00 CALC

Évalue l'expression correcte.

Lorsque l'expression dans la pile d'instructions contient ↓, la pression de **END LINE** ré-évalue l'expression.

Exécution à rebours (**f** **BACK**). Avant qu'une expression soit complétée par **END LINE**, vous pouvez rappeler dans leur forme initiale les termes que le HP-71 a déjà utilisé dans des résultats intermédiaires en utilisant **f** **BACK**.

Exemple: Pour illustrer l'utilisation de l'exécution à rebours, supposons que vous introduisiez l'expression $4/8*3$ et qu'avant l'évaluation vous vous aperceviez que le 8 est erroné, ce devrait être un 9. Vous pouvez alors utiliser l'exécution à rebours pour remettre dans leur forme initiale les termes de l'expression jusqu'à la valeur erronée. Il suffit alors de frapper le nombre correct, de terminer l'expression et de l'évaluer.

Entrée/Résultat

4/8*3

Entrez l'expression incorrecte.

0.50*3↓ CALC

Le HP-71 affiche le résultat partiel.

f **BACK** **f** **BACK** **f** **BACK**

Vous devez exécuter **f** **BACK** pour chaque opérande ou opérateur y compris l'erreur (⊘).

4.00/↓ CALC

Vous êtes maintenant prêt à compléter l'expression correcte.

9*3 **END LINE**

Complétez et évaluez l'expression correcte.

1.33 ⊘ CALC

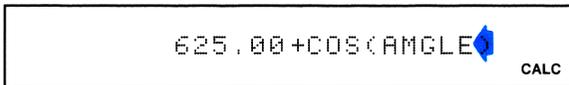
Correction des erreurs de frappe

Si vous essayez de compléter une fonction après en avoir incorrectement épilé le nom, le HP-71 affiche un message vous avertissant de l'erreur. Après un tel message, vous *devez* effacer tous les caractères du nom de la fonction en utilisant l'exécution à rebours, même si l'affichage suggère que cela ne soit pas nécessaire. Frappez ensuite le nom correct et complétez l'expression. L'exemple suivant montre pourquoi vous devez effacer tous les caractères.

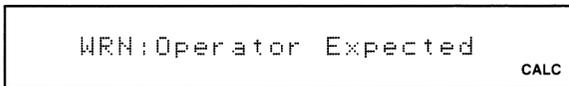
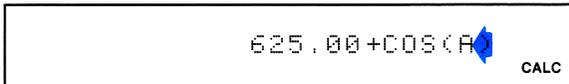
Exemple: Supposons que vous vouliez évaluer une expression dans laquelle se trouve `ANGLE(-5, 4)` (arc tangente de 4/-5 dans le quadrant correct) et que vous fassiez une erreur lors de la frappe de `ANGLE`.

Entrée/Résultat

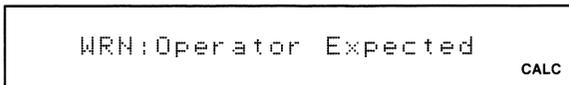
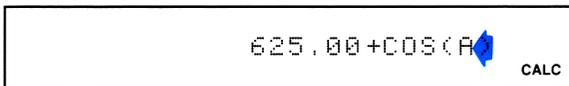
`5^4+COS(AMGLE`



`(`

`NGLE(`

Vous avez appuyé sur la touche **M** au lieu de **N**.

Vous ne vous apercevez pas encore de votre erreur et continuez la frappe.

Le HP-71, lui s'est aperçu de l'erreur et affiche un message.

Nous discuterons bientôt de la signification de ce message qui se trouve rapidement remplacé par:

Vous êtes déterminé à le frapper correctement cette fois-ci, vous entrez donc:

et:

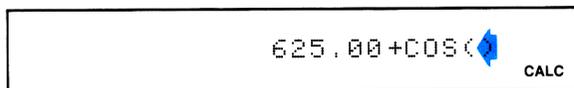
rapidement remplacé à nouveau par:

Que se passe-t-il?

Lorsque vous avez frappé `AMGLE(`, le HP-71 a cherché une fonction de ce nom. N'en trouvant pas, il a cherché un nom de variable et a trouvé `A`. Après un nom de variable, le HP-71 attend un opérateur, ce que `NGLE` n'est certainement pas; le HP-71 a donc ignoré tous les caractères suivant le `A`. Lorsque vous avez ensuite frappé `NGLE(`, l'affichage semblait correct, mais le HP-71 cherchait toujours un opérateur et a donc rejeté `NGLE` comme il avait rejeté `MGLE`. Pour corriger une erreur de ce type, vous devez effacer tous les caractères du nom de la fonction avec **f** **BACK** puis ré-introduire le nom complet.

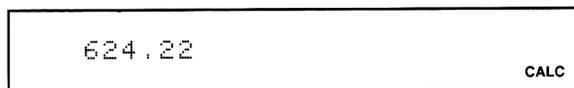
Entrée/Résultat

f BACK



ANGLE(-5,4) END LINE

Corrigez votre erreur de frappe et évaluez l'expression.


Opérations non autorisées

Le mode CALCUL est un outil puissant et simple pour les résolutions de problèmes au clavier. Néanmoins, certaines opérations ne sont pas autorisées dans ce mode:

- Chaînes alphanumériques.
- Fonctions de conversion décimale et hexadécimale \square TH\$ et HTD.
- Fonctions utilisateur multi-lignes.
- Instructions, sauf les affectations.
- Lignes de programme.

Messages d'avertissement en mode CALCUL

Dans les calculs de types suivants en mode CALCUL, veuillez à vérifier le contenu de la pile d'instructions avant de continuer.

- Evaluation d'une expression dans la pile d'instructions donnant un message d'avertissement (WRN).
- Evaluation à rebours (f BACK) donnant un message d'avertissement.
- Utilisation d'une touche définie du clavier Utilisateur donnant un message d'avertissement.

Remarquez que le HP-71 n'accepte pas les caractères venant après ceux ayant provoqués l'erreur.

Opérateurs arithmétiques (+, -, *, /, ^, DIV, %)

Le HP-71 comporte l'opérateur % en plus des opérateurs arithmétiques usuels du BASIC. Le tableau suivant illustre l'utilisation de ces opérateurs. Exécutez STD **END LINE** en mode BASIC pour placer l'affichage du HP-71 en format STD (voir page 55) de façon à obtenir des résultats semblables à ceux du tableau suivant.

Opérateurs arithmétiques

Opérateurs	Opération	Exemple avec résultat
+	Addition	5.25+4.75 10
-	Soustraction	5.25-4.75 .5
*	Multiplication	2*9 18
/	Division	9/2 4.5
^	Exponentiation	2^9 512
DIV	Division entière (sans reste)	3 DIV 2 1
%	L'opération x%y calcule x pourcent de y.	5%100 5

Fonctions numériques

Les *fonctions numériques* sont en fait des programmes incorporés à l'ordinateur qui traitent une information numérique ou alphanumérique, appelée *argument*, et donnent des valeurs simples. Les fonctions du HP-71 opèrent sur 0, 1 ou plusieurs arguments. Un argument peut être une variable, une autre fonction ou une expression entière à condition qu'elle puisse être réduite à une simple valeur au moment de l'évaluation.

Pour exécuter une fonction du HP-71 au clavier:

1. Frappez le nom de la fonction.
2. Frappez l'argument, si la fonction en demande un, entre parenthèses. Si la fonction demande plusieurs arguments, séparez-les par des virgules.
3. Appuyez sur **END LINE** pour calculer le résultat.

La suite de ce chapitre groupe les fonctions numériques du HP-71 selon leur utilisation.

Fonctions d'altération des nombres (ABS, IP, FP, INT, FLOOR, CEIL)

Le tableau ci-après indique la valeur donnée par chaque fonction à partir d'une expression x .

Fonction d'altération des nombres

Fonction et argument	Signification	Exemple avec résultat
ABS(x)	Valeur absolue de x .	ABS(-235) 235
IP(x)	Partie entière de x (à la droite de la virgule).	IP(10/3) 3
FP(x)	Partie fractionnaire de x (à la gauche de la virgule et y compris celle-ci).	FP(10/3) .3333333333
INT(x)	Plus grand entier inférieur ou égal à x .	INT(-7.23) -8
FLOOR(x)	Plus grand entier inférieur ou égal à x (fonction identique à INT(x)).	FLOOR(7.23) 7
CEIL(x)	Plus petit entier supérieur à x .	CEIL(7.23) 8

Remarquez les différences entre IP, FLOOR (ou INT) et CEIL. Avec un argument positif, IP et FLOOR donnent le même résultat; avec un argument négatif, IP et CEIL donnent le même résultat.

Conversions décimale et hexadécimale (DTH#, HTD)

Vous ne pouvez pas utiliser ces deux fonctions en mode CALCUL. Pour utiliser le résultat de HTD en mode CALCUL, passez en mode BASIC (f [CALC]), exécutez la fonction pour obtenir le résultat, revenez en mode CALCUL puis frappez () pour afficher automatiquement le même résultat. Vous pouvez utiliser ce dernier dans d'autres calculs.

Conversions décimale et hexadécimale

Fonction et argument	Signification	Exemple avec résultat
DTH#	Convertit un nombre décimal positif inférieur ou égal à 16^5-1 (1048575) en une chaîne représentant sa valeur hexadécimale à cinq chiffres.	DTH#(16^5-1) FFFFF
HTD	Convertit une valeur hexadécimale de un à cinq chiffres en un nombre décimal. La valeur hexadécimale doit être entrée sous forme d'une chaîne alphanumérique.	HTD("AC4F") 44111

Fonctions générales (PI, SQR, FACT, MAX, MIN, MOD, RMD, RED, RES, SGN)

Fonctions générales

Fonction et argument	Signification	Exemple avec résultat
PI	Approximation de π à 12 chiffres.	PI 3.14159265359
SQR(x)	Racine carrée positive de x .	SQR(16.10) 4.01248052955
FACT(x)	Factorielle de l'entier positif x .	FACT(253) 5.17346099264E499
MAX(x, y)	Maximum de deux valeurs.	MAX(4.5, 4.67) 4.67
MIN(x, y)	Minimum de deux valeurs.	MIN(-3, -2.999999) -3
MOD(x, y)	Réduction de x modulo y , c'est-à-dire $x - y * \text{INT}(x/y)$.	MOD(-11, 3) 1 MOD(11, 3) 2
RMD(x, y)	Reste de x/y , c'est-à-dire $x - y * \text{IP}(x/y)$.	RMD(-11, 3) -2 RMD(11, 3) 2
RED(x, y)	Réduction de x par y , c'est-à-dire $x - y * n$, où n est l'entier le plus proche de x/y .	RED(-11, 3) 1 RED(11, 3) -1
RES	Valeur de la dernière expression exécutée.	5+2 7 RES-2 5
SGN(x)	Signe de x . Donne 1 si x est positif, 0 si x est nul et -1 si x est négatif.	SGN(-5) -1

Fonctions logarithmiques (LGT, LOG, EXP, EXPONENT, LOGP1, EXPM1)

Fonctions logarithmiques

Fonction et Argument	Signification	Exemple avec résultat
LGT(x) ou LOG10(x)	$\log x$. Logarithme en base 10 d'un nombre positif.	LGT(1000) 3
LOG(x) ou LN(x)	$\ln x$. Logarithme naturel (base e) d'un nombre positif.	LOG(26) 3.25809653802
EXP(x)	e^x . Exponentielle.	EXP(1) 2.71828182846
EXPONENT(x)	Exposant de x normalisé.	EXPONENT(123456789) 8
LOGP1(x)	$\ln(1+x)$ (LOG(1+x)). Fonction utilisée pour l'évaluation précise de LOG(x) lorsque x est très proche de 1.	LOGP1(1.2345E-10) 1.23449999992E-10
EXPM1(x)	$e^x - 1$ (EXP(x) - 1). Fonction utilisée pour l'évaluation précise de EXP(x) lorsque x est très proche de 0.	EXPM1(.0001) 1.00005000167E-4

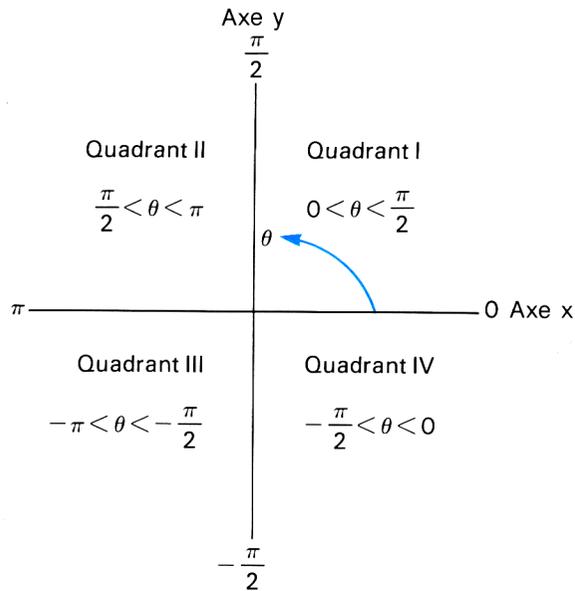
Les fonctions hyperboliques, hyperboliques inverses et certains calculs financiers utilisent les expressions $\ln(1+x)$ et $(e^x)-1$ avec des arguments proches de zéro. Vous pouvez donc utiliser directement les fonctions LOGP1 et EXPM1 pour effectuer ces calculs avec toute la précision souhaitée.

Unités d'angle (RADIANS, DEGREES)

Après une ré-initialisation de la mémoire, le HP-71 suppose que les angles sont exprimés en degrés. Si vous voulez utiliser le radian comme unité d'angle, exécutez l'ordre RADIANS (ou OPTION ANGLE RADIANS) en mode BASIC. Lorsque vous voulez revenir en mode degrés, exécutez DEGREES (ou OPTION ANGLE DEGREES). *Remarquez que ces ordres ne convertissent pas d'arguments d'une unité dans l'autre.* Une telle conversion est faite par les fonctions DEG et RAD décrites ci-après.

Fonctions trigonométriques (SIN, COS, TAN, ASIN, ACOS, ATAN, DEG, RAD, ANGLE)

Le HP-71 possède 9 fonctions trigonométriques prédéfinies. Souvenez-vous bien des intervalles de valeurs que les fonctions inverses (arc sinus, arc cosinus et arc tangente) donnent, dans les quadrants I à IV. En supposant l'ordinateur en mode radians, le HP-71 représente les angles de la façon suivante:



Fonctions trigonométriques

Fonction et Argument	Signification	Exemple avec résultat
SIN(x)	Sinus de x.	SIN(PI/2) 1
COS(x)	Cosinus de x.	COS(0) 1
TAN(x)	Tangente de x.	TAN(PI/4) 1
ASIN(x) ou ASH(x)	Arc sinus de x, où $-1 \leq x \leq 1$. Dans le quadrant I ou IV.	ASIN(1) 1.57079632679
ACOS(x) ou ACS(x)	Arc cosinus de x, où $-1 \leq x \leq 1$. Dans le quadrant I ou II.	ACOS(1) 0
ATAN(x) ou ATN(x)	Arc tangente de x. Dans le quadrant I ou IV.	ATAN(1) .785398163397
DEG(x)	Conversion de radians en degrés.	DEG(PI) 180
RAD(x)	Conversion de degrés en radians.	RAD(45) .785398163397
ANGLE(x, y)	Arc tangente de y/x, dans le quadrant correct; c'est-à-dire l'angle entre (x,y) et l'axe des abscisses positives.	ANGLE(SQR(3), 1) .523598775598

Il y a une très grande différence entre les fonctions ANGLE et ATAN. La première utilise deux arguments pour calculer l'arc tangente de leur quotient dans le *quadrant correct* et la seconde donne la valeur principale de l'arc tangente, c'est-à-dire la valeur dans le quadrant I ou IV pour un seul argument. Exemple: ANGLE(-3, -2) donne -146.309932474 degrés dans le quadrant III et ATAN(-2/-3) donne 33.690067526 degrés dans le quadrant I.

Nombres aléatoires (RND, RANDOMIZE)

La fonction RND, qui n'utilise pas d'argument, génère le nombre suivant R dans une séquence de nombres pseudo-aléatoires tels que $0 \leq R < 1$. Chaque exécution de RND donne un nouveau nombre aléatoire. Le nombre de départ d'une séquence détermine l'entière séquence de valeurs que retourne RND.

RANDOMIZE [*expression numérique*]

Pour définir le nombre de départ du générateur de nombres aléatoires:

- Exécutez `RANDOMIZE` seul pour indiquer au HP-71 de générer le nombre de départ basé sur la valeur courante de l'horloge.
- Spécifiez une constante ou une expression dans l'intervalle de valeurs du HP-71 comme paramètre de l'ordre `RANDOMIZE`. Le HP-71 utilise alors la valeur de cette expression pour définir le nombre de départ de la séquence (la valeur zéro donne une séquence composée uniquement de zéros).

Par exemple, si vous exécutez `RANDOMIZE 423`, la fonction `RND` donne `.629385058782` lors de sa première exécution.

Après une ré-initialisation de la mémoire, l'exécution de `RND` sans `RANDOMIZE` donne une séquence de nombres spécifiques commençant à `.529199358633`. Si vous voulez une séquence différente, vous devez exécuter `RANDOMIZE` avant `RND`.

Utilisez la formule suivante pour générer des entiers aléatoires, $i_1, i_2, \dots, i_j, \dots$, tels que $S \leq i_j \leq L$, où S et L représentent deux nombres réels quelconques.

$$i_j = \text{IP}((L+1-S)*\text{RND}+S)$$

Exemple: Pour illustrer la règle donnée ci-dessus, entrez une expression `RND` qui donne un nombre aléatoire dans l'intervalle 1 à 100, bornes comprises.

Entrée/Résultat

```
IP(100*RND+1) END LINE
```

53

Ceci est le premier nombre retourné après une ré-initialisation de la mémoire et avant l'exécution de `RANDOMIZE` (après une ré-initialisation de la mémoire, le HP-71 est en mode BASIC).

Ce générateur de nombres aléatoires donne de bons résultats statistiques si la taille de l'échantillon considéré est statistiquement significative.*

* Le générateur de nombres aléatoires du HP-71 satisfait au test spectral. Donald E. Knuth, *The Art of Computer Programming* (Massachusetts, 1969), vol.2, Chapitre 3.4.

Format des nombres

Les nombres sont toujours stockés dans le HP-71 avec 12 chiffres, mais vous pouvez les afficher dans quatre formats différents: `STD`, `FIX n` , `SCI n` et `ENG n` . Le paramètre n indique le nombre de chiffres de la partie fractionnaire pour `FIX n` ou le nombre de chiffres significatifs moins un pour `SCI n` et `ENG n` . Les résultats de $100 \div 2$ dans les quatre formats sont:

```
STD:    50
FIX 2:  50.00
SCI 2:  5.00E1
ENG 2:  50.0E0
```

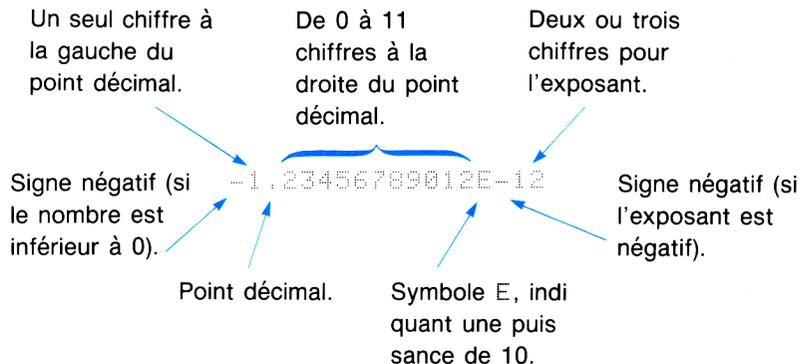
Chacun de ces formats de nombre est décrit plus en détails après la présentation de la notation exponentielle.

Notation exponentielle (E)

La notation exponentielle, ou scientifique, est un système permettant d'exprimer aisément des nombres trop grands ou trop petits pour apparaître normalement sur l'affichage, c'est-à-dire des nombres qui ne peuvent pas s'écrire en douze chiffres. Le nombre

— .0000000000123456789012

en notation exponentielle est:



Les représentations exponentielles comportent deux parties: la *mantisse*, qui compose les chiffres significatifs et l'*exposant*, qui est une puissance entière de dix.

Vous pouvez entrer les nombres sous toute forme. Néanmoins, le HP-71 n'affiche un nombre en notation exponentielle que lorsque son format le justifie comme le montrent les exemples suivants.

Exemple: Exécutez `FIX 2` en mode BASIC.

Entrée/Résultat

`1E12-1000` `END LINE`

999999999000.

En format `FIX 2`, ce nombre est affiché sans notation exponentielle car il est inférieur à 10^{12} .

`1E12+1000` `END LINE`

1.00E12

Les nombres supérieurs à $1 \times 10^{12} - 1$ sont toujours affichés en notation exponentielle.

Format d'affichage standard (STD)

STD

En format d'affichage standard, le HP-71 affiche le plus petit nombre possible de chiffres compatible avec le maximum de précision. Le HP-71 affiche le résultat de $1/2$ sous la forme `.5` alors qu'il affiche celui de $1/3$ sous la forme `.333333333333`. Il affiche en outre en notation exponentielle les nombres trop grands ou trop petits pour apparaître avec une précision maximum sans exposants.

Notation à décimale fixe (FIX)

`FIX` *nombre de chiffres*

En format d'affichage à virgule fixe, le HP-71 affiche les nombres arrondis au nombre spécifié de chiffres après la virgule. L'intervalle de valeurs pour le paramètre *nombre de chiffres* va de 0 à 11 bornes comprises. Le HP-71 affiche en format scientifique les nombres trop grands ou trop petits pour être affichés dans le format fixe courant. En format d'affichage `FIX 2`, le résultat de $1/3$ est affiché sous la forme `0.33`.

Notation scientifique (SCI)

`SCI` *nombre de chiffres*

En format d'affichage scientifique, le HP-71 affiche les nombres avec un exposant. La mantisse du nombre contient le nombre de chiffres spécifiés (*nombre de chiffres*) après la virgule et l'exposant se compose du nombre de chiffres nécessaires à la représentation correcte du nombre. Le paramètre peut se trouver entre 0 et 11 bornes comprises. En format d'affichage `SCI 2`, le résultat de $1/3$ est affiché sous la forme `3.33E-1`.

Notation ingénieur (ENG)

ENG nombre de chiffres

En format d'affichage ingénieur, le HP-71 affiche les nombres de façon similaire au format scientifique mais les exposants sont toujours des multiples de trois et le paramètre *nombre de chiffres* correspond au nombre de chiffres significatifs moins un. Le paramètre peut se trouver entre 0 et 11 bornes comprises. En format d'affichage ENG 2, le résultat de $1/3$ est affiché sous la forme $333.3E-3$.

Précision numérique (OPTION ROUND)

Le HP-71 effectue tous les calculs internes sur 15 chiffres significatifs puis arrondit le résultat de ces calculs à 12 chiffres pour le stockage et l'affichage. Cet arrondi a lieu selon l'une des quatre méthodes proposées par l'ordre OPTION ROUND:

- OPTION ROUND NEAR arrondit les valeurs aux 12 chiffres les plus proches du résultat interne à 15 chiffres. Lorsque ce critère ne suffit pas, le HP-71 arrondit au *nombre dont le dernier chiffre est pair*. OPTION ROUND NEAR est actif par défaut après une ré-initialisation. Les nombres REAL (page 57) de plus de 12 chiffres sont toujours arrondis selon l'option OPTION ROUND NEAR quelle que soit l'option active. Par exemple, si vous entrez le nombre de treize chiffres 1.234567890125, le HP-71 affiche 1.23456789012. Le nombre n'est pas arrondi pour donner un 3 comme dernier chiffre; l'option OPTION ROUND NEAR impose l'arrondi avec le dernier chiffre pair.
- OPTION ROUND ZERO arrondit les valeurs vers zéro.
- OPTION ROUND POS arrondit à la valeur supérieure.
- OPTION ROUND NEG arrondit à la valeur inférieure.

Les résultats de calculs stockés dans des variables de type SHORT et REAL sont arrondis conformément à l'option courante. Les résultats stockés dans des variables de type INTEGER sont arrondis au chiffre le plus proche. Les variables en précision REAL, SHORT et INTEGER sont présentées à la page suivante et aussi au chapitre 3 sous le titre «Déclarations des tableaux».

Si le format courant provoque l'affichage de moins de 12 chiffres, les résultats sont toujours arrondis à la valeur affichable la plus proche. Si ce critère ne suffit pas (lorsqu'un cinq se trouve après la dernière position affichable par exemple), le HP-71 arrondit au chiffre supérieur en valeur absolue.

Entrée/Résultat

FIX 1 END LINE
-1.25

Le critère de la valeur la plus proche ne suffit pas.

-1.3

Arrondi au chiffre supérieur en valeur absolue.

Précision des variables numériques (REAL, SHORT, INTEGER)

Lorsque vous déclarez des variables, vous pouvez non seulement en indiquer le nom (chapitre 3) mais aussi la précision, c'est-à-dire le nombre de chiffres utilisés par le HP-71 pour en stocker la valeur. Cette caractéristique vous permet par exemple d'économiser la mémoire lors de la manipulation de tableaux. Le HP-71 vous permet d'utiliser trois types de précision:

- **REAL.** Le HP-71 stocke ces nombres en pleine précision. L'intervalle de leurs valeurs va de $-MAXREAL$ à $+MAXREAL$. La représentation interne des nombres **REAL** comporte 12 chiffres pour la mantisse et 3 chiffres pour l'exposant.
- **SHORT.** Le HP-71 stocke ces nombres en demi-précision. Les valeurs sont réduites à l'intervalle -9.9999×10^{499} à 9.9999×10^{499} . La représentation interne des nombres **SHORT** comporte 5 chiffres pour la mantisse et 3 pour l'exposant.
- **INTEGER.** Le HP-71 stocke ces nombres sous formes d'entiers. Les valeurs sont réduites à l'intervalle -99999 à $+99999$. La représentation interne des nombres **INTEGER** comporte 5 chiffres pour la mantisse et pas d'exposant.

Exceptions mathématiques (IVL, DVZ, OVF, UNF, INX)

Lors des calculs, diverses opérations peuvent donner des résultats inusuels selon les valeurs des termes concernés. Ces exceptions comprennent les racines carrées de nombres négatifs, les divisions par zéro, les résultats trop grands ou trop petits pour être représentés par le HP-71 et les résultats que le HP-71 ne peut pas représenter exactement dans un format à virgule fixe avec 12 chiffres. A chacune de ces exceptions est affectée un indicateur que le HP-71 arme lorsque cette condition apparaît. Ces indicateurs restent armés jusqu'à ce que vous les effaciez. Vous pouvez accéder à ces indicateurs par leurs numéros ou par leurs noms et vous pouvez les effacer comme tous les autres indicateurs.

Pour plus d'informations sur les indicateurs, consultez le chapitre 11. La section «Proposition IEEE pour le traitement des exceptions mathématiques» du *Manuel de référence* indique en détails les conditions pour lesquelles le HP-71 arme les indicateurs des exceptions.

Exceptions mathématiques

Exception	Indicateur		Exemple
	Nom	Numéro	
Opération invalide	IVL	-8	ACOS(2), LOG(-23), (-14)^(-1/3)
Division par zéro	DVZ	-7	187/0, TAN(90)
Dépassement supérieur de capacité	OVF	-6	FACT(254), 10*1E499
Dépassement inférieur de capacité	UNF	-5	EXP(-1149), 1/3E499
Résultat inexact	INX	-4	1/3, 1+1E-50

Traitement des exceptions mathématiques (DEFAULT ON, DEFAULT OFF, DEFAULT EXTEND)

Le HP-71 vous propose trois méthodes pour traiter les exceptions mathématiques:

- **DEFAULT ON** est actif après une ré-initialisation de la mémoire. Avec cette option, les divisions par zéro et les dépassements provoquent l'affichage d'un message d'avertissement et le calcul continue en utilisant des valeurs par défaut. L'exception **INV** arrête le calcul.
- **DEFAULT OFF**. Lorsque cette option est active, toutes les exceptions sauf un résultat inexact provoquent l'affichage d'un message d'erreur et l'arrêt des calculs. Vous pouvez alors utiliser l'ordre **ON ERROR** (page 172) pour traiter ces erreurs.
- **DEFAULT EXTEND**. Lorsque cette option est active, le HP-71 fournit des valeurs par défaut spéciales. Cette option est décrite plus en détails dans la section suivante.

Quelle que soit l'option choisie, les résultats inexacts sont toujours arrondis selon l'option d'arrondi active (page 56).

En supposant l'option **DEGREES** active, les conditions **DEFAULT ON** et les valeurs par défaut sont résumées dans le tableau suivant:

Valeurs par défaut pour les exceptions mathématiques
(DEFAULT ON actif)

Numéro (ERRN) et exception	Condition	Valeur par défaut (option DEGREES)
1, UNF	Dépassement inférieur; c'est-à-dire résultat non-nul entre -EPS et +EPS.	0
2, OVf	Dépassement supérieur: <ul style="list-style-type: none"> • Pour variables INTEGER. • Pour variables SHORT. • Pour variables REAL. 	±99999 ±9.9999E499 ±9.9999999999999E499
3, DVZ	EXPONENT(0)	-9.9999999999999E499
4, DVZ	TAN est infini à cause d'un argument égal à 90° ou à un multiple de 90°.	±9.9999999999999E499
5, DVZ	Zéro à une puissance négative.	±9.9999999999999E499
6	Zéro à une puissance nulle.*	1
12, DVZ	LN(0)	-9.9999999999999E499

* 0^0 et Inf^0 n'arme pas l'indicateur d'exception mais arrêtent les calculs si l'option **DEFAULT OFF** est active.

Proposition IEEE pour le traitement des exceptions mathématiques (+Inf, -Inf, INF, NAN, NaN, TRAP, ?)

Au moment de la conception du HP-71, la société de normalisation internationale IEEE n'avait pas encore terminé sa norme pour les calculs arithmétiques avec virgule flottante. Les deux aspects principaux de la proposition IEEE concernant l'arithmétique décimale sont la précision des résultats arithmétiques et le traitement des exceptions. Le HP-71 satisfait à la proposition IEEE telle qu'elle existait lors de la conception de l'ordinateur.

Chaque indicateur d'exception mathématique *trappe* l'exception correspondante et indique l'action à prendre suivant le tableau ci-après.

Actions correspondant aux valeurs de trappe des exceptions mathématiques

Valeur de trappe	Action de trappe
0	Interrompt l'exécution avec un message d'erreur.
1	Pour UNF, OVZ et DVZ, fournit les valeurs par défaut indiquées dans le tableau ci-dessus. Pour IVL, interrompt l'exécution avec un message d'erreur. Pour INX, donne le résultat arrondi.
2	Donne les valeurs par défaut de IEEE.

TRAP est une fonction qui donne la valeur courante d'une trappe ou définit une nouvelle valeur de trappe pour un indicateur d'exception mathématique spécifié.

TRAP(*n° d'indicateur*, *nom d'exception* [, *nouvelle valeur de trappe*])

Exemple:

TRAP(DVZ, 0)

Indique au HP-71 de suspendre l'exécution et d'afficher un message d'erreur en réponse aux divisions par zéro.

TRAP(DVZ, 1)

Indique au HP-71 d'utiliser la valeur par défaut 9.999999999999999E499 en réponse aux divisions par zéro.

TRAP(DVZ, 2)

Indique au HP-71 d'utiliser la valeur par défaut IEEE (Inf ou -Inf) en réponse aux divisions par zéro.

Le tableau suivant indique les valeurs de trappe données par les trois options DEFAULT pour chacune des cinq exceptions mathématiques.

Valeurs de trappe données par les options DEFAULT

Défaut	IVL	DVZ	OVF	UNF	INX
DEFAULT OFF	0	0	0	0	1
DEFAULT ON	1	1	1	1	1
DEFAULT EXTEND	1	2	2	2	2

Les réponses spéciales aux valeurs de trappe 2 sont $\pm Inf$ (infini) et NaN (pas un nombre - Not a Number). Inf est la valeur TRAP 2 donnée pour un dépassement supérieur (OVF) ou une division par zéro (DVZ) et NaN est la valeur TRAP 2 donnée pour une opération invalide (INV). INF est une fonction sans argument qui donne Inf, lequel se comporte comme une valeur infinie dans les calculs suivants. NAN est aussi une fonction sans argument qui donne NaN, que vous pouvez utiliser pour initialiser des données non-initialisées de façon à armer l'indicateur IVL lorsque ces données sont utilisées dans un calcul. La section «*Proposition IEEE pour le traitement des exceptions mathématiques*» dans le *Manuel de référence* décrit plus en détails les fonctions INF et NAN et les réponses TRAP 2.

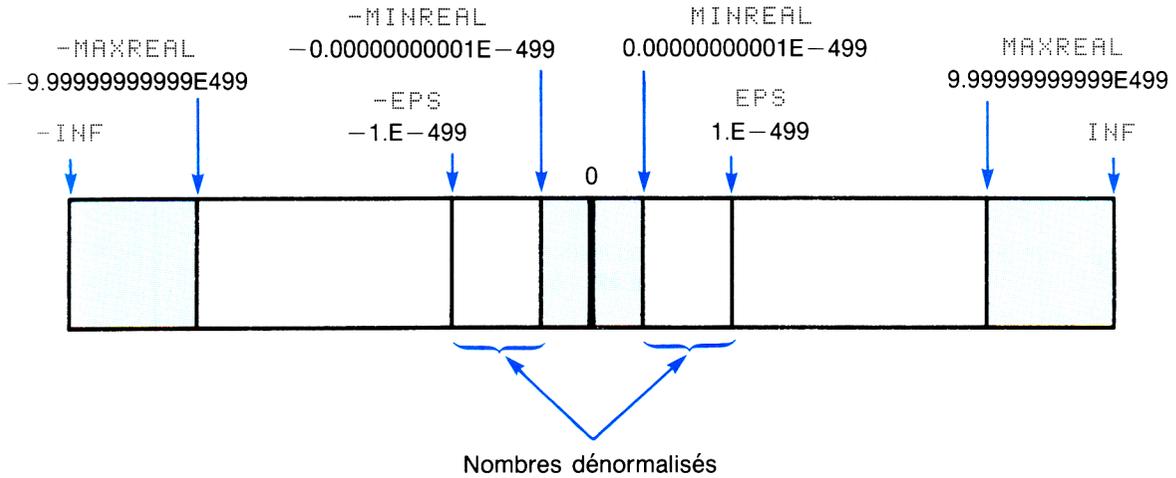
Le *Manuel de référence* présente en plus l'opérateur ? qui donne 1 lorsque l'une ou les deux expressions comparées sont NaN.

Catégories de nombres (CLASS)

La présence des valeurs par défaut TRAP 2 pour les exceptions mathématiques étend l'intervalle normal et le type des nombres. Cet intervalle étendu est divisé en six classes. La classe 3 comprend les nombres normalisés de EPS à MAXREAL (bornes comprises). Les cinq autres couvrent zéro, les nombres dénormalisés (entre zéro et EPS), l'infini et NaN. La fonction CLASS donne un nombre avec signe indiquant la classe et le signe de l'argument. Cette fonction est conçue principalement pour le contrôle de programme. Vous trouverez plus de détails sur la fonction CLASS dans le *Manuel de référence*.

Intervalle de valeurs des nombres (MINREAL1, EPS, MAXREAL)

Le diagramme suivant illustre l'intervalle des valeurs que vous pouvez entrer et stocker:



Tous les opérandes numériques sont représentés par un *signe*, une *mantisse* de 12 chiffres et un *exposant* de dix entre -499 et 499 (bornes comprises). Le premier chiffre de la plupart des nombres est non nul; ces nombres sont dits *normalisés*. Par exemple, le nombre -1234.56 est affiché en format SCI 11 sous la forme $-1.234560000000E3$. Le plus petit nombre normalisé est appelé EPS ($1.000000000000E-499$). Le plus grand nombre normalisé est appelé MAXREAL ($9.999999999999E499$).

Le HP-71 affiche des nombres très petits, dont les exposants normaux sont inférieurs à -499 , comme des nombres *dénormalisés*, avec un ou plusieurs zéros. Par exemple, lorsque la valeur de trappe (page 59) de l'indicateur de dépassement inférieur (page 57) vaut 2 ($\text{TRAP}(\text{UNF}, 2)$), $\text{EPS}/1000$ est affiché en format SCI 11 sous la forme $0.001000000000E-499$. Le plus petit nombre positif dénormalisé est appelé MINREAL ($0.000000000001E-499$). Les valeurs inférieures donnent généralement un dépassement inférieur et sont remplacées par zéro.

Les nombres entrés ou les résultats inférieurs au plus petit nombre positif normalisé du HP-71 (EPS) peuvent provoquer un dépassement inférieur. Ces conditions interrompent un calcul avec un message d'erreur ou indiquent au HP-71 de continuer le calcul avec diverses valeurs par défaut. Ces erreurs et valeurs par défaut sont traitées en pages 57 à 60.

Opérateurs de relation (combinaisons de <, =, >, #, ?)

Les opérateurs de relation comparent les valeurs de deux expressions numériques ou alphanumériques et donnent 1 si la comparaison est vraie et 0 si elle est fausse (les valeurs alphanumériques sont présentées au chapitre 3, «Variables: simples et en tableaux»). L'opérateur de relation ? est décrit dans le *Manuel de référence*, au chapitre intitulé «Proposition IEEE pour le traitement des exceptions mathématiques».

Exemples d'opérateurs de relations

Opérateur de relation	Définition	Exemple avec résultat
>	Supérieur à?	1>-1 1
>=	Supérieur ou égal à?	3.14>=PI 0
<	Inférieur à?	3.14<PI 1
<=	Inférieur ou égal à?	-EPS<=0 1
<>	Inférieur à ou supérieur à?	3<>3 5<>2 0 1 NAN<>NAN 0
#	Différent de ?	3#3 5#2 0 1 NAN<>NAN 1
?	Numérique?	45?-12 8?NAN 0 1

Le signe = sert pour les instructions d'affectation de variables et comme opérateur de relation. Lorsqu'une entrée au clavier peut être interprétée des deux façons, le HP-71 suppose que l'entrée est une affectation de variable.

Opérateurs logiques (AND, OR, EXOR, NOT)

Les quatre opérateurs logiques utilisent des arguments binaires et donnent des résultats binaires. Les opérateurs interprètent tous les opérandes numériques non nuls comme égaux à 1, ou vrais et les opérandes nuls comme égaux à 0, ou faux. Les opérateurs AND, OR et EXOR donnent 1 comme résultat si la relation entre les opérandes est vraie et 0 si elle est fausse. NOT est un opérateur monadique et donne la valeur opposé (0 ou 1) d'un opérande seul.

Opérateurs logiques

Opérateur Logique	Définition	Exemple avec résultat
AND	Les deux expressions vraies (non nulles)?	3 AND 4 1 91 AND 4-4 0
OR	Une ou les deux expressions vraies?	3 OR 0 1 0 OR 6-8+2 0
EXOR	Une des deux expression vraies mais non les deux? Equivalent à (A AND NOT B) OR (B AND NOT A).	3 EXOR 0 1 5 EXOR 5 0
NOT	Expression fausse (nulle)?	NOT (5#31 1 NOT 23 0

Vous pouvez utiliser les opérateurs logiques et de relation pour comparer des constantes numériques (3 OR 0), des variables (A AND B), des fonctions (SIN(A) AND COS(A)) et des expressions.

Exemple: Si A = 0 and B = 20:

Entrée/Résultat

NOT A AND (B<50-B)

Entrez une expression avec des opérateurs arithmétiques, logiques et de relation.

1

L'expression est vraie.

Hiérarchie des opérateurs

La liste ci-dessous donne les opérateurs du HP-71 dans leur ordre hiérarchique, de la plus haute priorité à la plus basse.

Le HP-71 exécute en premier les opérateurs de plus haute priorité et évalue de gauche à droite les expressions de même priorité.

1. Expressions entre parenthèses. Les parenthèses emboîtées sont évaluées de l'intérieur vers l'extérieur.
2. Fonctions, telles que SIN, LOG et FACT.
3. \wedge .
4. Signe moins, opérateur NOT. Le signe moins devant une variable change le signe du contenu de celle-ci, qu'il soit positif ou négatif.
5. *, /, %, DIV.
6. +, -.
7. Opérateurs de relation: combinaisons de <, =, >, # et ?.
8. Opérateur AND.
9. Opérateurs OR et EXOR.

Variables: simples et en tableaux

Table des matières

Présentation	66
Caractéristiques des variables et des tableaux	66
Partage des variables entre clavier et programmes	67
Récupération de la mémoire (DESTROY)	67
Variables numériques: simples et en tableaux	68
Définition de la limite basse des tableaux (OPTION BASE)	68
Déclaration des tableaux (DIM, REAL, SHORT, INTEGER)	69
Chaînes alphanumériques	71
Chaînes entre guillemets	72
Variables alpha: simples et en tableaux (DIM, OPTION BASE)	72
Concaténation de chaînes (&)	73
Sous-chaînes	73
Fonctions de chaînes (LEN, POS, VAL, STR\$, NUM, CHR\$, UPRC\$)	74

Présentation

Ce chapitre couvre les sujets suivants:

- Caractéristiques particulières du langage du HP-71 concernant les chaînes entre guillemets, les fonctions de chaînes et les variables numériques et alphanumériques.
- Variables simples numériques et alphanumériques.
- Variables en tableaux, numériques et alphanumériques.
- Manipulation de chaînes.

Caractéristiques des variables et des tableaux

Le langage du HP-71 comprend certaines caractéristiques qui sont peut-être nouvelles à ceux ayant travaillé avec d'autres versions du BASIC. Les plus importantes sont listées ci-dessous avec les références des pages où elles sont traitées.

- Les indices de dimensions des chaînes et tableaux peuvent être des expressions.
- Les fonctions VAL et STR\$ évaluent des expressions numériques et donnent les résultats sous forme de valeurs numériques ou de chaînes (page 76).
- Les variables sont communes au clavier et aux programmes (voir ci-après).
- Le HP-71 donne automatiquement les valeurs de variables par défaut (0 ou "") sans message d'avertissement lorsque vous utilisez une variable non affectée au clavier ou dans un programme (page 68).
- Vous pouvez détruire toutes les variables et récupérer la mémoire qu'elles utilisent (page 67).
- Une variable simple et un tableau ne peuvent pas porter le même nom (page 68).
- Le symbole @ concatène plusieurs instructions sur une seule ligne (page 146).
- Les instructions d'affectation multiple (telles que A,B,C=100) ne sont pas autorisées. Une même ligne de programme peut cependant affecter plusieurs variables en utilisant la concaténation d'instructions (exemple A=100 @ B=100 @ C=100 @ D=35).
- Le mot-clé LET peut être omis des instructions d'affectation.
- Le mot-clé DISP peut être omis des instructions d'affichage sauf après THEN et ELSE (page 226).
- Vous pouvez utiliser l'apostrophe (') ou le guillemet (") dans les ordres DISP et PRINT.

Partage des variables entre clavier et programmes

- Lorsque vous exécutez un programme, les variables utilisées par ce dernier peuvent contenir des valeurs affectées préalablement au clavier ou par un autre programme.
- Si vous ne voulez pas que les variables d'un programme puissent contenir des valeurs venant d'ailleurs, vous pouvez détruire toutes les affectations de variables avec l'ordre DESTROY ou, pour les variables simples seulement, dimensionner les variables avec les ordres DIM, REAL, SHORT, INTEGER. Vous pouvez utiliser ces mots-clés dans les programmes ou au clavier. DESTROY est la seule façon de vous assurer que les éléments des tableaux ne contiendront pas de valeurs antérieures.

Récupération de la mémoire (DESTROY)

L'ordre DESTROY vous permet de récupérer toute la mémoire affectée à une variable, à un groupe de variables ou à toutes les variables.

```
DESTROY variable [, variable...]
DESTROY ALL
```

Exemples:

```
DESTROY C4#,W,R2
```

Les variables C4#, W et R2 n'existent plus, le HP-71 a récupéré l'espace mémoire qui leur était préalablement affecté et leurs noms sont maintenant disponibles pour d'autres utilisations.

```
DESTROY ALL
```

Toutes les variables sont détruites et toute la mémoire qui leur était affectée peut être utilisée à d'autres fins.

Variables numériques: simples et en tableaux

Un nom de variable ou de tableau se compose d'une lettre seule ou d'une lettre suivie d'un chiffre de 0 à 9; exemple: A, W3, F7. Les variables numériques simples et en tableaux utilisent les mêmes noms. Par conséquent, si vous utilisez pour une variable un nom, tel que A, vous ne pouvez pas utiliser ce même nom pour un tableau.

On appelle variable nonexistante, une variable à laquelle vous n'avez pas affecté de valeur ou qui n'a pas été déclarée par un ordre INTEGER, SHORT ou REAL, au clavier ou par programme. Si vous essayez de rappeler une variable inexistante, le HP-71 donne zéro sans message ni bip mais ne crée pas la variable.

Si vous essayez de rappeler un élément d'un tableau inexistant dont les numéros de rang et de colonne sont dans l'intervalle des dimensions par défaut (0 à 10), le HP-71 donne la valeur zéro sans message ni bip.

Si vous spécifiez un nom de tableau existant avec un numéro de rang ou de colonne supérieur à celui défini lors de la déclaration, le HP-71 donne une erreur `Subscript` et arrête le programme (`TRAP<IVL,0>` ou `TRAP<IVL,1>` actif) ou donne un avertissement `Subscript`, donne la valeur `NaN` et continue le programme (`TRAP<IVL,2>` actif).

Définition de la limite basse des tableaux (OPTION BASE)

Tous les indices des limites des tableaux du HP-71 commencent à 0 ou 1, selon l'option de base au moment de la création du tableau (`OPTION BASE 0`, `OPTION BASE 1`).

```
OPTION BASE 0
OPTION BASE 1
```

Un ordre `OPTION BASE` exécuté au clavier ou dans un programme reste actif jusqu'à l'exécution d'un autre ordre `OPTION BASE` ou la ré-initialisation de la mémoire. Le paramètre de l'ordre `OPTION BASE` peut être une expression numérique quelconque dont le résultat vaut 0 ou 1.

Exemple: Le segment de programme suivant illustre l'action de l'ordre `OPTION BASE`.

```
10 OPTION BASE 0
```

Tout tableau déclaré après l'exécution de cette instruction aura 0 (ou 0,0) comme indice(s) inférieur(s).

```
20 DIM A(5,5)
```

Le tableau `A` a donc 6 rangs (numérotés de 0 à 5) et 6 colonnes et comporte 36 éléments.

```
30 OPTION BASE 1
```

Tout tableau déclaré après l'exécution de cette instruction aura 1 (ou 1,1) comme indice(s) inférieur(s).

```
40 DIM B(5,5)
```

Le tableau `B` a donc 5 rangs (numérotés de 1 à 5) et 5 colonnes et comporte 25 éléments.

```
:
```

L'exécution de la ligne 30 n'affecte pas la limite inférieure du tableau `A`.

Déclaration des tableaux (DIM, REAL, SHORT, INTEGER)

Une déclaration de tableau définit non seulement l'indice supérieur du tableau mais aussi la précision de ses éléments. Si le tableau n'existait pas auparavant, une déclaration initialise en-outré tous les éléments à zéro. Les mots-clés `DIM` et `REAL` déclarent tous deux des variables numériques en pleine précision. Cependant, seul `DIM` permet de déclarer des variables alphanumériques comme décrit en page 72.

— syntaxe simplifiée —

```
DIM liste de variables
```

— syntaxe simplifiée —

```
REAL liste de variables
```

syntaxe simplifiée

SHORT *liste de variables*

syntaxe simplifiée

INTEGER *liste de variables*

Exemples: Les exemples suivants supposent que les tableaux déclarés n'existent pas encore.

```
DIM A(5),S(15,15)
```

Les deux tableaux sont déclarés en pleine précision. Si on suppose `OPTION BASE 1` actif, le tableau `A` a 5 éléments et le tableau `S` en a 225. Tous les éléments de chaque tableau valent 1.

```
INTEGER C3(7,10)
```

En supposant `OPTION BASE 1` actif, le tableau `C3` est déclaré pour contenir des valeurs entières et a 88 éléments tous initialisés à zéro.

Dimensions de tableau par défaut: (10) ou (10,10). Le HP-71 affecte par défaut les dimensions (10) ou (10,10) à un tableau non-existant lorsqu'il est créé par une instruction d'affectation. Cette instruction crée le tableau en pleine précision aux dimensions par défaut, affecte la valeur spécifiée à l'élément spécifié et affecte 0 à tous les autres éléments. Si, dans l'instruction d'affectation, un indice de l'élément affecté est supérieur à zéro, le HP-71 affiche un message d'erreur.

Exemples: Dans ces exemples, on suppose les tableaux inexistantes et `OPTION BASE 0` actif.

```
D(7,3)=V
```

Les deux indices ((7,3)) étant inférieurs à 10, le HP-71 dimensionne le tableau (10,10) et lui affecte 121 éléments tous nuls sauf l'élément 7,3 qui contient la valeur de la variable `V`.

```
R(11)=4
```

L'indice du tableau dans cette instruction d'affectation étant supérieur à 10, le HP-71 affiche le message `ERR: Subscript`, ne crée pas le tableau et n'affecte pas de valeur.

Changement des dimensions d'un tableau par programme. Le HP-71 vous permet de redimensionner un tableau pendant l'exécution d'un programme. Cette caractéristique vous permet de concevoir des programmes qui adaptent automatiquement la taille des tableaux à la quantité de données. Les redimensionnements s'effectuent avec les mêmes quatre ordres qui permettent de déclarer initialement les dimensions: `DIM`, `REAL`, `SHORT` et `INTEGER`.

Nota: Lors du redimensionnement d'un tableau, vous devez déclarer la même précision que lors de la déclaration initiale, sinon toutes les valeurs du tableau seront perdues.

Si vous redimensionnez un tableau à une taille inférieure, vous perdrez certains éléments (les derniers éléments du tableau); le HP-71 conserve sinon tous les éléments existants. Les éléments de tableau étant stockés rang par rang, c'est-à-dire que l'élément du premier rang dernière colonne est suivi de l'élément du second rang première colonne, les éléments des tableaux redimensionnés peuvent apparaître ré-arrangés après un redimensionnement si vous modifiez le nombre de colonnes. Si vous redimensionnez un tableau à une taille supérieure, le HP-71 initialise tous les nouveaux éléments à zéro.

Exemple: Le tableau A est déclaré par l'instruction `DIM A(3,3)` et contient les valeurs 1 à 9 arrangées de la façon suivante:

Tableau A			
	Colonne 1	Colonne 2	Colonne 3
Rang 1	1	2	3
Rang 2	4	5	6
Rang 3	7	8	9

Après l'exécution de `DIM A(5,2)`, les valeurs du tableau sont ré-arrangées de la façon suivante:

	Colonne 1	Colonne 2
Rang 1	1	2
Rang 2	3	4
Rang 3	5	6
Rang 4	7	8
Rang 5	9	0

Le HP-71 affecte la valeur zéro au nouvel élément `A(5,2)`.

Chaînes alphanumériques

Une *chaîne alphanumérique* peut être une collection de caractères entre guillemets, une variable ou une expression représentant une telle collection. Le HP-71 reconnaît les tableaux alphanumériques unidimensionnels et dispose d'un puissant jeu de fonctions de chaîne.

Chaînes entre guillemets

Les chaînes entre guillemets peuvent utiliser les guillemets usuels (") ou des apostrophes ('). Ces deux symboles doivent néanmoins aller par paires. Les exemples de ce manuel utilisent généralement les guillemets (l'emploi d'apostrophes permet par exemple d'imprimer simplement un texte comprenant des guillemets).

```
PRINT '"La pluie et le beau temps" est de Jacques PREVERT'
```

Dans les instructions BASIC, vous pouvez omettre les guillemets autour des noms de fichiers.

Variables alphanumériques: simples et en tableaux (DIM, OPTION BASE)

Les noms de variables alphanumériques se composent d'une lettre suivie du signe # ou d'une lettre et d'un chiffre suivi du signe #; exemple: A#, C4# et J7#. Une variable alphanumérique simple et un tableau alphanumérique ne peuvent pas porter le même nom.

Valeur par défaut: la chaîne vide (""). La chaîne vide, représentée par "" ou ''', est la valeur que le HP-71 donne pour toute référence à une variable alphanumérique inexistante. La chaîne vide ne contient aucun caractère; elle n'est ni imprimée ni affichée.

Déclaration de dimensions (DIM, OPTION BASE). L'ordre DIM permet de déclarer, entre crochets, la taille d'une variable alphanumérique, c'est-à-dire le plus grand nombre de caractères (y compris les espaces) qu'elle peut contenir. Lorsque vous voulez déclarer un tableau alphanumérique, vous pouvez spécifier le nombre d'éléments entre parenthèses suivi de la taille de ces éléments entre crochets. Tous les éléments d'un tableau ont la même taille. DIM initialise tous les éléments des tableaux de variables alphanumériques égaux à la chaîne vide, sauf pour les tableaux déjà existants.

OPTION BASE permet de définir la limite basse des numéros d'éléments des tableaux alphanumériques comme pour les tableaux numériques.

Les tableaux alphanumériques ne peuvent avoir qu'une seule dimension.

— syntaxe simplifiée —

```
DIM liste de variables
```

Exemples:

```
DIM A3#[25]
```

Dimensionne une variable alphanumérique simple de 25 caractères.

```
DIM C#(15)[40]
```

En supposant OPTION BASE 1 actif, cette instruction déclare un tableau alphanumérique de 15 éléments de 40 caractères chacun et affecte la chaîne vide à ces éléments.

Dimensions par défaut. Si vous ne déclarez pas la taille d'une variable alphanumérique avec une instruction DIM, le HP-71 la définit par défaut égale à 32 caractères. L'instruction OPTION BASE courante n'a aucun effet sur la taille ou la position des chaînes.

Si vous ne déclarez pas la taille d'un tableau alphanumérique avec une instruction DIM avant son utilisation dans une instruction d'affectation, le HP-71 la définit par défaut égale à 10 éléments de 32 caractères. Le nombre d'éléments d'un tableau alphanumérique dimensionné par défaut sera 10 ou 11 selon l'instruction OPTION BASE 1 ou OPTION BASE 0 active.

Exemple: L'exécution de l'instruction suivante active OPTION BASE 0.

```
DIM F5$(4)                                Le tableau alphanumérique F5$ a 5 éléments
                                           de 32 caractères.
```

Changement des dimensions d'un tableau dans un programme. L'ordre DIM vous permet aussi de changer les dimensions d'un tableau alphanumérique existant. Vous devez conserver la longueur des éléments si vous désirez garder les valeurs de ces éléments. Si vous changez la longueur des éléments du tableau, tous les éléments deviennent des chaînes vides. Si le tableau redimensionné contient moins d'éléments, les derniers éléments du tableau initial seront perdus. Si le tableau redimensionné contient plus d'éléments, les nouveaux éléments seront initialisés égaux à la chaîne vide.

Exemple: On suppose OPTION BASE 0 actif.

```
DIM W$(R)                                Cette instruction remplace la taille du tableau
                                           W$ par la valeur courante de la variable R. La
                                           longueur des chaînes devient 32 caractères car
                                           vous n'en avez pas spécifié.
```

Concaténation de chaînes (&)

Vous pouvez joindre plusieurs chaînes alphanumériques, variables ou chaînes entre guillemets pour en former une seule à l'aide de l'opérateur de concaténation &.

Sous-chaînes

Une sous-chaîne est un groupe de caractères adjacents dans une autre chaîne. La chaîne vide peut aussi être une sous-chaîne.

Spécification de sous-chaînes. Vous pouvez spécifier la position d'une sous-chaîne dans une chaîne en indiquant son ou ses indices entre crochets après le nom de la chaîne.

Exemples:

```
A$="ALARME"(3) END LINE      Affecte à A$ la sous-chaîne constituée du troi-
                                           sième au dernier caractères de la chaîne
                                           ALARME.
```

Affectation de valeurs à des sous-chaînes. Vous pouvez affecter une expression alphanumérique quelconque directement à une sous-chaîne d'une variable.

Exemples:

```
M$(5)C7]=S$
```

Affecte le contenu de la chaîne S\$ au cinquième élément du tableau M\$ en commençant à la position sept. Les caractères de la chaîne affectée remplacent le même nombre de caractères de la chaîne réceptrice à partir de la position sept de l'élément 5. Si cet élément de M\$ avait à l'origine moins de sept caractères, le HP-71 ajouterait le nombre de blancs nécessaires entre le cinquième élément et le début de S\$.

```
V$(I,J)="voilier"
```

Cette instruction dilate ou contracte la portion de V\$ de la position I à la position J de façon que *voilier* occupe exactement cet espace. Le HP-71 supprime tous les caractères existants préalablement de la position I à la position J y compris les caractères aux positions I et J.

Fonctions de chaînes (LEN, POS, VAL, STR\$, NUM, CHR\$, UPRC\$)

Le langage BASIC du HP-71 contient un grand jeu de fonctions de chaînes qui vous permettent de créer, d'analyser et de manipuler des chaînes. Les quatre fonctions numériques suivantes analysent des chaînes et donnent un résultat numérique:

Fonctions de chaînes

Fonction	Action
LEN(<i>chaîne</i>)	Donne le nombre de caractères dans une chaîne.
POS(<i>chaîne 1</i> , <i>chaîne 2</i> [, <i>expression numérique</i>])	Donne la position de <i>chaîne 2</i> dans <i>chaîne 1</i> . L' <i>expression numérique</i> optionnelle spécifie la position de départ de la recherche.
VAL(<i>chaîne</i>)	Evalue la chaîne comme si c'était une expression numérique et donne la valeur de cette expression.
NUM(<i>chaîne</i>)	Donne le code de caractère du premier caractère de la chaîne.

Les trois fonctions de chaînes suivantes donnent un résultat alphanumérique:

Fonctions de chaînes

Fonction	Action
STR\$(<i>expression numérique</i>)	Evalue l' <i>expression numérique</i> et donne le résultat sous forme d'une chaîne.
CHR\$(<i>expression numérique</i>)	Donne le caractère dont le code est égal à la valeur de l' <i>expression numérique</i> .
UPRC\$(<i>chaîne</i>)	Convertit toutes les minuscules d'une chaîne en majuscules.

Position de sous-chaîne (POS). Cette fonction de deux ou trois arguments donne la position d'une sous-chaîne dans une chaîne. Le HP-71 cherche la position de la sous-chaîne spécifiée par le deuxième argument dans la chaîne spécifiée par le premier argument. le troisième argument, optionnel, spécifie la position de caractère à laquelle la recherche doit commencer.

Si le HP-71 ne trouve pas la sous-chaîne dans la chaîne, la fonction donne zéro. Sans le troisième argument optionnel, la fonction ne donne que la position de la première occurrence de la sous-chaîne dans la chaîne.

Entrée/Résultat

```
POS("Deauville", "ville") [END LINE]
```

La position du premier caractère de ville dans Deauville est 5.

```
POS("Deauville", "city") [END LINE]
```

Cette fonction donne 0 car la sous-chaîne n'existe pas dans la chaîne spécifiée.

```
A$="TRAVAIL" @ B$="AIL" [END LINE]
```

```
POS(A$, B$) [END LINE]
```

Cette fonction POS donne 5 indiquant que la sous-chaîne "AIL" commence à la position 5 de la chaîne "TRAVAIL".

```
POS("Deauville", "e", 3) [END LINE]
```

Cette fonction cherche le premier e dont la position est supérieure ou égale à 3.

Le HP-71 ignore le premier e, car sa position est inférieure à 3. La fonction commence la comparaison caractère par caractère avec le caractère e en position 3 et continue jusqu'à la position 9 où se trouve un autre e.

Conversion alphanumérique en numérique (VAL). Cette fonction convertit une expression alphanumérique contenant une expression numérique valide en une valeur numérique. L'expression numérique peut être constituée de variables, de fonctions et d'opérateurs. Remarquez que VAL évalue une expression alphanumérique comme si c'était une expression numérique.

En résumé, VAL évalue les expressions suivantes comme des expressions numériques:

- Chaînes alphanumériques entre guillemets, telles que "473" ou "M*23".
- Caractères représentés par des expressions alphanumériques, tels que A# ou B#&"*"&C#.

Le HP-71 ignore tous les caractères suivant la première expression numérique valide. Si le premier caractère d'une chaîne ne peut pas être interprété comme partie d'une expression numérique, le HP-71 affiche un message d'erreur.

Entrée/Résultat

```
C#="FACT(62)^0.5" @ FIX2 END LINE
```

```
VAL(C#) END LINE
```

Donne 5.61E42. La variable C# représentant une expression numérique valide, la fonction VAL(C#) évalue l'expression et donne son résultat, illustré en format d'affichage FIX 2.

Exemple: Le programme suivant illustre la puissance de la fonction VAL par le calcul d'une intégrale d'une fonction arbitraire que vous entrez au clavier, en utilisant la règle des trapèzes. Exécutez EDIT TRAPINT pour ouvrir un fichier pour ce programme, puis entrez le programme et essayez-le.

```
10 ! INTEGRATION
20 DIM F$(90),L,U,X,T,S,I
30 INTEGER N
40 INPUT "f(X)=";F$
50 INPUT "Limite inf.=";L
60 INPUT "Limite sup.=";U
70 INPUT "Nombre de TRAPEZES=";N
80 X=L @ T=VAL(F$)/2
90 S=(U-L)/N
100 FOR X=L+S TO U-S STEP S
110 T=T+VAL(F$)
120 NEXT X
130 X=U @ T=T+VAL(F$)/2
140 I=T×S
150 DISP "INTEGRALE:";I
```

L'expression que vous entrez doit utiliser X comme variable de l'intégration.

Evalue l'intégrale à la limite basse.

Evalue l'intégrale en plusieurs points au milieu.

Evalue l'intégrale à la limite haute.

Lorsque vous exécutez ce programme, les lignes 80, 110 et 130 évaluent la fonction que vous entrez en ligne 40.

Après l'entrée de ce programme, exécutez `FIX 2` puis appuyez sur `[RUN]`. Lorsque le HP-71 affiche `f(X)=`, entrez `X^2+X`. Remarquez que la fonction doit utiliser `X` en tant que variable indépendante. Répondez ensuite aux autres messages en entrant 1 comme limite basse, 5 comme limite haute et 10 comme nombre de trapèzes utilisés pour calculer l'approximation de l'intégrale. Le HP-71 affiche alors:

```
INTEGRALE: 53.44
```

Approximation de l'intégrale.

L'accroissement du nombre de trapèzes augmente la précision de l'approximation mais allonge le temps de calcul.

Conversion numérique en alphanumérique (STR#). Cette fonction *évalue* une expression numérique et convertit le résultat en une chaîne, selon le format d'affichage courant.

Entrée/Résultat

```
STR#(2+6/7) [END LINE]
```

```
2.86
```

`STR#` évalue tout d'abord l'expression `2+6/7` puis convertit le résultat en une chaîne (illustrée en format d'affichage `FIX 2`). La chaîne `2.86` affichée ne peut pas être utilisée dans les calculs.

Conversion d'un caractère en son code (NUM). Le HP-71 utilise un jeu de 256 caractères dont vous pouvez trouver un tableau dans le *Manuel de référence*. Chaque caractère possède un code unique (0 à 255). 95 d'entre-eux (codes 32 à 126) sont les caractères standard d'impression définis par la norme ASCII*. La fonction `NUM` donne le code de caractère comme une valeur numérique (et non pas une chaîne) pour le premier caractère de la chaîne argument.

Conversion d'un code en son caractère (CHR#). Cette fonction est l'inverse de `NUM`. Elle convertit le code de caractère (0 à 255) en son caractère équivalent. La fonction `CHR#` accepte toute expression arithmétique comme argument et, si nécessaire, soustrait ou ajoute au résultat arrondi un multiple de 256 pour obtenir un nombre dans l'intervalle 0 à 255 puis convertit ce résultat en son caractère équivalent ASCII. Par exemple, `CHR#(600)` donne `X` car `MOD(600, 256)` vaut 88 qui est le code du caractère `X`.

* ASCII: American Standard Code for Information Interchange.

Fonctions statistiques

Table des matières

Présentation	78
Déclaration de tableaux statistiques (STAT,CLSTAT)	78
Utilisation des opérations statistiques	79
Ajout de points aux tableaux (ADD)	80
Suppression de points des tableaux (DROP)	81
Somme des points (TOTAL)	82
Calcul de moyennes (MEAN)	83
Calcul d'écart types (SDEV)	83
Calcul de corrélations (CORR)	84
Régression linéaire (LR)	84
Calcul de valeurs prédites (PREDV)	85
Ajustement de valeurs à des courbes	86

Présentation

Ce chapitre vous présente:

- L'utilisation des ordres et fonctions statistiques du HP-71 dans un exemple de régression linéaire
- L'ajustement des données à une droite.
- L'utilisation des ordres et fonctions statistiques, avec les transformations nécessaires, pour des ajustements à des courbes exponentielle, logarithmique et puissance.

Déclaration de tableaux statistiques (STAT, CLSTAT)

Le HP-71 utilise un tableau spécial unidimensionnel pour stocker les données (coordonnées des points) des calculs statistiques. L'ordre STAT crée et dimensionne ce tableau et CLSTAT efface les données préalablement stockées dans un tableau.

```
STAT nom de tableau [(nombre de variables)]
```

Cet ordre dimensionne un tableau statistique unidimensionnel pour un nombre de points inférieur à 15. Le nom de tableau peut être un nom de tableau numérique quelconque. STAT permet en outre de choisir un tableau statistique préalablement dimensionné. Le *nombre de variables* est optionnel lorsque vous rappelez un tableau statistique existant. La limite basse des tableaux statistiques est toujours zéro quelle que soit l'option de base active. Tous les tableaux statistiques sont en pleine précision.

```
CLSTAT
```

Cet ordre efface (remet à zéro) tous les éléments du tableau statistique courant.

Utilisation des opérations statistiques

Exemple: Le tableau suivant liste les variations de l'indice des prix à la consommation (IPC), celles de l'indice des prix à la production (IPP) et le taux de chômage (TC), tous en pourcentage, sur une période 12 ans aux Etats-Unis.

Nous allons entrer ces données dans le HP-71 et effectuer quelques calculs statistiques.

Pour obtenir des résultats semblables à ceux de ce manuel, utilisez le format d'affichage `FIX 2`.

Données de l'exemple

Année	IPC	IPP	TC
1968	4.2	2.5	3.6
1969	5.4	3.9	3.5
1970	5.9	3.7	4.9
1971	4.3	3.3	5.9
1972	3.3	4.5	5.6
1973	6.2	13.1	4.9
1974	11.0	18.9	5.6
1975	9.1	9.2	8.5
1976	5.8	4.6	7.7
1977	6.5	6.1	7.0
1978	7.6	7.8	6.0
1979	11.5	19.3	5.8

Vous devez tout d'abord déclarer le tableau statistique dans lequel vous allez stocker ces données. Remarquez que ce tableau unidimensionnel ne contiendra pas les données entrées mais uniquement des valeurs statistiques mises à jour lors de l'entrée ou du retrait de chaque nouvelle donnée. Vous devez créer un tableau permettant de stocker les données de trois variables (IPC, IPP et TC).

Exemple:

```
STAT S(3) 
```

Crée et dimensionne un tableau statistique S pour trois variables.

```
CLSTAT 
```

Efface le contenu du tableau S.

Si un autre tableau S existe déjà, l'instruction STAT S(3) le redimensionne et efface son contenu. Nous vous recommandons de toujours effacer un tableau statistique (avec CLSTAT) avant de l'utiliser sauf si vous désirez en ré-utiliser le contenu.

Ajout de points aux tableaux (ADD)

```
ADD [valeur 1 [, valeur 2 [...[, valeur 15]...]]]
```

Cet ordre ajoute un point de donnée, avec un maximum de 15 coordonnées numérotées de 1 à 15, au jeu de données déjà représenté dans le tableau.

Exemple (suite): Sur un dessin en deux dimensions, un point est souvent défini par ses coordonnées x et y. De la même façon, le point de donnée pour 1968 est défini par ses trois coordonnées, IPC, IPP et TC. Vous devrez stocker dans le tableau les données des douze points correspondant aux années 1968 à 1979. Entrez les points comme suit:

```
ADD 4.2,2.5,3.6 
```

Données pour 1968

```
ADD 5.4,3.9,4.9 
```

Données pour 1969

```
ADD 5.9,3.7,4.9 
```

Données pour 1970

```
ADD 4.3,3.3,5.9 
```

Données pour 1971

```
ADD 3.3,4.5,5.6 
```

Données pour 1972

```
ADD 6.2,13.1,4.9 
```

Données pour 1973

```
ADD 11,18.9,5.6 
```

Données pour 1974

```
ADD 9.1,9.2,8.5 
```

Données pour 1975

```
ADD 5.8,4.8,7.7 
```

Données pour 1976

Vous réalisez que vous venez de faire une erreur; le 4.8 est en fait 4.6. Vous pouvez utiliser DROP pour corriger l'erreur.

Suppression de points des tableaux (DROP)

L'ordre DROP vous permet de supprimer facilement des points d'un tableau.

Nota: Utilisez uniquement les méthodes suivantes pour la correction des points si vous exécutez les exemples de façon que vos résultats correspondent à ceux indiqués.

Utilisation de DROP pour supprimer des points.

```
DROP [valeur 1 [, valeur 2 [...[, valeur 15]...]]]
```

Cette ordre supprime un point du tableau spécifié.

Exemple (suite): Correction de l'erreur.

```
DROP 5.8,4.8,7.7 [END LINE]
```

Supprime les données incorrectes du tableau.

```
ADD 5.8,4.6,7.7 [END LINE]
```

Entre les données correctes pour 1976.

Utilisation de la pile d'instructions pour changer des données. Vous pouvez simplifier la procédure de correction des données en utilisant la pile d'instructions pour rappeler l'instruction ADD erronée.

Exemple (suite): Voici la procédure pour corriger l'erreur en utilisant la pile d'instructions.

Entrée/Résultat

```
[9] [CMDS]
```

Active la pile d'instructions.

```
\ADD 5.8,4.8,7.7
```

La pile d'instructions affiche l'instruction la plus récente*, prête à être modifiée.

```
DROP
```

Remplace ADD par DROP.

```
\DROP 5.8,4.8,7.7
```

* Cette séquence de touches suppose que l'erreur n'a pas été corrigée.

END LINE

9 CMDS

\DROP 5.8, 4.8, 7.7

▲

\ADD 5.8, 4.8, 7.7

Appuyez dix fois sur ►.

\ADD 5.8, 4.8, 7.7

6 END LINE

Entrez maintenant le reste des données:

ADD 6.5, 6.1, 7 END LINE

ADD 7.6, 7.8, 6 END LINE

ADD 11.5, 19.3, 5.8 END LINE

Supprime la donnée incorrecte du tableau et désactive la pile d'instructions.

Ré-active la pile d'instructions.

La pile d'instructions affiche l'instruction DROP.

Affiche l'instruction la plus récente.

Positionne le curseur de remplacement sur le chiffre incorrect.

Corrige l'erreur.

Données pour 1977

Données pour 1978

Données pour 1979

Sommation des points (TOTAL)

La fonction TOTAL somme les valeurs d'une coordonnée pour tous les points. Par exemple, si chacun de vos points possède deux coordonnées (variables), x et y , la fonction TOTAL vous permet de sommer toutes les valeurs des x ou toutes celles des y .

TOTAL [*numéro de variable*]

Cette fonction donne le total des coordonnées pour la variable spécifiée dans le tableau statistique courant. Si vous omettez le *numéro de variable*, la fonction donne le total des coordonnées pour la première variable. Si vous entrez la valeur zéro comme argument, la fonction TOTAL vous donne le nombre de points du tableau.

Exemple (suite): Utilisez la fonction `TOTAL` pour afficher les totaux des trois variables (IPC, IPP et TC).

<code>TOTAL(0)</code> <code>END LINE</code>	Affiche le nombre total d'échantillons: 12.
<code>TOTAL(1)</code> <code>END LINE</code>	Affiche le total des valeurs IPC: 80.80.
<code>TOTAL(2)</code> <code>END LINE</code>	Affiche le total des valeurs IPP: 96.90.
<code>TOTAL(3)</code> <code>END LINE</code>	Affiche le total des valeurs TC: 69.00.

Calcul de moyennes (MEAN)

```
MEAN [<numéro de variable>]
```

Cette fonction donne la moyenne des valeurs pour la variable spécifiée dans le tableau statistique courant. La valeur par défaut pour le paramètre *numéro de variable* est 1.

Exemple (suite): Affichez les moyennes des trois variables.

<code>MEAN(1)</code> <code>END LINE</code>	Affiche la moyenne des valeurs IPC: 6.73.
<code>MEAN(2)</code> <code>END LINE</code>	Affiche la moyenne des valeurs IPP: 8.08.
<code>MEAN(3)</code> <code>END LINE</code>	Affiche la moyenne des valeurs TC: 5.75.

Calcul d'écart types (SDEV)

```
SDEV [<numéro de variable>]
```

Cette fonction donne l'écart type de l'échantillon des coordonnées pour la variable spécifiée du tableau statistique courant. La valeur par défaut du paramètre est 1.

Exemple (suite): Calculez l'écart type des coordonnées.

SDEV(1)

Affiche l'écart type des valeurs IPC: 2.61.

SDEV(2)

Affiche l'écart type des valeurs IPP: 5.95.

SDEV(3)

Affiche l'écart type des valeurs TC: 1.48.

Calcul de corrélations (CORR)

`CORR (numéro de variable 1, numéro de variable 2)`

Cette fonction donne la corrélation de l'échantillon de valeurs pour les deux variables spécifiées du tableau statistique courant.

Toute corrélation avec une constante est indéfinie. Si vous essayez d'exécuter `CORR(0,1)`, par exemple, le HP-71 émet un bip et affiche `Invalid Stat Op.`

Exemple (suite): Déterminez les trois corrélations entre les variables IPC, IPP et TC.

CORR(1)

Affiche la corrélation entre IPC et IPP: 0.88.

CORR(2)

Affiche la corrélation entre IPC et TC: 0.33.

CORR(3)

Affiche la corrélation entre IPP et TC: 0.14.

Régression linéaire (LR)

`LR numéro de variable 1, numéro de variable 2 [, variable [, variable]]`

Cet ordre spécifie le modèle de régression linéaire courant. Vous devez spécifier la variable dépendante par le paramètre *numéro de variable 1* et la variable indépendante par le paramètre *numéro de variable 2*. L'ordre LR calcule ensuite l'intercept et la pente pour ce modèle. Si vous spécifiez la première variable optionnelle (tout nom de variable valide est acceptable), le HP-71 stocke l'intercept dans cette variable. Si vous spécifiez la deuxième variable optionnelle, le HP-71 y stocke la pente.

Le calcul des valeurs probables (en utilisant `PREDV`) n'utilise pas ces deux variables optionnelles. Leur spécification simplifie le rappel de l'intercept et de la pente pour ce modèle. Si vous n'utilisez pas ces variables optionnelles, vous pouvez rappeler la pente et l'intercept comme suit, car `PREDV` calcule la pente et l'intercept lors de chaque exécution.

`PREDV(0)` donne l'intercept, *a*.

`PREDV(1) - PREDV(0)` donne la pente, *b*.

Vous pouvez ajuster des valeurs à une droite en utilisant la méthode des moindres carrés avec une paire de variables quelconques et l'ordre `LR`. La variable indépendante ne doit pas avoir un écart type nul.

Exemple (suite): Supposons que vous cherchiez un ajustement linéaire entre la variation de l'indice des prix à la consommation (IPC, variable dépendante en variable 1) et celle de l'indice des prix de production (IPP, variable indépendante en variable 2). En résumé, vous voulez ajuster la droite

$$\text{IPC} = a + b \cdot \text{IPP}$$

aux données et déterminer les valeurs des paramètres *a* (intercept) et *b* (pente). La variable indépendante, PPI, n'ayant pas un écart type nul (nous avons calculé précédemment que `SDEV(2) = 5.95`), vous pouvez utiliser l'ordre `LR`. Utilisez deux variables optionnelles, *A* et *B*, pour stocker les valeurs de *a* et *b*.

```
LR 1,2,A,B END LINE
```

Détermine la meilleure droite ajustée aux douze points (IPP,IPC) et stocke l'intercept dans *A* et la pente dans *B*.

```
A END LINE
```

Affiche l'intercept: 3.61.

```
B END LINE
```

Affiche la pente: 0.39.

Calcul de valeurs probables (`PREDV`)

```
PREDV (argument)
```

Cette fonction donne la valeur probable de la variable dépendante à partir du modèle de régression linéaire courant et de la valeur de la variable indépendante spécifiée comme argument. Vous devez exécuter `LR` pour spécifier les variables dépendante et indépendante avant d'exécuter `PREDV`.

Exemple (suite): Prédisez maintenant les valeurs de IPC pour des valeurs de IPP égales à 4, 5, 6 et 7.

PREDV (4)

Affiche la valeur probable de IPC pour
IPP = 4: 5.16.

PREDV (5)

Affiche la valeur probable de IPC pour IPP = 5:
5.54.

PREDV (5)

Affiche la valeur probable de IPC pour IPP = 6:
5.93.

PREDV (5)

Affiche la valeur probable de IPC pour IPP = 7:
6.32.

Ajustement de valeurs à des courbes

En utilisant les transformations nécessaires, le HP-71 vous permet d'ajuster des données à des courbes exponentielle, logarithmique et puissance à l'aide du format de régression linéaire standard:

$$y = a + bx.$$

Le tableau suivant résume ces transformations.

Transformation en format de régression linéaire

Nom de la courbe	Equation non transformée	Pour y, Utilisez:	Pour a, Utilisez:	Pour x, Utilisez:	Equation transformée
Linéaire	$y = a + b \times x$	y	a	x	$y = a + b \times x$
Exponentielle	$y = a \times e^{(b \times x)}$ ($a > 0$)	ln(y)	ln(a)	x	$\ln(y) = \ln(a) + b \times x$
Logarithme	$y = a + b \times \ln(x)$	y	a	ln(x)	$y = a + b \times \ln(x)$
Puissance	$y = a \times x^b$ ($a > 0$)	ln(y)	ln(a)	ln(x)	$\ln(y) = \ln(a) + b \times \ln(x)$

Exemple: Supposons que vous deviez analyser les valeurs x et y, obtenues durant une expérience. Vous voulez déterminer laquelle des quatre courbes—linéaire, exponentielle, logarithmique et puissance—correspond le mieux aux données.

Données pour exemple de transformation

x	.1	1.3	4.7	9.0	17.9	24.4
y	16.69	13.51	7.498	3.662	.7170	.3271

Pour faciliter l'entrée des données dans un tableau statistique, vous pouvez les ré-arranger comme dans le tableau suivant.

Données ré-arrangées pour exemple de transformation

	Numéro de variable			
	1	2	3	4
	Variable			
	x	ln(x)	y	ln(y)
Observation	Valeur de la variable			
1	.1	-2.303	16.69	2.815
2	1.3	.2624	13.51	2.603
3	4.7	1.548	7.498	2.015
4	9.0	2.197	3.662	1.298
5	17.9	2.885	.7170	-.3327
6	24.4	3.195	.3271	-1.117

Créez maintenant le tableau statistique et entrez vos données:

```
STAT E(4) END LINE
```

Dimensionne un nouveau tableau (cette instruction choisit et redimensionne le tableau E s'il existe déjà).

```
CLSTAT END LINE
```

Cette instruction efface le tableau E s'il contenait des valeurs. Elle n'est pas nécessaire dans de nombreux cas. C'est néanmoins une bonne habitude à prendre de façon à éviter de mélanger d'anciennes données avec celles que vous désirez entrer.

```
ADD .1,-2.303,16.69,2.815
```

```
END LINE
```

Première observation.

```
ADD 1.3,.2624,13.51,2.603
```

```
END LINE
```

Deuxième observation.

```
ADD 4.7,1.548,7.498,2.015
```

```
END LINE
```

Troisième observation.

```
ADD 9,2.197,3.662,1.298 END LINE
```

Quatrième observation.

```
ADD 17.9,2.885,.717,-.3327
```

```
END LINE
```

Cinquième observation.

```
ADD 24.4,3.195,.3271,-1.117
```

```
END LINE
```

Sixième observation.

Déterminez les corrélations nécessaires pour savoir si un ou plusieurs des modèles doit être éliminé à ce stade. Exécutez les fonctions de corrélations ci-dessous et étudiez les résultats. Les arguments des fonctions de corrélation sont les numéros de variable du tableau précédent. Le tableau de transformation de la page 86 montre quelles variables utiliser pour chaque type de courbe.

Par exemple, la transformation permettant de chercher l'ajustement à une courbe logarithmique utilise $\ln(x)$ pour x et y pour y . Le tableau suivant (page 87) indique que $\ln(y)$ est la variable 2 et y la variable 3. La fonction de corrélation appropriée dans cet exemple pour une courbe logarithmique sous forme de régression linéaire est `CORR(2,3)`.

Pour les courbes exponentielle, logarithmique et puissance, vous vérifiez si les données transformées s'ajustent bien à une droite. Si une ou plusieurs de ces courbes présentent une bonne corrélation, vous pouvez utiliser `PREDV` pour calculer des valeurs probables de la variable dépendante (y ou $\ln(y)$) en fonction de valeurs de la variable indépendante (x ou $\ln(x)$). La dernière étape consiste à transformer les valeurs $\ln(y)$ en valeurs y d'origine à l'aide de la fonction `EXP`.

Corrélations résultant de l'exemple de transformation

Type de courbe	Fonction de corrélation	Résultat
Linéaire	<code>CORR(1,3)</code>	-0.90
Exponentielle	<code>CORR(1,4)</code>	-1.00
Logarithmique	<code>CORR(2,3)</code>	-0.96
Puissance	<code>CORR(2,4)</code>	-0.84

Aucune des corrélations n'est vraiment mauvaise. Remarquez que toutes les courbes transformées (droites) ont une pente négative, comme indiquée par leurs corrélations négatives.

Vous décidez de choisir la courbe ayant la corrélation la meilleure, l'exponentielle. Vous devez d'abord utiliser l'ordre `LR` pour spécifier le modèle de régression correspondant aux données transformées de la courbe exponentielle. Une fois ce modèle défini, vous pouvez utiliser la fonction `PREDV` pour calculer des valeurs probables $\ln(y)$ ou pour vérifier les valeurs transformées des données initiales.

`LR 4,1,A,B` `END LINE`

Spécifie une régression linéaire avec $\ln(y)$ (variable 4) comme variable dépendante et x (variable 1) comme variable indépendante. Le HP-71 stocke l'intercept dans `A` et la pente dans `B`.

`A` `END LINE`

Affiche l'intercept 2.80.

`B` `END LINE`

Affiche la pente -0.17.

Vous êtes maintenant prêt à calculer des valeurs probables. Calculez des valeurs probables de $\ln(y)$ pour les valeurs suivantes de x : -10 , -5 , 0 , 20 et 30 . Pour $x = 0$, la valeur probable $\ln(y)$ doit être égale à l'intercept A . Enfin, vous pouvez retransformer les valeurs $\ln(y)$ en valeurs y . Pour vérification, vous pouvez utiliser des valeurs de x égales à des points de données de façon à savoir si les valeurs probables de y s'approchent bien des coordonnées y correspondantes.

Entrée/Résultat

PREDV(-10) [ENDLINE]

Affiche 4.45 la valeur probable de $\ln(y)$ pour $x = -10$.

EXP(RES)

Affiche $e^{4.45}$, où 4.45 est la valeur probable $\ln(y)$ obtenue.

85.88

Valeur probable de y dans l'équation $y = a \times e^{(b \times x)}$ pour une valeur de x égale à -10 , lorsque les paramètres a et b ont les valeurs EXP(A) et B (les valeurs A et B proviennent du calcul par l'ordre LR).

Calculez les autres valeurs probables de la même façon. Vous devez obtenir des résultats identiques à ceux du tableau suivant.

Valeurs probables pour l'exemple de transformation

x	-5	0	4.7	9	20	24.4	30
ln(y) probable	3.63	2.80	2.02	1.31	-0.51	-1.24	-2.17
y probable	37.53	16.40	7.53	3.70	0.60	0.29	0.11

La valeur $\ln(y)$ pour $x = 0$ est 2.80, valeur égale à l'intercept A . De plus, les valeurs probables de y pour $x = 4.7$, 9 et 24.4 sont suffisamment proches des coordonnées y originales indiquées dans le tableau de la page 86.

Horloge et calendrier

Table des matières

Présentation	90
Calendrier du HP-71 (SETDATE, DATE, DATE\$)	90
Horloge du HP-71	91
Mise à l'heure (SETTIME, ADJABS)	92
Lecture de l'heure (TIME, TIME\$)	94
Ajustement de la vitesse (SETTIME, ADJUST, AF, EXACT, RESET CLOCK)	94

Présentation

Ce chapitre vous présente:

- La mise à jour et la lecture du calendrier.
- La mise à l'heure et la lecture de l'horloge.
- L'obtention de la meilleure précision.

Calendrier du HP-71 (SETDATE, DATE, DATE\$)

Le HP-71 accepte les dates du 1^{er} janvier 0000 au 31 décembre 9999. Néanmoins les dates antérieures au 15 octobre 1582, ou au 1^{er} janvier 1752 pour les pays de langue anglaise, ne correspondent pas directement à notre calendrier grégorien.

SETDATE *date sous forme numérique*
 SETDATE *date sous forme alphanumérique*

Cet ordre définit la date courante pour le HP-71 égale à un entier ou à une chaîne. Les dates numériques doivent être entrées sous la forme AADDD ou AAAADDD, où AA ou AAAA est l'année et DDD est le numéro du jour dans l'année. Le numéro du jour est compris entre 001 et 365 (366 pour les années bisextiles), bornes comprises et doit toujours comporter trois chiffres. Les dates alphanumériques doivent être de la forme "AA/MM/DD" ou "AAAA/MM/DD" et doivent toujours constituer une chaîne de huit ou dix chiffres.

Les résultats des fonctions `DATE` et `DATE#` ne sont pas affectés par le format d'entrée. Ces deux méthodes vous permettent simplement de choisir le format qui vous semble le plus pratique.

Exemples: Les deux instructions suivantes définissent la date du 7 mars 1985. Remarquez la présence des zéros d'en-tête dans le numéro de jour à trois chiffres du format numérique et dans les positions de mois et de jour du format alphanumérique.

```
SETDATE 85066
SETDATE "85/03/07"
```

DATE

Cette fonction donne la date sous la forme d'un nombre tel que `AADDD`, où `AA` est égal aux deux derniers chiffres de l'année et `DDD` est égal au numéro du jour dans l'année.

Exemple: Supposons la date du 7 mars 1985.

```
DATE 
```

Donne `85066`. En 1985, le 7 mars est le 66^{me} jour de l'année.

DATE#

Cette fonction donne la date sous la forme `"AA/MM/DD"`, où `AA` est égal aux deux derniers chiffres de l'année, `MM` est le numéro du mois, et `DD` est le jour du mois.

```
DATE# 
```

Donne `85/03/07`. Même date (7 mars 1985) sous forme d'une chaîne alphanumérique.

Horloge du HP-71

Le HP-71 possède un jeu d'ordres et de fonctions très souples pour la mise à l'heure et l'ajustement de l'horloge à quartz ainsi que pour la régulation de la vitesse (avance ou retard). Une fois les mots-clés présentés, vous pourrez aisément changer la mise à l'heure pour tenir compte des changements de fuseaux horaires par exemple et pour maintenir la précision de l'horloge à environ quelques secondes par semaines ou même par mois.

L'ordre `ADJABS` ajuste l'horloge sans introduire de facteur de correction de la vitesse. Deux autres ordres, `SETTIME` et `ADJUST`, mettent l'horloge à l'heure ou l'ajustent et introduisent un facteur de correction de la vitesse. Ces facteurs s'accumulent entre deux exécutions de l'ordre `EXACT` et sont appliqués à l'heure lors de la deuxième exécution de `EXACT`. Enfin, l'ordre `AF` introduit et applique les facteurs de correction. Néanmoins, lors de l'exécution de `AF`, l'heure de l'horloge reste inchangée.

Le tableau suivant résume ces ordres.

Mots-clés pour ajuster et corriger l'horloge

Mise à l'heure	Mise à l'heure et accumulation d'erreur de vitesse	Correction de vitesse basée sur l'erreur accumulée	Correction de vitesse basée sur le paramètre
ADJABS	SETTIME ADJUST	EXACT	AF

Mise à l'heure (SETTIME, ADJABS)

Lorsque vous utilisez l'horloge du HP-71 pour la première fois ou après une ré-initialisation, vous devez utiliser l'ordre SETTIME pour mettre l'horloge à l'heure. *N'utilisez plus SETTIME ensuite sauf si vous voulez accumuler un facteur de correction de vitesse.*

ADJABS met l'horloge à l'heure *sans* accumuler le facteur de correction de vitesse. Le paramètre de cet ordre est un incrément de temps, tel que 15 secondes ou "-1:00:00" (reculer l'heure d'une heure). L'ordre ADJABS permet entre-autre d'ajuster l'heure lorsque vous changez de fuseaux horaires.

Mise à l'heure initiale (SETTIME)

SETTIME	<i>secondes depuis minuit</i>
SETTIME	<i>expression alphanumérique de temps</i>

Vous pouvez mettre l'horloge du HP-71 à l'heure en utilisant une expression numérique (*secondes depuis minuit*) de valeur entre 0 et 86399 ou une *expression alphanumérique* de la forme "HH:MM:SS", où HH représente les heures en format sur 24 heures, MM les minutes et SS les secondes. Vous devez spécifier les zéros d'en-tête de façon que la chaîne ait toujours huit caractères.

Le chapitre 1, page 17, liste les opérations permettant de mettre l'horloge du HP-71 à l'heure.

Après une ré-initialisation de la mémoire, si vous exécutez SETTIME avant EXACT, le HP-71 n'accumule pas l'erreur pour la correction de la vitesse, l'ajustement ne sert qu'à mettre l'horloge à l'heure. Une fois l'ordre EXACT exécuté, le HP-71 continue à utiliser l'ajustement total pour mettre l'horloge à l'heure. Toute partie de l'ajustement autre que les heures pleines ou les demi-heures est accumulée comme facteur de correction de la vitesse*. Ce facteur sert à changer la vitesse de l'horloge lors de l'exécution suivante de EXACT.

* Par conséquent, si vous désirez ajuster l'heure à cause d'un changement de fuseau horaire, vous pouvez utiliser SETTIME. Si vous changez l'heure d'une heure juste, le HP-71 n'accumule pas d'erreur. Mais si vous ajoutez, par exemple, une heure et trois minutes, le HP-71 accumule les trois minutes comme facteur de correction.

En résumé:

- Après une ré-initialisation de la mémoire, exécutez `SETTIME` avant `EXACT` pour mettre l'horloge à l'heure.
- Après l'exécution de `EXACT`, utilisez `SETTIME` pour simultanément:
 1. Remettre l'horloge à l'heure car elle avance ou retarde.
 2. Accumuler un facteur de correction de vitesse qui sera converti en modification de vitesse lors de l'exécution suivante de `EXACT`.

Exemples: On suppose les instructions suivantes exécutées après une ré-initialisation de la mémoire et avant l'ordre `EXACT`.

```
SETTIME 8*3600+15*60
```

Met l'horloge à l'heure à 8 heures 15 (nombre de secondes pour 8 heures et pour 15 minutes).

```
SETTIME "08:15:00"
```

Met aussi l'horloge à l'heure à 8 heures 15.

```
SETTIME "18:08:05"
```

Met l'horloge à l'heure à 18 heures, 8 minutes et 5 secondes.

Ajustement de l'heure (ADJABS)

```
ADJABS ajustement en secondes
```

```
ADLABS chaîne d'ajustement
```

Le paramètre *ajustement en secondes* peut être une expression numérique quelconque. Le HP-71 accepte les changements positifs et négatifs. La *chaîne d'ajustement* doit être de la forme "HH:MM:SS" ou "-HH:MM:SS". Vous devez entrer les zéros d'en-tête de façon que la chaîne comporte huit ou neuf caractères. L'ensemble de l'ajustement est considéré comme un changement de fuseau horaire; rien n'est utilisé comme facteur de correction.

Exemple: Supposons que la montre que vous avez utilisée pour mettre l'horloge du HP-71 à l'heure avait 43 secondes de retard. Vous pouvez exécuter cette instruction pour ajouter 43 secondes à l'horloge:

```
ADJABS 43
```

Supposons maintenant que vous partiez en voyage sur la côte ouest des Etats-Unis, vous pouvez utiliser l'instruction suivante pour mettre l'horloge à l'heure:

```
ADJABS -32400
```

Retarde l'heure de 32400 secondes (9 heures).

La suite de votre voyage vous emmène à Salt-Lake City. Vous devez ajouter une heure à l'horloge:

```
ADJABS "01:00:00"
```

Avance l'horloge d'une heure.

Lecture de l'heure (TIME, TIME#)

TIME

Cette fonction donne l'heure de l'horloge sous forme d'un nombre de secondes depuis minuit. Vous pouvez utiliser cette valeur dans des expressions numériques comme toute autre fonction numérique.

Exemple: Supposons que vous vouliez écrire un programme permettant de chronométrer les durées d'une série d'expériences commençant toutes au même moment. L'instruction suivante affecte l'heure de départ des expériences à la variable T.

```
230 T=TIME
```

La ligne 780 de votre programme (ci-dessous) est exécutée lorsque la première expérience se termine. Le HP-71 affecte alors la durée de l'expérience à la variable E1. Supposons maintenant que vos expériences commencent à 9 heures (32400 secondes après minuit) et que la première expérience se termine à 10 heures 30 (37800 secondes après minuit). La ligne 780 du programme affecte dans ce cas la valeur 5400 à la variable E1.

```
780 E1=TIME-T
```

TIME#

Cette fonction donne l'heure sous la forme d'une chaîne alphanumérique telle que "HH:MM:SS", dans le format sur 24 heures.

Ajustement de la vitesse (SETTIME, ADJUST, AF, EXACT, RESET CLOCK)

La suite de ce chapitre présente l'utilisation la plus efficace de ces mots-clés.

Correction de vitesse. Ces mots-clés sont normalement utilisés dans les situations suivantes:

- **SETTIME.** Exécutez SETTIME lorsque vous désirez utiliser la partie principale pour changer l'heure et utiliser les valeurs autres que des heures pleines ou des demi-heures dans le facteur de correction de vitesse. Après une ré-initialisation, vous utilisez cet ordre pour mettre l'horloge à l'heure.
- **ADJUST.** Exécutez ADJUST si vous voulez ajouter ou soustraire de l'heure de l'horloge la partie principale du paramètre et utiliser le reste, partie autre que les heures pleines ou les demi-heures, dans le facteur de correction de la vitesse.
- **AF.** Exécutez AF si vous voulez modifier uniquement la vitesse de l'horloge sans attendre l'exécution suivante de EXACT.

La syntaxe de `SETTIME` est présentée en page 92.

```
ADJUST secondes
ADJUST chaîne d'ajustement
```

Cet ordre vous permet d'ajuster l'heure de l'horloge pour les changements de fuseaux horaires, les passages d'heure d'été à heure d'hiver, etc. et d'accumuler en même temps une durée (inférieure à ± 15 minutes) pour une correction de vitesse ultérieure. Le HP-71 effectue cette correction de vitesse lors de l'exécution suivante de `EXACT`. Le paramètre de cet ordre peut être une expression numérique de valeur comprise entre -360.000 et 360.000 secondes (100 heures) ou une expression alphanumérique de la forme "HH:MM:SS" ou "-HH:MM:SS" (vous devez spécifier les zéros d'en-tête de façon que la chaîne comporte huit ou neuf caractères).

Exemple: Vous vous préparez à retourner sur la côte est des Etats-Unis. Vous savez que l'horloge de votre HP-71 a perdu une minute depuis sa dernière remise à l'heure quatre mois auparavant. Vous désirez maintenant:

1. Avancer l'horloge d'une heure.
2. Accumuler un facteur de correction de vitesse pour compenser le retard que l'horloge prend.
3. Relancer l'horloge pour rattraper la minute perdue.

Vous pouvez exécuter ces trois opérations en même temps:

```
ADJUST "01:01:00" END LINE
```

```
AF (nouveau facteur d'ajustement)
```

Le *facteur d'ajustement* change la vitesse de l'horloge. Ce paramètre correspond au nombre de secondes écoulées avant que l'horloge n'ajoute (positif) ou ne soustrait (négatif) une seconde à l'heure. L'ordre `AF` affiche en outre la valeur courante du facteur d'ajustement. Si vous exécutez `AF` avec un paramètre, la nouvelle valeur remplace l'ancienne. L'ajustement est immédiat contrairement aux ordres `ADJUST` et `SETTIME` et ne nécessite pas l'exécution de `EXACT`.

Exemple: Le nouveau facteur d'ajustement change la vitesse de l'horloge de la façon suivante (pour cet exemple, on suppose le facteur courant égal à 24000).

```
AF (-28800) END LINE Affiche le facteur d'ajustement courant
(24000) et en définit un nouveau (-28800).
```

Après environ 56 secondes (28800/512), l'horloge retire une fraction de seconde (1/512) de l'heure.

Correction de l'erreur de vitesse accumulée.

```
EXACT
```

Cet ordre permet d'optimiser la précision de la vitesse de l'horloge. La première exécution de EXACT après une ré-initialisation de la mémoire définit le début d'une période d'ajustement. Chaque exécution ultérieure de EXACT définit la fin de la période d'ajustement courante et le début d'une nouvelle. Lors de l'exécution de EXACT, le HP-71 utilise toutes les corrections de vitesse accumulées par l'exécution de SETTIME et ADJUST pendant la période d'ajustement courante (peut-être plusieurs semaines ou plusieurs mois). Souvenez-vous que les ordres SETTIME et EXACT ne corrigent pas la vitesse de l'horloge, ils accumulent simplement les facteurs de correction. Le HP-71 utilise le nouveau facteur de correction comme avec l'ordre AF.

L'ordre EXACT servant à optimiser la précision de l'horloge, ne l'utilisez que lorsque vous êtes absolument sûr de l'heure.

Annulation du facteur d'ajustement de la vitesse.

```
RESET CLOCK
```

Cet ordre efface le facteur d'ajustement et annule les effets de l'ordre EXACT.

Opérations sur les fichiers

Table des matières

Présentation	98
Fichier courant	100
Fichier de travail: <code>workfile</code>	100
Introduction à l'utilisation des fichiers	101
Structure de la mémoire du HP-71	103
Types de mémoire: RAM et ROM	103
Mémoires vives principale et indépendantes	104
Déclaration d'une mémoire vive indépendante (<code>FREE PORT</code>)	105
Rappel de la mémoire vive indépendante (<code>CLAIM PORT</code>)	107
Obtention d'informations sur la mémoire (<code>MEM, SHOW PORT</code>)	107
Noms de fichier	109
Caractéristiques des noms de fichiers	109
Fichiers par défaut	110
Mots réservés	110
Noms d'unités	110
Caractéristiques des noms d'unités	111
Unités par défaut	111
Ordre de recherche des fichiers	112
Copie de fichier (<code>COPY</code>)	112
Changement de nom de fichier (<code>RENAME</code>)	115
Suppression de fichier (<code>PURGE</code>)	115
Protection de fichier	116
Protection du contenu (<code>SECURE, UNSECURE</code>)	116
Contrôle de l'accès (<code>PRIVATE</code>)	116
Utilisation de <code>SECURE</code> et <code>PRIVATE</code>	117
Catalogues de fichiers	117

Présentation

Le HP-71 conserve les informations en mémoire sous forme de fichiers. Il peut contenir plusieurs fichiers en même temps, chacun d'entre-eux étant identifié par un nom unique. Ce chapitre vous présente comment le HP-71 organise les fichiers. La création et l'ajout d'informations dans les fichiers sont traités séparément dans d'autres chapitres.

Ce chapitre présente:

- La notion de *fichier courant*.
- Le *fichier de travail* `workfile`.
- La structure de la mémoire du HP-71.
- La copie de fichiers.
- Le changement de noms de fichiers.
- La suppression de fichiers en mémoire.
- La protection des fichiers.
- L'obtention d'informations sur des fichiers.

Si vous désirez simplement exécuter des programmes pré-enregistrés dans des modules, ce chapitre ne vous est pas nécessaire. Par contre si vous voulez exécuter des programmes stockés sur cartes magnétiques ou si vous voulez utiliser des périphériques HP-IL, lisez attentivement ce chapitre.

Si vous voulez créer des fichiers, y stocker des informations ou utiliser les types de fichiers suivants, consultez les chapitres indiqués:

- **Fichiers BASIC.**

Ce type de fichier contient un programme BASIC. Voir chapitre 8, «*Rédaction et exécution de programmes*».

- **Fichiers BIN et LEX.**

Les fichiers de ces deux types contiennent des programmes écrits en langage machine. Vous pouvez exécuter un fichier BIN comme sous-programme. Un fichier LEX ajoute des mots-clés au langage du HP-71. Voir chapitre 8, «*Rédaction et exécution de programmes*».

- **Fichiers DATA.**

Les fichiers DATA contiennent des données numériques et alphanumériques et sont utilisés par les programmes pour le stockage des données. Voir chapitre 14, «*Stockage et rappel de données*».

- **Fichiers TEXT.**

Ce type de fichiers particulier sert au transfert de données entre le HP-71 et d'autres ordinateurs. Vous pouvez convertir un fichier BASIC en un fichier TEXT de façon à le transférer à un autre ordinateur ou convertir en fichier BASIC un fichier TEXT provenant d'un autre ordinateur. Voir chapitre 14, «*Stockage et rappel de données*».

- **Fichiers KEY.**

Les fichiers KEY contiennent les redéfinitions de touches que vous créez. Plusieurs fichiers de touches peuvent être présents en mémoire; néanmoins, seul un d'entre-eux peut être actif à un moment donné. Voir chapitre 7, «*Personnalisation du HP-71*».

- **Fichiers SDATA.**

Les fichiers SDATA sont des fichiers de données que vous pouvez envoyer ou recevoir d'un ordinateur HP-41. Voir chapitre 14, «*Stockage et rappel de données*».

Fichier courant

La mémoire du HP-71 peut contenir plusieurs fichiers en même temps. Un seul d'entre-eux est le *fichier courant* à un moment donné. Vous ne pouvez modifier un fichier que lorsqu'il est défini comme fichier courant. Ce fichier est de plus celui sur lequel le HP-71 effectue de nombreuses opérations par défaut.

Les fonctions et ordres suivants changent le fichier courant:

- EDIT.
- RUN.
- CHAIN.
- FREE PORT.
- CLAIM PORT.
- PURGE (seulement lorsque vous supprimez le fichier courant).
- Insertion ou retrait d'un module RAM ou ROM d'un logement.
- TRANSFORM (seulement lorsque vous transformez le fichier courant en un fichier non-BASIC).

Fichier de travail: `workfile`

Le HP-71 contient un fichier programme appelé `workfile`, qui sert de fichier de brouillon. Le *fichier de travail* devient fichier courant lorsque vous:

- Installez la batterie pour le première fois.
- Ré-initialisez le HP-71 (INIT: 3).
- Supprimez le fichier courant.
- Exécutez EDIT sans spécifier de nom de fichier.
- Transformez le fichier courant en un fichier non-BASIC.
- Insérez ou retirez un module RAM ou ROM d'un logement.
- Exécutez FREE PORT ou CLAIM PORT.

Pour plus d'informations sur l'utilisation du fichier de travail, consultez le chapitre 8 «Rédaction et exécution de programmes».

Introduction à l'utilisation des fichiers

Chaque fichier en mémoire est identifié par un nom unique composé de huit lettres et chiffres (le premier caractère doit être une lettre). Vous pouvez donc identifier vos fichiers par des mots ou des mnémoniques représentatives de leurs contenus ou de leurs fonctions.

Lors de l'utilisation de l'ordinateur, vous aurez sans doute à créer et copier des fichiers, à changer leurs noms et à en supprimer de la mémoire. Nous vous donnons ci-dessous quelques exemples de ces opérations. Les détails de fonctionnement des ordres utilisés seront présentés plus loin dans ce chapitre ainsi que dans le *Manuel de référence* du HP-71.

Si vous rédigez un programme dans le fichier de travail et voulez lui donner un nom, vous utilisez l'ordre NAME:

Entrée/Résultat

EDIT

Désigne `workfile` comme le fichier courant.

workfile	BASIC	Ø
----------	-------	---

Affiche le nom du fichier (`workfile`) et le type du fichier (`BASIC`).

1Ø GOTO 1Ø

Entre une ligne dans le fichier.

NAME TEST

Nomme le fichier `TEST`.

EDIT

Crée un nouveau fichier de travail.

workfile	BASIC	Ø
----------	-------	---

Plutôt que d'utiliser le fichier de travail, vous pouvez simplement créer un fichier avec un nom et y entrer directement les lignes de programme:

Entrée/Résultat

EDIT TEST1

Crée un nouveau fichier nommé `TEST1`. Ce fichier devient le fichier courant.

TEST1	BASIC	Ø
-------	-------	---

Indique que vous avez créé `TEST1` et que ce fichier est de type `BASIC`.

Pour éviter de confondre un fichier avec d'autres fichiers ayant des noms similaires, vous pouvez changer le nom d'un fichier en utilisant l'ordre `RENAME`:

Entrée/Résultat

RENAME TEST1 TO VOLTAGE1

Remplace le nom `TEST1` par `VOLTAGE1`.

Si vous voulez effectuer des modifications dans un fichier, il est souvent prudent de conserver une copie de l'original. Vous pouvez copier un fichier sous un autre nom en utilisant l'ordre COPY:

```
COPY VOLTAGE1 TO VOLTEST1
END LINE
```

Crée une copie du fichier VOLTAGE1 et la nomme VOLTEST1. Les deux fichiers sont maintenant présents en mémoire.

Après avoir créé plusieurs fichiers, vous pouvez afficher un catalogue des fichiers en mémoire en utilisant l'ordre CAT ALL:

Entrée/Résultat

```
CAT ALL ENDLINE
```

Demande à l'ordinateur d'afficher une liste des fichiers en mémoire.

```
NAME      S TYPE      LEN
```

```
 N      DATE      TIME PORT
```

Affiche l'en-tête du catalogue.

```
TEST          BASIC      12
```

Indique que le fichier le plus ancien est TEST.

▼

Affiche le nom du fichier suivant.

```
workfile     BASIC      0
```

Le fichier suivant est workfile.

▼

```
VOLTAGE1     BASIC      0
```

Affiche l'étiquette du fichier VOLTAGE1.

▼

```
VOLTEST1     BASIC      0
```

Affiche l'étiquette de VOLTEST1. La pression suivante de ▼ affiche le même nom de fichier car c'est le dernier du catalogue.

```
ATTN
```

Rappelle le symbole BASIC à l'affichage.

Lorsque vous ne voulez plus utiliser un fichier, parce que vous disposez d'une version plus récente par exemple, vous pouvez le supprimer de la mémoire avec l'ordre `PURGE` de façon à libérer la mémoire pour d'autres utilisations:

```
PURGE VOLTEST1
```

Supprime `VOLTEST1` de la mémoire.

Cette démonstration rapide des opérations de fichiers vous montre comment créer des fichiers, les nommer, en changer le nom, en afficher l'étiquette et les supprimer. Le HP-71 vous offre des possibilités bien plus grandes mais avant de les étudier, nous devons d'abord comprendre la structure de la mémoire du HP-71.

Structure de la mémoire du HP-71

Le HP-71 vous offre une très grande souplesse quant à l'emplacement de stockage des fichiers dans la mémoire de l'ordinateur. Vous pouvez découper la mémoire du HP-71 en petites unités, appelés *logements* (voir ci-après), dans lesquels vous pouvez stocker des programmes et des données. Le stockage d'informations dans un logement particulier permet à l'ordinateur de les rappeler rapidement car il ne doit les chercher que dans le logement spécifié et non pas dans toute la mémoire. Cette caractéristique augmente la vitesse d'utilisation des fichiers.

Types de mémoire: RAM et ROM

Le HP-71 comporte deux types de mémoire:

- Mémoire morte (ROM). Vous ne pouvez pas en modifier le contenu.
- Mémoire vive (RAM). Vous pouvez stocker des informations dans cette mémoire.

Mémoire morte (ROM). Le HP-71 contient 64K octets de ROM*. La ROM contient le système d'exploitation et toutes les fonctions de l'ordinateur. Vous ne pouvez pas stocker d'informations dans cette zone mais vous pouvez augmenter les capacités de l'ordinateur en ajoutant des ROM enfichables dans les logements d'entrée/sortie. Vous pouvez aussi exécuter des programmes stockés en ROM.

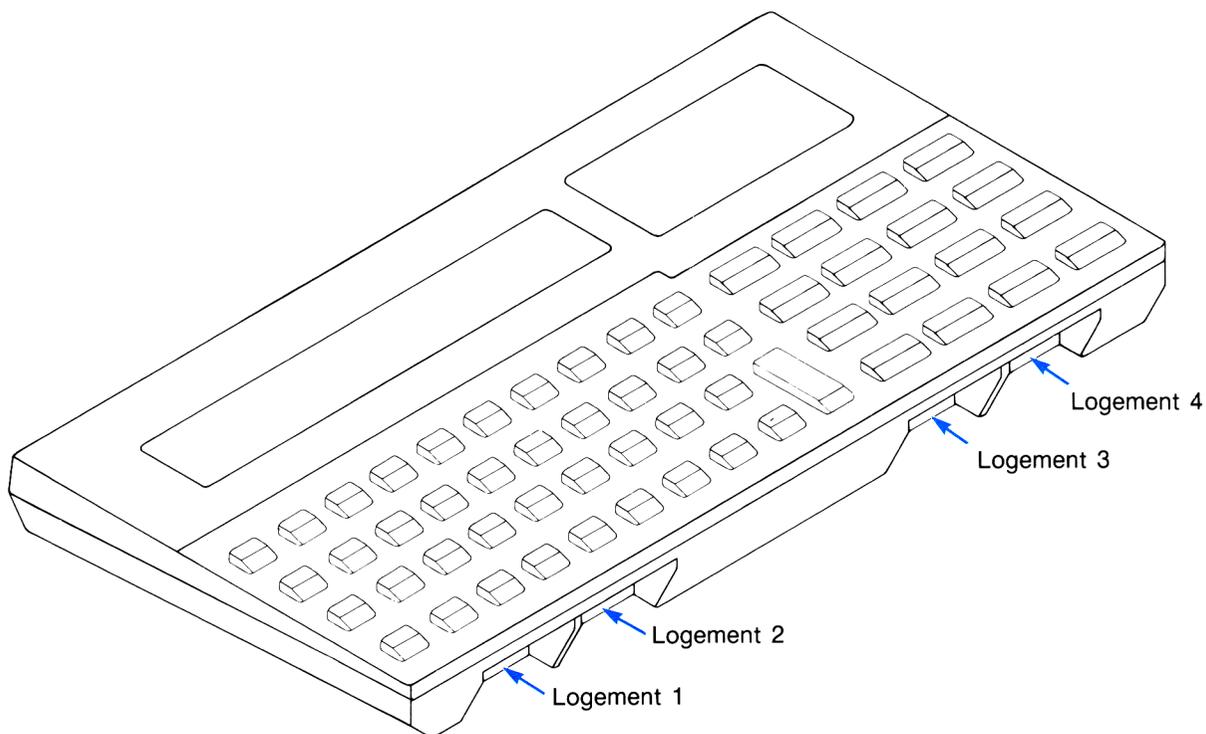
Mémoire vive (RAM). Le HP-71 contient 17.5K octets de mémoire vive, entièrement disponible pour l'utilisateur (l'ordinateur utilise néanmoins environ 1K octet de mémoire vive pour ses opérations). Vous pouvez ajouter quatre modules de mémoire vive pour accroître la capacité de stockage du HP-71.

Le présent chapitre traite principalement de la manipulation des fichiers en mémoire vive. Il décrit en outre comment copier des fichiers d'un module enfichable.

* Un Kilo-octet égale 1024 (2^{10}) octets.

Mémoires vives principale et indépendantes

Le HP-71 comporte quatre logements externes en plus des logements de l'interface HP-IL et du lecteur de carte. Ces logements sont numérotés de 1 à 4 de gauche à droite. Vous pouvez connecter des modules d'applications (mémoire morte) ou des modules mémoire (mémoire vive) dans un ordre quelconque dans chacun de ces logements.



Le HP-71 comporte un logement supplémentaire, numéroté 0, interne. Vous ne pouvez pas ajouter de module dans ce logement car il contient déjà 16K octets de mémoire vive qui peuvent être séparés du reste de la mémoire vive interne.

La mémoire vive du HP-71 est de deux types:

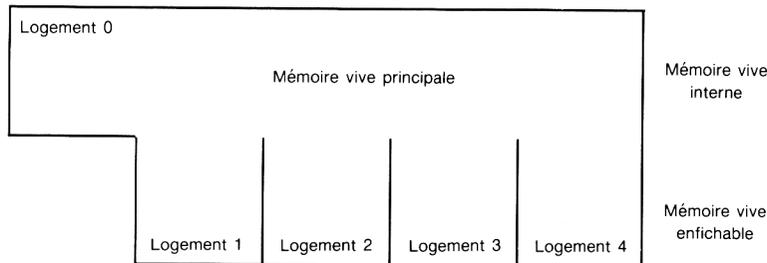
- Mémoire vive principale (RAM).
- Mémoire vive indépendante (IRAM).

Mémoire vive principale. Le HP-71 est équipé à l'origine d'une certaine quantité de mémoire vive (voir ci-dessus) dont une partie se trouve dans le logement 0. Cette mémoire vive et, par défaut, toute celle connectée aux logements extérieurs est appelée *mémoire vive principale*. Le HP-71 utilise la mémoire vive principale pour ses opérations, le stockage des fichiers et des variables.

Mémoire vive indépendante. Cette mémoire est séparée de la mémoire vive principale et n'est pas utilisée directement par l'ordinateur mais sert uniquement de support de stockage pour vos fichiers. La *mémoire vive indépendante* est utile pour:

- Protéger les fichiers contre une ré-initialisation de la mémoire INIT: 3.
- Permettre à l'ordinateur de localiser rapidement des fichiers, car les recherches se trouvent limitées à une petite partie de la mémoire.
- Vous permettre de retirer un module mémoire d'un logement sans détruire le contenu du reste de la mémoire vive.

Si vous placez des modules mémoire dans les quatre logements (et en supposant qu'ils ne sont pas déclarés comme mémoire vive indépendante), la mémoire principale se compose de la mémoire vive interne plus des modules enfichables tel qu'illustré ci-après:



Les mémoires vives principale et indépendantes et les modules de mémoire morte sont appelés des *unités de mémoire*. Par exemple, lorsque la mémoire vive principale se compose de toute la mémoire vive interne et de tous les modules de mémoire vive (comme dans l'illustration ci-dessus), il y a une seule unité de mémoire: la mémoire vive principale. Toute portion de celle-ci déclarée comme mémoire vive indépendante devient une unité de mémoire séparée. La mémoire morte de l'ordinateur est toujours une unité de mémoire séparée.

Déclaration d'une mémoire vive indépendante (FREE PORT)

Vous pouvez déclarer la mémoire vive d'un logement comme mémoire vive indépendante en exécutant l'ordre FREE PORT.

```
FREE PORT (numéro de logement)
```

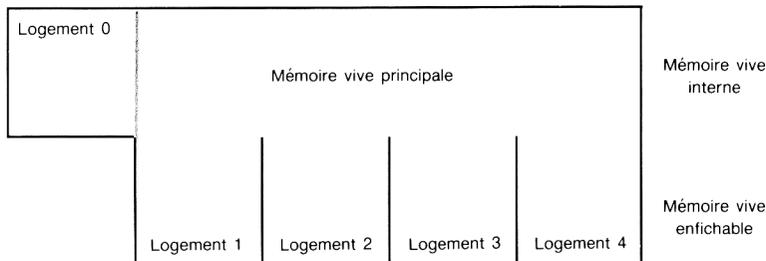
Le numéro de logement peut être un entier entre 0 et 5 (bornes comprises). Le paramètre 5 représente le logement du lecteur de cartes.

Exemple: Déclarez le module du logement 0 comme mémoire vive indépendante:

```
FREE PORT (0)
```

La mémoire vive du logement 0 devient un support de mémoire vive indépendante et est séparée du reste de la mémoire vive de l'ordinateur.

S'il n'y a pas de module de mémoire vive connecté aux logements, on peut représenter la mémoire de l'ordinateur par le diagramme suivant. Remarquez le trait entre la mémoire du logement 0 et la mémoire principale. Ce trait illustre le résultat de l'instruction précédente.



Nota: Lorsque vous retirez un module mémoire, vous devez d'abord le déclarer comme mémoire vive indépendante sinon le retrait du module efface le reste de la mémoire principale.

Si vous essayez de déclarer comme mémoire indépendante un module qui l'est déjà ou s'il n'y a pas assez de mémoire principale inutilisée, le HP-71 affiche un message d'erreur.

Si le HP-71 ne possède pas assez de mémoire principale inutilisée pour vous permettre de déclarer un logement comme mémoire vive indépendante, vous devrez supprimer des fichiers de la mémoire principale (voir «Suppression de fichiers», page 115).

Le HP-71 contient 16K octets de mémoire vive dans le logement 0. Cet espace est divisé en quatre unités de 4K octets. Vous pouvez déclarer chacune de ces unités comme mémoire vive indépendante en exécutant l'ordre `FREE PORT` avec les paramètres suivants: 0, 0.01, 0.02 et 0.03. (Vous pouvez omettre le zéro d'en-tête). Exemple:

```
FREE PORT(.01)
```

déclare comme mémoire vive indépendante la deuxième unité de 4K octets du logement 0.

Rappel de la mémoire vive indépendante (CLAIM PORT)

Pour ré-incorporer un support de mémoire vive indépendant dans la mémoire vive principale, exécutez l'ordre CLAIM PORT.

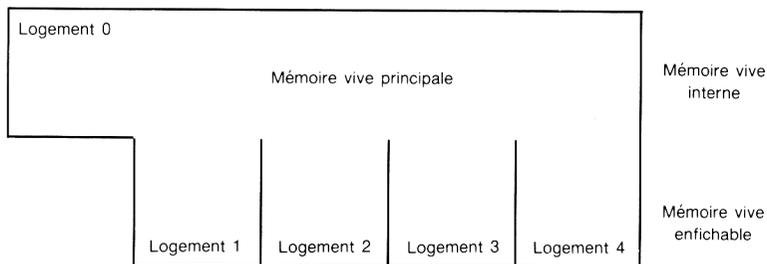
```
CLAIM PORT (Numéro de logement)
```

Le numéro de logement peut être un nombre entre 0 et 5 (bornes comprises). Le paramètre 5 représente le logement du lecteur de cartes.

Exemple: Ré-incorporez la mémoire du logement 0 dans la mémoire principale.

```
CLAIM PORT (0)
```

La mémoire du HP-71 se présente maintenant ainsi:



Nota: Lorsque vous ré-incorporez un support de mémoire vive indépendant, le HP-71 efface son contenu. Nous vous suggérons donc d'en copier le contenu en mémoire principale ou dans un autre support de stockage avant de le ré-incorporer.

Obtention d'informations sur la mémoire (MEM, SHOW PORT)

Lors de la création de fichiers, de leur stockage et copie, il est utile de connaître la capacité de stockage d'un module et l'espace de stockage libre. Cette information est particulièrement utile lorsque vous devez déterminer l'espace mémoire à libérer pour créer un support de mémoire vive indépendant.

Détermination de la quantité de mémoire inutilisée. Vous pouvez déterminer la quantité de mémoire inutilisée en mémoire principale ou dans un support de mémoire vive indépendant en exécutant l'ordre MEM.

```
MEM [(numéro de logement)]
```

Cet ordre affiche une valeur entière indiquant le nombre d'octets de mémoire inutilisés.

Exemples:

MEM	Quantité de mémoire inutilisée dans la mémoire principale.
MEM(A)	Quantité de mémoire inutilisée dans le logement spécifié par la valeur de la variable A.
MEM(1)	Quantité de mémoire inutilisée dans le logement 1.

Détermination de la capacité mémoire. Vous pouvez déterminer la taille (en octets) et le type de mémoire d'un logement en utilisant l'ordre `SHOW PORT`.

```
SHOW PORT
```

Lorsque vous exécutez cet ordre, l'ordinateur vous indique le numéro du logement, sa capacité mémoire en octets et le type de mémoire pour les modules (ROM ou RAM) en commençant par le logement de plus petit numéro. Le type de mémoire 1 représente un module de mémoire vive et le type 2 représente un module de mémoire morte.

Exemple: Déclarez le logement 0 comme support de mémoire indépendant et cherchez la taille et le type des logements.

Entrée/Résultat

```
MEM 
```

Détermine s'il y a suffisamment de place en mémoire principale pour déclarer le logement 0 comme un support mémoire indépendant.

```
5025
```

Affiche le nombre d'octets inutilisés de la mémoire principale. Si la valeur est supérieure ou égale à 4096 octets, vous avez de mémoire. Si la valeur affichée assez est inférieure à 4096, vous devrez supprimer des fichiers avant de pouvoir déclarer le logement comme support indépendant.

```
FREE PORT(0) 
```

Déclare le logement 0 comme support indépendant.

```
SHOW PORT 
```

Donne le type et la taille des supports mémoire dans les logements.

```
0      4096  1
```

Affiche les renseignements concernant le logement 0: mémoire vive de 4096 octets.

Noms de fichier

Chaque fichier que vous créez en mémoire porte un nom unique. Lorsque vous désirez effectuer des opérations sur un fichier, vous identifiez ce dernier par son nom. Avant de commencer à créer des fichiers, il est donc judicieux de connaître les règles utilisées pour les noms de fichiers.

Caractéristiques des noms de fichiers

Une nom de fichier peut être une combinaison quelconque de huit lettres ou chiffres commençant par une lettre. Les caractères autres que lettres et chiffres ne sont pas autorisés. Vous pouvez utiliser des majuscules ou des minuscules mais le HP-71 convertit tous les noms en majuscules.

Exemples de noms de fichiers valides et non valides

Non valide	Raison	Valide
TANGENTIEL 4DATA Test:12\$	Trop long. Ne peut pas commencer par un chiffre. Ne peut pas comporter un symbole : ou \$.	TANGENTE DATA4 Test12

Un nom de fichier peut être une chaîne sans guillemets ou une expression alphanumérique* donnant un nom valide.

Exemple:

`COPY ETUDE`

Chaîne sans guillemets.

`COPY "ETUDE"`

Chaîne entre guillemets identifiant le même fichier.

`COPY A$`

Le contenu de la variable A\$ doit être un nom de fichier valide.

`COPY A$ & B$`

Le résultat de l'expression alphanumérique doit être un nom de fichier valide.

Le HP-71 peut contenir plusieurs fichiers portant le même nom à condition qu'ils se trouvent dans des supports mémoire différents. Par contre, le HP-71 affiche un message d'erreur si vous essayez de stocker dans un support un fichier portant un nom déjà utilisé dans ce support. L'instruction suivante est valide:

`COPY RADIAL:PORT(0) TO RADIAL:MAIN`

* Une chaîne entre guillemets est la forme la plus simple d'une expression alphanumérique. Par conséquent lorsqu'une expression alphanumérique est un paramètre valide, une chaîne entre guillemets l'est aussi.

Par contre, l'instruction suivante n'est pas valide:

```
COPY RADIAL:PORT(0) TO RADIAL:PORT(0)
```

car le nom de fichier RADIAL existe déjà dans le logement 0.

Fichiers par défaut

Pour les ordres opérant sur des fichiers, vous pouvez parfois omettre de spécifier le nom du fichier. Le HP-71 utilise alors automatiquement un fichier par défaut (généralement le fichier courant).

Vous pouvez déterminer les fichiers par défaut pour tous les ordres en vous référant au dictionnaire des mots-clés dans le *Manuel de référence*.

Mots réservés

Le HP-71 atache une signification particulière à certains mots utilisés dans des ordres opérant sur des fichiers. Ces mots sont résumés dans le tableau suivant:

Mots réservés du HP-71

Mot	Description
ALL	Utilisé dans certains ordres pour indiquer que toutes les options sont valides.
CARD	Spécifie le lecteur de cartes magnétiques.
INTO	Utilisé conjointement au mot-clé TRANSFORM.
KEYS	Spécifie le fichier courant de définitions de touche (keys).
TO	Utilisé conjointement à certains mots-clés tels que COPY et RENAME.

Vous ne pouvez pas utiliser ces mots comme noms de fichiers sauf s'ils sont spécifiés sous forme d'expressions alphanumériques. Nous vous recommandons de ne jamais les utiliser pour éviter toute ambiguïté.

Noms d'unité

Pour les opérations sur fichiers, vous pouvez spécifier *l'emplacement* d'un fichier en plus de son nom. Cette caractéristique accélère les recherches de fichiers et permet de lever les ambiguïtés lorsque le même nom est utilisé sur des unités différentes. Dans la suite de ce chapitre, les noms d'unités sont des noms de support de mémoire, tel que MAIN.

Caractéristiques des noms d'unités

Les noms d'unités diffèrent des noms de fichiers car ils ne peuvent pas être créés par l'utilisateur. Le HP-71 ne reconnaît que certains noms d'unités et l'utilisation d'un autre nom provoque une erreur. Les mots suivants sont des noms d'unités valides sur le HP-71:

Noms d'unités valides

Nom d'unité	Description
PORT	Spécifie tous les logements en commençant au numéro 0.
PORT(<i>n</i>)	Spécifie un logement particulier (ou <i>n</i> est entre 0 et 5).
MAIN	Spécifie la mémoire vive principale.
CARD	Spécifie le lecteur de cartes magnétiques.
PCRD	Spécifie un fichier privé sur le lecteur de cartes magétiques.

Pour spécifier une unité dans un ordre tel que COPY ou PURGE, son nom doit être précédé du symbole `:`. Ce symbole indique que ce nom correspond à une unité de mémoire et non pas à un fichier. Exemple:

`COPY nom de fichier TO :PORT(0)` copie un fichier dans le logement 0.

Si vous voulez spécifier à la fois un fichier et son emplacement, utilisez la forme:

`nom de fichier : nom d'unité`

Exemple:

`COPY FICHER1:MAIN TO :PORT(0)`

copie le fichier FICHER1 de la mémoire principale dans le logement 0.

Unité par défaut

Dans de nombreux cas, si vous omettez de spécifier le nom d'unité, l'ordinateur utilise un nom d'unité par défaut et applique les règles suivantes:

- Si vous spécifiez un nom de fichier, l'ordinateur le cherche d'abord dans la mémoire principale puis dans les logements en commençant au logement numéro 0.
- Si vous ne spécifiez pas de nom de fichier, l'ordinateur considère comme unité par défaut celle dans laquelle réside le fichier courant.

Si vous ne spécifiez pas d'unité pour un fichier destination dans une opération COPY, l'unité par défaut est la mémoire principale.

Ordre de recherche des fichiers

Lorsque l'ordinateur doit chercher un fichier en mémoire (lorsque vous ne spécifiez pas d'unité), il le cherche d'abord dans la mémoire principale puis dans les différents logements en commençant au logement 0.

Si l'unité est représenté par le mot `PORT` sans paramètre, le HP-71 cherche le fichier dans les logements en commençant à celui de plus petit numéro. Si vous spécifiez le numéro du logement, le HP-71 ne cherche le fichier que dans ce logement. De même si vous spécifiez `MAIN` comme unité, le HP-71 ne cherche le fichier que dans la mémoire principale.

Copie de fichier (COPY)

L'ordre `COPY` vous permet de stocker et de rappeler des fichiers de la mémoire principale, des supports indépendants, des cartes magnétiques et même des modules de mémoire morte. Cet ordre crée simplement une copie du fichier spécifié à l'emplacement spécifié ou à l'emplacement par défaut.

— syntaxe simplifiée —

```
COPY [fichier source] TO [fichier destination]
```

Cet ordre spécifie deux fichiers:

- Le *fichier source* (celui à copier).
- Le *fichier destination* (celui qui devra contenir la copie).

Lors de l'exécution de `COPY`, l'ordinateur crée le fichier destination et y stocke une copie du fichier source. Le fichier destination ne doit pas exister préalablement.

Les fichiers source et destination peuvent être spécifiés par des noms de fichier, des noms d'unité ou les deux. Vous pouvez en outre exécuter `COPY` sans spécifier explicitement un fichier ou une unité. Vous pouvez donc spécifier un fichier de l'une des façons suivantes:

- *nom de fichier*
- *: nom d'unité*
- *nom de fichier : nom d'unité*
- pas de nom de fichier ni de nom d'unité.

Lorsque vous omettez un nom de fichier, un nom d'unité ou les deux, l'ordinateur utilise les noms par défaut tels que décrits précédemment.

Le tableau suivant résume les effets de `COPY` avec les différentes combinaisons de noms de fichier et de noms d'unité spécifiés et leurs valeurs par défaut.

Effets de COPY avec divers paramètres

Source*		Destination*		Réponse de l'ordinateur et exemple
Fichier	Unité	Fichier	Unité	
				Copie le fichier courant, s'il est en mémoire indépendante ou morte, dans la mémoire principale. Le fichier destination a le même nom que le fichier courant. COPY
			X	Copie le fichier courant dans le support spécifié. Le fichier destination a le même nom que le fichier courant. COPY TO :PORT(2)
		X		Copie le fichier courant en mémoire principale. le fichier destination a le nom spécifié. COPY TO NEW1
		X	X	Copie le fichier courant dans le support spécifié. le fichier destination a le nom spécifié. COPY TO NEW1:PORT(2)
	X			Copie le fichier de CARD ou PCRD dans la mémoire principale. Vous ne pouvez pas spécifier d'autre unité. Le fichier destination a le même nom que le fichier sur carte. COPY :PCRD
	X		X	Valide seulement si CARD ou PCRD est l'unité de fichier source et MAIN est l'unité de fichier destination. Copie un fichier sur carte dans la mémoire principale. Le fichier destination a le même nom que le fichier sur carte.† COPY CARD TO :MAIN
	X	X		Si l'unité spécifiée est CARD ou PCRD, copie le fichier sur carte en mémoire principale. Si vous spécifiez une unité différente, copie un fichier de l'unité spécifiée dans la mémoire principale. L'ordinateur cherche un fichier source portant le même nom que le fichier destination spécifié.† COPY :PORT(0) TO NAME1
	X	X	X	Si l'unité source est CARD ou PCRD, l'unité destination doit être MAIN; le HP-71 copie alors le fichier sur carte dans la mémoire principale. Si l'unité source est différente, cherche un fichier source portant le même nom que le fichier destination et le copie dans le fichier destination spécifié de l'unité spécifiée‡. Le fichier destination porte le nom spécifié. COPY :MAIN TO NAME1:PORT(0)

* Un X indique que le paramètre est spécifié.

† Lorsque seule une unité est spécifiée pour la source (autre que CARD ou PCRD), vous devez spécifier un nom pour la destination.

‡ Les fichiers du lecteur de cartes ne peuvent être copiés que dans la mémoire principale. Donc, si vous spécifiez CARD ou PCRD comme unité source, vous ne pouvez spécifier que MAIN comme unité destination (pour plus d'informations sur l'utilisation du lecteur de carte, référez-vous à l'annexe C «Utilisation du lecteur de cartes magnétiques HP 82400A»).

Effets de COPY avec divers paramètres (suite)

Source*		Destination*		Réponse de l'ordinateur et exemple
Fichier	Unité	Fichier	Unité	
X				Copie le fichier spécifié en mémoire principale. Le fichier destination porte le même nom que le fichier source. COPY ANCIEN1
X			X	Copie le fichier spécifié dans l'unité spécifiée. Le fichier destination porte le même nom que le fichier source. COPY ANCIEN1 TO :PORT(3)
X		X		Copie le fichier spécifié en mémoire vive principale. Le fichier destination porte le nom spécifié. COPY ANCIEN1 TO NOUVEAU1
X		X	X	Copie le fichier spécifié dans l'unité spécifiée. Le fichier destination porte le nom spécifié. COPY ANCIEN1 TO NOUVEAU1:PORT(1)
X	X			Copie le fichier spécifié de l'unité spécifiée dans la mémoire principale. Le fichier destination porte le même nom que le fichier source. COPY ANCIEN1 TO:PORT(3)
X	X		X	Copie le fichier spécifié de l'unité spécifiée dans l'unité de destination spécifiée. Le fichier destination porte le même nom que le fichier source. COPY ANCIEN1:PORT(3) TO :PORT(1)
X	X	X		Copie le fichier spécifié de l'unité spécifiée dans la mémoire principale. Le fichier destination porte le nom spécifié. COPY ANCIEN1:PORT(3) TO NOUVEAU1
X	X	X	X	Copie le fichier spécifié de l'unité spécifiée dans l'unité de destination spécifiée. Le fichier destination porte le nom spécifié. COPY ANCIEN1:PORT(3) TO NOUVEAU1:PORT(1)

* Un X indique que le paramètre est spécifié.

Changement de nom de fichier (RENAME)

Vous pouvez changer le nom d'un fichier à l'aide de l'ordre RENAME.

— syntaxe simplifiée —

```
RENAME [ancien nom de fichier] TO nouveau nom de fichier
```

Un *ancien nom de fichier* peut être exprimé par:

- *nom de fichier*
- *nom de fichier : nom d'unité*
- rien (le HP-71 utilise le fichier courant par défaut).

Un *nouveau nom de fichier* peut être exprimé par:

- *nom de fichier*
- *nom de fichier : nom d'unité*

Il n'y a pas de nom par défaut pour un nouveau fichier, vous devez toujours le spécifier. Vous pouvez spécifier l'unité avec l'ancien ou le nouveau nom de fichier.

Exemples:

```
RENAME TO FICHER2
```

Remplace le nom du fichier courant par FICHER2.

```
RENAME FICHER1 TO FICHER2
```

Remplace le nom FICHER1 par FICHER2.

```
RENAME FICHER1:PORT(0) TO  
FICHER2
```

Remplace le nom du fichier FICHER1 dans le logement 0 par FICHER2.

```
RENAME FICHER1 TO FICHER:  
PORT(0)
```

Remplace le nom FICHER1 dans le logement 0 par FICHER2.

Suppression de fichier (PURGE)

Pour supprimer un fichier de la mémoire, exécutez l'ordre PURGE.

```
PURGE [nom de fichier [: unité]]
```

Si vous ne spécifiez pas de fichier, le HP-71 utilise par défaut le fichier courant. Si vous spécifiez une unité, vous devez aussi spécifier un nom de fichier.

Exemples:

```
PURGE PROTON:MAIN
```

Supprime le fichier PROTON de la mémoire principale.

```
PURGE
```

Supprime le fichier courant.

Vous pouvez supprimer tous les fichiers non protégés de la mémoire principale en exécutant `PURGE ALL`. `PURGE ALL` n'affecte pas les fichiers stockés dans les supports indépendants.

```
PURGE ALL
```

Protection de fichier

Le HP-71 vous permet d'effectuer de nombreuses opérations sur les fichiers, telles que le listage, des modifications et des copies. Cependant, dans certains cas, vous pouvez désirer restreindre les opérations possibles sur des fichiers. Le HP-71 vous permet de contrôler l'accès à vos fichiers et de les protéger contre les modifications, suppressions ou listages.

Protection du contenu (SECURE, UNSECURE)

Vous pouvez protéger un fichier contre les modifications ou les suppressions en utilisant l'ordre `SECURE`. Les protections activées par cet ordre peuvent être annulées par l'ordre `UNSECURE`.

```
SECURE [nom de fichier [:unité]]
```

```
UNSECURE [nom de fichier [:unité]]
```

Vous pouvez protéger des fichiers de tout type. Les fichiers protégés ne peuvent être modifiés ou supprimés mais peuvent être exécutés (fichier programme), listés, lus ou copiés.

Contrôle de l'accès (PRIVATE)

Vous pouvez interdire le listage, la copie et les modifications d'un fichier avec l'ordre `PRIVATE`.

```
PRIVATE nom de fichier [:unité]
```

L'ordre `PRIVATE` est permanent, vous ne pouvez pas en renverser les effets.

Cet ordre ayant des effets permanents, vous *devez* spécifier explicitement le nom du fichier affecté. Ceci assure que vous ne rendez pas privé par erreur le fichier courant par exemple.

Exemples:

```
PRIVATE EPHEM
```

```
PRIVATE AZIMUTH:PORT(0)
```

L'ordre PRIVATE n'opère que sur les fichiers programme. Vous pouvez exécuter ou supprimer un fichier privé en mémoire mais personne (y compris vous) ne peut le lister, le copier ni le modifier (vous pouvez copier les fichiers privés de cartes magnétiques en mémoire principale mais vous ne pouvez pas les copier dans d'autres parties de la mémoire ou sur des cartes magnétiques).

Utilisation de SECURE et de PRIVATE

Vous pouvez protéger *et* rendre privé un fichier programme. Le tableau suivant résume les types d'opérations possibles sur un fichier protégé par SECURE, PRIVATE ou le deux.

Opérations possibles sur des fichiers protégés

	UNSECURE (défaut)	SECURE
Non privé (défaut)	Exécution, listage, copie, modification, suppression	Exécution, listage, copie,
PRIVATE	Exécution suppression	Exécution

Si un fichier est protégé par SECURE et par PRIVATE, vous ne pouvez que l'exécuter. Si vous exécutez UNSECURE sur un tel fichier, vous pouvez alors le supprimer mais vous ne pourrez jamais le lister, le copier ou le modifier.

Catalogues de fichiers

Lorsque vous désirez obtenir des informations sur des fichiers en mémoire, vous pouvez utiliser CAT ALL pour lister le catalogue des fichiers:

Un catalogue vous donne les informations suivantes sur un fichier:

- Nom du fichier.
- Type de protection.
- Type du fichier.
- Taille du fichier (en octets).
- Date de création.
- Heure de création.
- Logement (si nécessaire) dans lequel se trouve le fichier.

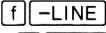
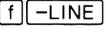
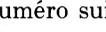
Le HP-71 affiche les informations concernant chaque fichier sous un format standard, appelé étiquette. Lorsque vous demandez le catalogue d'un groupe de fichiers, le HP-71 affiche d'abord un en-tête. Par exemple, lors de l'exécution de `CAT ALL`, l'ordinateur affiche d'abord:

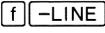
```
NOM      S TYPE      LEN      DATE      TIME PORT
```

Pour obtenir les catalogues de tous les fichiers de la mémoire principale, des supports indépendants et des modules de mémoire morte (appelé collectivement unités de mémoire), exécutez `CAT ALL`.

```
CAT ALL
```

Cet ordre affiche le catalogue des fichiers de chaque unité de mémoire, en commençant par la mémoire principale. Les touches  et  vous permettent de vous déplacer dans le catalogue courant pour afficher les étiquettes des différents fichiers. Vous pouvez vous déplacer rapidement au début et à la fin du catalogue courant en appuyant sur  et sur .

Lorsque vous voulez afficher le catalogue de l'unité suivante, appuyez sur . Si par exemple, l'étiquette d'un fichier de la mémoire principale est affichée, la pression de  vous permet de passer au catalogue du logement 0 (en supposant que ce dernier soit déclaré comme support indépendant). Une nouvelle pression de  passe au catalogue de l'unité de numéro suivant. Le HP-71 ré-affiche l'en-tête de catalogue entre les catalogues des différentes unités.

Si après la pression de  vous voulez revenir à une unité de numéro inférieur ou à la mémoire principale, vous devez ré-exécuter `CAT ALL`.

Le tableau suivant résume les séquences de touches qui vous permettent d'afficher les différentes étiquettes lorsque vous appelez un catalogue.

Séquences de touches pour l'affichage des étiquettes

Séquence	Réponse de l'ordinateur
▼	Affiche l'étiquette suivante.
▲	Affiche l'étiquette précédente.
g ▼	Affiche la dernière étiquette du catalogue.
g ▲	Affiche la première étiquette du catalogue.
f -LINE	Passes au catalogue de l'unité suivante. Après la dernière unité, affiche le symbole de mode BASIC.

Il est parfois utile d'afficher uniquement l'étiquette d'un seul fichier ou la catalogue d'un seul unité de mémoire. Vous pouvez obtenir des catalogues pour:

- Tous les fichiers de la mémoire principale, des supports indépendants et des modules de mémoire morte (CAT ALL).
- Tous les fichiers de la mémoire principale seulement (CAT:MAIN).
- Tous les fichiers de tous les logements (CAT:PORT).
- Tous les fichiers d'un logement (CAT:PORT(n)).
- Un fichier spécifié par son nom (CAT *nom de fichier*).
- Un fichier spécifié par son numéro d'ordre de stockage dans un support mémoire. c'est-à-dire l'ordre dans lequel il apparaît dans le catalogue de cette unité (CAT#(n)).

Si vous voulez obtenir des catalogues pour des fichiers ou des unités mémoire spécifiques, consultez les descriptions des mots-clés CAT et CAT# dans le *Manuel de référence* du HP-71.

Personnalisation du HP-71

Table des matières

Présentation	121
Redéfinition du clavier	121
Spécification de nom de touche	122
Types de définition de touche	124
Listage et modification de définition de touche (FETCH KEY, KEYDEF#, f VIEW)	125
Activation du clavier Utilisateur (USER, f USER, g 1USER)	126
Fichiers de touches (SECURE, UNSECURE, CAT, COPY, RENAME, PURGE, LIST, MERGE)	127
Annulation de définitions de touches	128
Interactions programme/clavier	129
Test de pression de touche (KEYDOWN)	129
Détermination de la touche utilisée (KEY#)	130
Pression de touche par programme (PUT)	131
Jeu de caractères secondaire	132
Définition de caractères (CHARSET, CHARSET#)	132
Conservation et destruction de caractères secondaires	125
Champs d'affichage protégés (WINDOW)	135
Lecture de caractères à l'affichage (DISP)	136
Affichage graphique	137
Lecture de colonnes de points à l'affichage (GDISP#)	137
Affichage de graphiques (GDISP)	137
Restriction de l'utilisation du HP-71 (LOCK)	139
Exécution automatique d'instructions (STARTUP)	139
Contrôle de l'affichage (LC)	140

Présentation

Ce chapitre vous présente:

- La puissance des touches définies par l'utilisateur.
- Le stockage de plusieurs jeux de définitions de touches.
- L'utilisation d'un clavier actif pendant l'exécution d'un programme.
- La création et l'utilisation d'un jeu de caractères personnalisé.
- La protection de parties de l'affichage contre l'entrée de caractères
- L'utilisation de caractères affichés dans un programme ou une définition de touche.
- Le contrôle de chacune des 132 colonnes de points de l'affichage.
- Le blocage du HP-71 contre les utilisations non autorisées.
- Quatre façons de contrôler comment le HP-71 affiche les informations.

Redéfinition du clavier (DEF KEY)

Vous pouvez redéfinir chaque touche du clavier sauf les deux touches **f** et **g** pour les utiliser comme aides à la frappe ou pour exécuter une ou plusieurs instructions. Vous pouvez non seulement redéfinir les fonctions primaires des touches mais aussi chacune des fonctions secondaires (précédées de **f** ou **g**). Le HP-71 place automatiquement les définitions de touches dans un fichier système spécial, `KEYS`, présenté plus loin dans ce chapitre (page 127).

```
[DEF] KEY nom de touche [, chaîne affectée [type d'affectation]]
```

Le mot-clé `DEF` est optionnel. Vous pouvez spécifier le *nom de touche* de deux façons, par le symbole imprimé (précédé de `F` ou `G` pour les positions secondaires) ou par son numéro d'identification. Si vous omettez la *chaîne affectée* (et le *type d'affectation*), cet ordre annule toute définition pour cette touche et la touche reprend sa fonction de clavier normal (par défaut).

La *chaîne affectée* définit l'aide à la frappe ou la séquence d'instructions affectée à la touche par l'ordre `DEF KEY`. Le *type d'affectation* est l'un des trois suivants: aide à la frappe, aide à la frappe/exécution ou exécution seule. Si vous omettez le type d'affectation, le HP-71 utilise par défaut le type aide à la frappe/exécution.

Vous pouvez spécifier le *nom de touche* et la *chaîne affectée* par des expressions alphanumériques valides *quelconques*, y compris les chaîne entre guillemets.

Spécification de nom de touche

Par caractère. La façon la plus simple de spécifier une touche est le plus souvent d'utiliser un ou deux caractères *entre guillemets*. Pour toutes les touches sauf *celles des lettres*, utilisez les règles suivantes:

- Pour la position primaire d'une touche, utilisez le caractère affiché par la touche seule.
- Pour les positions secondaires d'une touche, utilisez le caractère affiché par la touche seule, précédé de la lettre F ou G (vous pouvez utiliser f ou g).

Pour les touches de lettres, nous illustrerons la règle générale en utilisant [a] et [A] comme exemples. Tous les noms de touche suivants se réfèrent à la séquence produisant le caractère a: "g a", "g A", "G a", "G A" et "a". Le seul nom de touche se référant à la séquence affichant A est "A".

Lorsque les *minuscules sont en positions primaires* sur le clavier et que vous appuyez sur [A] sur le clavier Utilisateur, le HP-71 utilise la définition de la touche [a]. Si vous appuyez sur [a], le HP-71 utilise la définition de la touche [A].

Exemples de noms de touches DEF KEY

Nom de touche	Représente sur le clavier normal
"FH"	La touche [H] précédée de [f].
"<"	La touche [] précédée de [g].
"g"	La séquence produisant la lettre minuscule "g."
"Q"	La séquence produisant la lettre majuscule "Q."
"G "	La touche [SPC] précédée de [g].
"GB"	La séquence produisant la lettre minuscule "b."
"gd"	La séquence produisant la lettre minuscule "d."

Par numéro de touche. Certaines touches, telles que [ON], ne peuvent pas être redéfinies en utilisant un caractère de touche car elles n'affichent pas de caractère. Ces touches, sauf [f] et [g], peuvent être représentées par *un numéro de touche précédé du symbole #, le tout entre guillemets*. Comme indiqué dans les diagrammes suivants, les positions primaires sont numérotées de 1 à 56, les positions précédées de [f] sont numérotées de 57 à 112 et celles précédées de [g] sont numérotées de 113 à 168.

Le tableau suivant illustre les relations entre les numéros de touches, les caractères affichés, la séquence et le type de lettre (majuscule ou minuscule).

Exemples de numérotation de touche

	Jeu de majuscules	Jeu de minuscules
La touche numéro 15 représente	A et [A]	A et [g][A]
La touche numéro 127 représente	a et [g][A]	a et [A]

Remarquez que le numéro de touche 15 représente toujours la lettre H majuscule, mais que la séquence représentée dépend de l'option du clavier (majuscule ou minuscule) lors de l'utilisation de la définition sur le clavier Utilisateur.

Les diagrammes suivants montrent que neuf nombres ne sont pas utilisés 44, 45, 52, 100, 101, 108, 156, 157 et 164. Ces nombres identifient les touches [f], [g] et la moitié basse de la touche [END LINE]. Ces valeurs supposent le clavier en majuscules. Les nombres correspondants à [f] et [g] ne sont pas utilisables.

Numéro d'identification des touches

Position primaire

Code Touche	1 Q	2 W	3 E	4 R	5 T	6 Y	7 U	8 I	9 O	10 P	11 7	12 8	13 9	14 /
Code Touche	15 A	16 S	17 D	18 F	19 G	20 H	21 J	22 K	23 L	24 =	25 4	26 5	27 6	28 *
Code Touche	29 Z	30 X	31 C	32 V	33 B	34 N	35 M	36 (37)	38 END L I N E	39 1	40 2	41 3	42 -
Code Touche	43 ON	f	g	46 RUN	47 ◀	48 ▶	49 SPC	50 ▲	51 ▼		53 0	54 .	55 ,	56 +

Touches préfixées par [f]

Code Touche	57 IF	58 THEN	59 ELSE	60 FOR	61 TO	62 NEXT	63 DEF	64 KEY	65 ADD	66 LR	67 PREDV	68 MEAN	69 SDEV	70 SQR
Code Touche	71 CALL	72 GOSUB	73 RETURN	74 GOTO	75 INPUT	76 PRINT	77 DISP	78 DIM	79 BEEP	80 FACT	81 SIN	82 COS	83 TAN	84 EXP
Code Touche	85 EDIT	86 CAT	87 NAME	88 PURGE	89 FETCH	90 LIST	91 DELETE	92 AUTO	93 COPY	94 RES	95 ASIN	96 ACOS	97 ATAN	98 LOG
Code Touche	99 OFF			102 SST	103 BACK	104 -CHAR	105 I/R	106 LC	107 -LINE		109 USER	110 VIEW	111 CALC	112 CONT

Touches préfixées par [g]

Code Touche	113 q	114 w	115 e	116 r	117 t	118 y	119 u	120 i	121 o	122 p	123 '	124 {	125 }	126 ^
Code Touche	127 a	128 s	129 d	130 f	131 g	132 h	133 j	134 k	135 l	136 ;	137 \$	138 %	139 &	140 :
Code Touche	141 z	142 x	143 c	144 v	145 b	146 n	147 m	148 [149]	150 CMDS	151 !	152 "	153 #	154 @
Code Touche	155			158 CTRL	159 ◀	160 ▶	161 ERRM	162 ▲	163 ▼		165 1 USER	166 <	167 >	168 ?

Exemple:

"#94"

Cette chaîne représente f RES.**Types de définitions de touche**

Aide à la frappe (;). Si vous faites suivre la *chaîne d'affectation* d'un point-virgule, la pression de la touche sur le clavier Utilisateur affiche la chaîne mais ne l'exécute pas.

Aide à la frappe/exécution. Si aucun symbole ne suit la *chaîne d'affectation*, la pression de la touche affiche la chaîne puis l'exécute. Le HP-71 affiche la chaîne à l'emplacement courant du curseur et essaie d'exécuter la ligne entière, y compris tous les caractères qui pouvaient être présents à l'affichage avant la pression de la touche. Ce type de définition est utile comme aide à la frappe pour afficher la dernière partie d'une instruction ou d'une ligne à entrer.

Exécution seule (:). Si vous faites suivre la *chaîne d'affectation* du symbole deux-points, la pression de la touche sur le clavier Utilisateur exécute la chaîne sans l'afficher. Le HP-71 ignore tous les caractères affichés lorsque vous utilisez une touche de ce type. Vous pouvez utiliser ces définitions pour répondre à des instructions d'entrée d'informations par exemple.

Exemples: Les exemples suivants supposent le clavier en mode majuscule.

```
DEF KEY "c", 'RUN "HORLOGE"');
```

On suppose que le HP-71 contient un programme nommé HORLOGE. Lorsque vous appuyez sur g C sur le clavier Utilisateur après l'exécution de cette définition, le HP-71 lance l'exécution du programme HORLOGE. Les guillemets autour de HORLOGE sont optionnels.

```
KEY "#94", CHR#(92);
```

Lorsque vous appuyez sur f RES sur le clavier Utilisateur après l'exécution de cette définition, le HP-71 affiche l'opérateur de division entière, \. Remarquez que CHR#(92) n'est pas entre guillemets. Si CHR#(92) se trouvait entre guillemets, la pression de f RES afficherait la chaîne CHR#(92) et non pas l'opérateur \.

```
KEY "fC", "CHR#(");
```

Cette instruction affecte une aide à la frappe à la séquence f C. Cette aide à la frappe simplifie la frappe de la fonction CHR#.

```
DEF KEY "F3", "P3=74,95"
```

Aucun symbole ne suit la chaîne d'affectation, la pression de f ATAN affiche donc l'instruction d'affectation de variable et l'exécute.

Listage et modification de définition de touche (FETCH KEY, KEYDEF#,)

L'ordre FETCH KEY affiche l'instruction DEF KEY spécifiée pour modification. La fonction KEYDEF# donne la chaîne d'affectation de l'instruction DEF KEY spécifiée (l'aide à la frappe ou l'instruction affectée à la touche) pour affichage uniquement.

```
FETCH KEY nom de touche
```

Les ordres FETCH KEY et DEF KEY utilisent tous deux la même spécification de *nom de touche*.

Exemples: L'exemple suivant illustre l'utilisation de FETCH KEY pour afficher deux des définitions de touche indiquées auparavant.

Entrée/Résultat

```
FETCH KEY "c" 
```

Affiche la définition de touche affectée à c.

```
>DEF KEY 'c', 'RUN "HOR
LOGE";
```

Le HP-71 affiche cette définition de touche telle que vous l'avez entrée (guillemets optionnels autour de HORLOGE). Vous pouvez modifier et ré-entrer cette définition.

```
FETCH KEY "#94" 
```

Vous devez utiliser le numéro de touche pour , car le HP-71 ne reconnaît pas RES comme symbole de .

```
>DEF KEY '#94', 'I';
```

```
KEYDEF# (nom de touche)
```

La fonction KEY DEF# utilise la même spécification de *nom de touche* que les ordres DEF KEY et FETCH KEY.

Exemples: Les exemples suivants illustrent l'utilisation de KEYDEF# pour afficher les deux autres définitions de touche effectuées.

Entrée/Résultat

KEYDEF#("FC") END LINE

```
;CHR#<
```

Le HP-71 affiche la chaîne affectée à **f****C**, précédée de ; pour indiquer que la définition est une aide à la frappe.

KEYDEF#("F3") END LINE

```
P3=74.95
```

L'espace précédant l'instruction d'affectation de variable identifie la définition comme étant du type aide à la frappe/exécution.

La pression de **f****VIEW** vous permet d'afficher la chaîne d'affectation de la touche sur laquelle vous appuyez ensuite, précédée d'un point-virgule, d'un blanc ou de deux points. La définition reste affichée tant que vous maintenez la touche enfoncée. Si la touche ne possède pas d'affectation, le HP-71 affiche Unassigned. La pression de **f****VIEW** suivi de **g****C** (majuscules en positions primaires) par exemple, permet d'afficher la définition de cette touche, :RUN "HORLOGE", tant que vous maintenez **C** enfoncée.

Activation du clavier Utilisateur (**USER**, **f****USER**, **g****1****USER**)

Lorsque le clavier Utilisateur est actif, le HP-71 allume le témoin **USER** dans l'affichage.

L'ordre **USER** possède trois formes:

```
USER
USER ON
USER OFF
```

L'ordre **USER** vous permet de basculer entre le clavier normal et le clavier Utilisateur. La séquence de touches **f****USER** a le même effet.

L'ordre **USER ON** active le clavier Utilisateur s'il ne l'est pas déjà.

L'ordre **USER OFF** désactive le clavier Utilisateur s'il est actif.

La séquence **g****1****USER** active ou désactive le clavier Utilisateur pour la séquence suivante uniquement. Cette touche est particulièrement utile lorsque vous voulez utiliser temporairement une seule touche du clavier Utilisateur.

Fichiers de touches (SECURE, UNSECURE, CAT, COPY, RENAME, PURGE, LIST, MERGE)

Lorsque vous effectuez votre première définition de touche, le HP-71 crée automatiquement un fichier spécial nommé `keys` dans lequel il stocke la définition de touche et toutes les suivantes. Vous pouvez manipuler ces fichiers comme tous les autres avec les ordres `LIST`, `COPY`, `RENAME`, `MERGE`, `SECURE`, `UNSECURE` et `PURGE` et vous pouvez en afficher l'étiquette avec `CAT KEYS` et `CAT ALL`.

Fichier des définitions de touches courantes. Toutes les définitions de touches *courantes* sont stockées dans le fichier `keys`. Lorsque vous activez le clavier Utilisateur, le HP-71 active toutes les définitions de touches que le fichier `keys` contient. Vous ne pouvez pas rendre un fichier de touches privé mais vous pouvez le protéger avec l'instruction `SECURE KEYS` et le déprotéger avec `UNSECURE KEYS` (voir page 116 la présentation des méthodes de protection des fichiers).

Création de plusieurs fichiers de définitions de touches. Vous pouvez créer plusieurs fichiers de définitions de touches mais seules les définitions stockées dans le fichier nommé `keys` sont actives lorsque vous passez au clavier Utilisateur. Vous pouvez copier les définitions du fichier `keys` courant dans un autre fichier de façon à pouvoir rappeler un autre jeu de définitions de touches. Vous pouvez utiliser les ordres suivants pour échanger le fichier `keys` et un autre fichier de touches:

- `RENAME KEYS TO nom de fichier`
- `COPY KEYS TO nom de fichier`
- `COPY nom de fichier TO KEYS`
- `RENAME nom de fichier TO KEYS`

Listage du fichier `keys`. L'ordre `LIST KEYS` affiche toutes les définitions de touches actives dans l'ordre de leurs numéros de touche, tels que définis en page 123. Le HP-71 affiche chaque définition dans le format `DEF KEY` pendant l'intervalle de temps défini par l'ordre `DELAY` (page 26). Vous pouvez aussi lister uniquement une partie des définitions de touches du fichier `keys` en spécifiant un ou deux numéros de touches comme paramètres de l'ordre.

Exemples:

```
LIST KEYS,1,14
```

Liste toutes les définitions de touches affectées aux positions primaires des touches du premier rang.

```
LIST KEYS,24
```

Affiche la définition de touche affectée à la touche .

Fusion d'un fichier de touches à keys. Vous pouvez fusionner un fichier de touches, créé préalablement, au fichier `keys`.

```
MERGE nom de fichier [:unité][, numéro de première touche [, numéro de dernière touche]]
```

Le *nom de fichier*, avec en option le nom d'*unité*, spécifie le fichier de touches que vous voulez fusionner au fichier `keys`. Les définitions dans tous les fichiers de touches sont classées par ordre croissant des numéros de touche. Si vous ne spécifiez pas de *numéro de touche*, le HP-71 fusionne le fichier de touches complet à `keys`. Si vous ne spécifiez que le *numéro de première touche*, le HP-71 fusionne uniquement la définition correspondant à ce numéro à `keys`. Si vous spécifiez les deux numéros de touches, le HP-71 fusionne le bloc de touches ainsi défini, bornes comprises, à `keys`. Lorsque l'instruction fusionne des définitions pour des touches déjà définies à `keys`, la nouvelle définition remplace l'ancienne. Cet ordre n'altère pas le fichier de touches fusionné. Après l'exécution de `MERGE`, le fichier spécifié est inchangé; seul le fichier `keys` est modifié.

Exemples:

```
MERGE KEYS1,1,14
```

Fusionne les définitions affectées aux positions primaires des touches du premier rang et stockées dans `KEYS1` aux définitions de `keys`. Les nouvelles définitions remplacent les anciennes dans le fichier `keys` pour les positions primaires des touches du premier rang. `KEYS1` est inchangé.

```
MERGE KEYS2,67
```

Fusionne la définition de `KEYS2` affectée à la séquence `f7` dans le fichier `keys`.

```
MERGE MATHKEYS
```

Fusionne le fichier `MATHKEYS` en entier à `keys`. Les définitions de touches de `MATHKEYS` sont ajoutées à celles de `keys` mais celles touches définies dans les deux fichiers auront les définitions spécifiées dans `MATHKEYS`.

Annulation de définitions de touches

L'exécution de l'ordre `DEF KEY` avec un nom de touche mais sans chaîne d'affectation annule l'affectation courante de la touche et supprime la définition du fichier `keys`.

Exemples:

```
KEY "#155"
```

Annule la définition affectée à `gON`.

Pour annuler toutes les définitions de touches courantes, il vous suffit d'exécuter l'ordre `PURGE KEYS`. Si vous voulez conserver une copie des définitions vous pouvez changer le nom du fichier `keys` avec un ordre `RENAME KEYS TO`.

Interactions programme/clavier

Dans certaines situations, il est utile de pouvoir répondre directement à un programme en cours d'exécution plutôt que de passer par un ordre INPUT. Le HP-71 vous offre trois types de réponse directe. Le programme peut:

- Tester si vous avez appuyé sur une touche quelconque ou sur une touche particulière.
- Déterminer quelle touche vous avez utilisée.
- "Appuyer" sur une touche.

Test de pression de touche (KEYDOWN)

```
KEYDOWN
```

Cette forme de la fonction KEYDOWN donne un 1 si une touche, y compris [f] et [g], est enfoncée au moment de l'exécution de la fonction et donne 0 dans le cas contraire.

```
KEYDOWN (nom de touche)
```

Cette forme de la fonction KEYDOWN donne un 1 si la touche identifiée est enfoncée et donne 0 dans le cas contraire. Le *nom de touche* peut être spécifié par un symbole ou par le numéro de touche. Cette fonction n'est valide qu'avec les positions primaires des touches.

Exemples:

```
K=KEYDOWN ("7")
```

Si vous maintenez la touche [7] enfoncée pendant l'exécution de cette instruction, cette dernière affecte la valeur 1 à la variable K.

```
X=KEYDOWN (CHR#(87))
```

Si vous maintenez la touche [W] enfoncée pendant l'exécution de cette instruction, cette dernière affecte la valeur 1 à la variable X.

```
K4=KEYDOWN ("#10")
```

Si vous maintenez la touche [P] enfoncée pendant l'exécution de cette instruction, cette dernière affecte la valeur 1 à la variable K4.

Détermination de la touche utilisée (KEY\$)

Le HP-71 peut stocker 15 séquences de touches dans une zone de mémoire spéciale appelée *tampon de clavier*. En règle générale, lorsque vous appuyez sur une touche, la séquence est momentanément stockée dans le tampon de clavier puis rappelée et l'action de la touche semble immédiate. Par contre, lorsque vous appuyez sur une touche pendant l'exécution d'un programme, la séquence est stockée dans le tampon.

Nota: Si un ordre `DISP` envoie un message de fin de ligne à l'affichage et si le délai de ligne n'est pas nul, le tampon est ré-initialisé (il ne reste plus de séquence de touches dans le tampon).

```
KEY$
```

Cette fonction rappelle la séquence la plus ancienne dans le tampon et en donne le nom dans le même format que celui utilisé par `DEF KEY`, `KEYDOWN` et `KEYDEF$`.

Exemple: Le programme suivant transforme votre HP-71 en un compteur à deux canaux. La pression de `[+]` ajoute un au premier compteur et celle de `[−]` ajoute un au second. L'affichage indique en permanence la valeur des deux compteurs. Vous pouvez appuyer sur `[R]` pour remettre les compteurs à zéro.

Exécutez `EDIT COMPTEUR` et entrez le programme suivant.

```
10 ! COMPTEUR
20 STD @ DIM A$,I,J           Affichage sans point décimal.
30 DELAY 0,0                 Permet la vitesse d'enregistrement maximum.
40 I=0 @ J=0
50 DISP I;TAB(10);J
60 A$=KEY$
70 IF A$="−" THEN I=I+1 @ GOTO 50   La pression de [−] incrémente le compteur de gauche.
80 IF A$="+" THEN J=J+1 @ GOTO 50   La pression de [+] incrémente le compteur de droite.
90 IF A$="R" THEN 40 ELSE 60
```

Si vous n'appuyez sur aucune touche pendant l'exécution de ce programme, le HP-71 exécute répétitivement les lignes 60 à 90. Si vous appuyez sur `[−]`, le HP-71 incrémente le compteur `I` de un et si vous appuyez sur `[+]`, il incrémente le compteur `J` de un. Si vous appuyez sur `[R]`, le HP-71 remet les deux compteurs à zéro.

La clé de ce programme est la ligne 60. A chaque fois que le HP-71 exécute $A\# = KEY\#$, il affecte à $A\#$ le nom de la première touche enfoncée depuis la dernière exécution de la ligne 60.

Si vous appuyez sur plusieurs touches entre deux exécutions de la ligne 60, le HP-71 n'affecte que le nom de la première à la variable $A\#$. Si vous appuyez trop rapidement sur les touches, le tampon de clavier se remplit et vous risquez de perdre des pressions de touches dans votre décompte.

En pratique, il est très difficile d'appuyer sur plus d'une touche entre deux exécutions de la ligne 60, il y a donc peu de risques que vous remplissiez le tampon. Néanmoins, si la durée entre deux exécutions de $KEY\#$ dans un autre programme est suffisamment longue, un utilisateur de ce programme peut remplir le tampon, perdre ainsi des pressions de touches et peut-être provoquer un mauvais fonctionnement du programme. Lorsque vous utilisez la fonction $KEY\#$ dans un programme, veillez à ne laisser qu'un court instant entre deux exécutions de $KEY\#$.

Lorsque vous exécutez le programme COMPTEUR, veillez à relâcher une touche avant d'appuyer sur la suivante de façon que le programme ne manque pas de touche.

Pour arrêter le programme, appuyez sur `ATTN` puis exécutez END.

Pression de touche par programme (PUT)

```
PUT nom de touche
```

Vous devez spécifier le *nom de touche* dans le même format que pour DEF KEY. Lors de l'exécution de PUT dans un programme, son effet est généralement similaire à celui de la pression de la touche indiquée sur le clavier.

Exemple: Ce programme HORLOGE affiche l'heure et la date de façon continue au lieu de l'affichage immobile des fonctions TIME $\#$ et DATE $\#$. Ce programme illustre en outre la compression des instructions par concaténation. Créez un programme en exécutant EDIT HORLOGE et entrez les instructions suivantes.

```
10 DELAY 0 @ FOR X=1 TO 30 @ DISP           Placez trois espaces entre les guillemets.
   TIME$& " " "& DATE$ @ NEXT X @
   DELAY 1 @ PUT "#43"
```

Comme le diagramme de la page 123 l'indique, le numéro 43 correspond à la touche `ON`. La pression de `ON` (c'est-à-dire `ATTN` lorsque le HP-71 est allumé) efface l'affichage puis affiche le symbole BASIC et le clavier Utilisateur. Cette opération n'interrompt pas un programme en exécution et le HP-71 n'allume pas le témoin **SUSP**. L'instruction PUT "#43" place la touche `ATTN` dans le tampon de clavier mais n'arrête pas l'exécution du programme. Dans l'exemple, le programme s'arrête après l'exécution de la dernière instruction (qui se trouve être PUT "#43"). A ce moment `ATTN` quite la mémoire tampon et l'affichage est effacé.

L'effet de `PUT "#43"` dans un programme est identique à la pression de `[ATTN]` lorsque l'exécution du programme est terminée. Vous pouvez vérifier cela en ajoutant un `!` juste avant `PUT "#43"` pour convertir cette instruction en une remarque. Exécutez à nouveau le programme `HORLOGE`. Lorsque le programme se termine (lorsque le témoin `PRGM` s'éteint), l'affichage conserve l'heure et la date de la dernière exécution. Appuyez sur `[ATTN]` pour effacer l'affichage, afficher le symbole `BASIC` et le curseur.

Si vous retirez le `!`, l'effacement de l'affichage et la répartition du symbole `BASIC` et du curseur ont lieu sous contrôle du programme.

Comme vous avez affecté l'instruction `RUN "HORLOGE"` à la séquence `[9][C]`, vous pouvez afficher l'heure et la date à tout moment, même lorsque le clavier Utilisateur n'est pas actif, en maintenant `[9]` enfoncée et en appuyant sur `[0]` et `[C]`. Souvenez-vous que `[9][1]USER` active le clavier Utilisateur pour la séquence de touches suivante uniquement.

Jeu de caractères secondaire

Les caractères de codes ASCII compris entre 128 et 255 représentent normalement les mêmes caractères que ceux de codes compris entre 0 et 127 comme l'indique le tableau à la page 323 du *Manuel de référence*. Vous pouvez cependant redéfinir chacun de ces 128 caractères, à condition que vous puissiez représenter chaque nouveau caractère dans un rectangle de six points de large par huit points de haut.

Définition de caractères (`CHARSET`, `CHARSET#`)

Le HP-71 affecte automatiquement des codes aux caractères que vous définissez en commençant par 128, puis 129 etc. L'ordre `CHARSET` suffit pour la définition du premier caractère, mais pour tous les suivants, il est souvent pratique d'utiliser `CHARSET#` et `CHARSET`.

```
CHARSET chaîne de définition
```

```
CHARSET#
```

La *chaîne de définition* représente la configuration de points pour tous les caractères existants du jeu secondaire. La fonction `CHARSET#` donne cette chaîne. Lors de la création du premier caractère, vous exécutez une instruction `CHARSET#` spécifiant les six colonnes de points formant le caractère. Lors de la définition du deuxième caractère, vous devez spécifier les douze colonnes correspondant au premier et au second caractère. Vous pouvez alors utiliser la fonction `CHARSET#` pour représenter les six premières colonnes de façon à n'entrer que les colonnes du deuxième caractère. Lorsque vous créez le neuvième caractère, la fonction `CHARSET#` vous permet de représenter les huit premiers caractères (c'est-à-dire 48 colonnes) et il ne vous reste qu'à entrer les six colonnes du caractère que vous voulez définir.

Définition du premier nouveau caractère. Le diagramme de la page suivante doit vous permettre de définir comme caractère numéro 128 (`CHR#(128)`) le symbole d'intégrale \int . Voici la procédure à suivre.

1. Tracez une configuration de six par huit points et indiquez les points qui doivent être allumés pour représenter le caractère.
2. Chaque colonne représente un octet de la définition. Chaque rang d'une colonne représente un des huit bits de l'octet. Pour affecter les valeurs des points allumés, notez la valeur décimale de chaque bit à droite de chaque rang.
3. Additionnez maintenant les valeurs des points de chaque colonne pour obtenir les valeurs de chaque octets. Dans l'exemple, tous les points de la sixième colonne sont éteints, la valeur de l'octet est donc zéro. Il est judicieux de laisser la sixième colonne blanche pour séparer le caractère défini des suivants à l'affichage.
4. Vous êtes maintenant prêt à écrire l'instruction `CHARSET` comme suit.

Exemple: Lors de l'exécution de cette instruction, le HP-71 définit le 128^{me} caractère du jeu comme le symbole d'intégrale \int .

```
CHARSET CHR#(64)&CHR#(128)&CHR#(126)&CHR#(1)&CHR#(2)&CHR#(0)
```

Exécutez cette instruction puis exécuter `CHR#(128)` pour afficher le symbole d'intégrale.

La fonction `CHR#` sert dans `CHARSET` uniquement pour donner les valeurs d'octet de votre nouveau caractère au HP-71. L'expression `CHR#(64)` par exemple, normalement représentée par le caractère E , ne sert dans l'instruction `CHARSET` qu'à indiquer au HP-71 que l'octet représenté vaut 64.

Nouveau caractère: symbole d'intégrale

						Valeurs de bits des rangs	
						1	
						2	
						4	
						8	
						16	
						32	
						64	
						128	
64	128	2	1	2	0		
		4					
		8					
		16					
		32					
		64					
64	128	126	1	2	0	Valeurs d'octets des colonnes	

Définition de nouveaux caractères (`CHARSET#`, `CHARSET`). Le premier caractère défini remplace le caractère de code 128, le second vient prendre la place du code 129 etc. Si vous avez déjà défini le caractère et voulez maintenant ajouter le caractère 129, vous devez spécifier les configurations de points des deux caractères dans la chaîne de définition.

Si le nombre d'octets défini par l'instruction `CHARSET` n'est pas un multiple de six, le HP-71 affecte la valeur zéro (colonnes blanches) au(x) dernier(s) octets de la définition du dernier caractère. Vous auriez donc pu omettre le `CHR#(0)` dans l'exemple.

Cependant, si l'instruction `CHARSET` définit plusieurs caractères sans utiliser `CHARSET#` (s'il y a plus de six fonctions `CHR#` dans l'instruction `CHARSET`), et si vous désirez une colonne blanche pour séparer chaque caractère, vous devez spécifier `CHR#(0)` comme sixième colonne pour tous les caractères autres que le dernier car le HP-71 utilise automatiquement six octets pour chaque caractère.

Vous ne pouvez pas définir plus de 128 caractères secondaires dans un jeu. Les codes de caractère supérieurs à 255 identifient les caractères existants dont les codes sont compris entre 0 et 255 (modulo 256), y compris les caractères secondaires existants.

Conservation et destruction des caractères secondaires.

La fonction `CHARSET#` donnant les caractères du jeu secondaire actif, vous pouvez stocker un jeu de caractères secondaire en affectant le contenu de `CHARSET#` à une variable convenablement dimensionnée. Une variable alphanumérique dimensionnée par défaut peut contenir les définitions de cinq caractères. Chaque colonne de points est représentée par un caractère, la définition d'un caractère secondaire utilise donc six caractères et vous pouvez ainsi stocker cinq définitions de caractère dans 30 octets. Pour stocker plus de cinq caractères dans une variable alphanumérique, vous devez dimensionner cette dernière avant de lui affecter le contenu de `CHARSET#`.

Exécutez `CHARSET" "` pour détruire un jeu de caractères.

Champs d'affichage protégés (WINDOW)

Vous pouvez protéger une partie de l'affichage visible de façon que les caractères qui s'y trouvent ne soient pas affectés par la plupart des opérations. Si des caractères à la gauche de la fenêtre d'affichage sont protégés, le symbole BASIC et le curseur clignotant sont placés juste à droite de la partie protégée et définissent la limite gauche de l'affichage actif. Cette partie d'affichage actif est disponible pour les interactions normales avec l'affichage, telles qu'entrées de données, déroulement, affichage d'instructions `DISP`, listages de programme etc.

```
WINDOW première colonne [, dernière colonne]
```

La partie d'affichage définie par cette instruction est celle qui reste active. Le reste de l'affichage devient *protégé*. Les deux paramètres doivent être des expressions numériques arrondies à un entier. L'intervalle de valeurs de la *première colonne* est 1—22 et celui de la *dernière colonne* est première colonne—22.

Exemples:

```
WINDOW 5,22
```

Cette instruction protège les quatre premières colonnes de l'affichage et laisse actif le reste.

```
WINDOW 7
```

Cette instruction protège les six premières colonnes de l'affichage.

```
WINDOW 6,18
```

Cette instruction protège les cinq premières et les quatre dernières colonnes de l'affichage et laisse actives les colonnes 6 à 18.

La partie protégée de la fenêtre d'affichage reste inchangée jusqu'à ce que l'une des opérations suivantes ait lieu:

- Exécution d'un autre ordre WINDOW définissant un champ protégé différent. L'exécution de WINDOW 1, par exemple, réactive l'ensemble de l'affichage.
- Exécution d'un ordre GDISP (voir page 137) changeant la configuration de points du champ protégé. Cette ordre ne change pas les positions des zones active et protégée de l'affichage. La nouvelle configuration de points dans le champ protégé est immédiatement protégée.
- Exécution d'un ordre INIT: 1, INIT: 2 ou INIT: 3 (voir chapitre 1, page 13).
- Ré-initialisation de la mémoire due par exemple à une coupure d'alimentation.

Lecture de caractères à l'affichage (DISP\$)

La fonction DISP\$ donne une chaîne de longueur comprise entre 0 et 96 caractères contenant tous les caractères lisibles de l'affichage. Les caractères lisibles sont ceux envoyés à l'affichage lorsque le curseur est allumé. La fonction DISP\$ permet à une touche définie par l'utilisateur ou un programme prêt à être exécuté d'utiliser directement un nombre (VAL(DISP\$)) ou une chaîne affichée.

```
DISP$
```

Le HP-71 considère comme lisibles tous les caractères envoyés à l'affichage lorsque le curseur est allumé. La fonction DISP\$ ne donne donc pas les caractères envoyés à l'affichage lorsque le curseur est éteint. Si l'affichage est vide, la fonction DISP\$ donne la chaîne vide.

Exemples: Les deux définitions de touches suivantes facilitent les conversions entre les degrés Fahrenheit et les degrés Celsius. Ces exemples supposent le clavier en majuscules ([C] affiche C), le clavier Normal actif et le format d'affichage FIX 2.

```
DEF KEY "C", "(VAL(DISP$)-  
32)*5/9": [END LINE]
```

```
212 [G] [1 USER] [C]
```

```
DEF KEY "F", "(VAL(DISP$)  
*9/5)+32": [END LINE]
```

```
0 [G] [1 USER] [F]
```

Cette définition de touche à exécution seule affecte la formule de conversion de degrés Fahrenheit en degrés Celsius à la touche [C].

Affiche 100.00, l'équivalent en degrés Celsius de 212°F.

Cette définition de touche à exécution seule affecte la formule de conversion de degrés Celsius en degrés Fahrenheit à la touche [F].

Zéro degré Celsius est équivalent à 32° Fahrenheit.

Affichage graphique

Vous pouvez contrôler chacune des 132 colonnes de points de la fenêtre d'affichage. Vous pouvez y afficher tous les dessins que vous voulez et stocker ces derniers dans des variables. Pour cela vous utilisez les mêmes mots-clés que pour les caractères de jeu secondaire: `CHARSET` et `CHARSET#`. La suite de cette section vous présente deux autres mots-clés utilisés principalement pour l'affichage des *graphiques*. Nous vous donnerons quelques exemples de graphiques à la fin de cette section.

Lecture de colonnes de points à l'affichage (`GDISP#`)

```
GDISP#
```

Cette fonction donne toujours une chaîne de 132 caractères, où chaque caractère représente les points de l'une des 132 colonnes de l'affichage. Le premier caractère de cette chaîne correspond à la première colonne et le 132^{me} à la dernière. La fonction `GDISP#` traite chaque colonne de points comme un octet, de la même façon que `CHARSET` et `CHARSET#`. Le bit le moins significatif (valeur 1) de chaque octet correspond au rang supérieur de la colonne et le bit le plus significatif (valeur 128) correspond au rang inférieur.

Une fois un affichage lu par `GDISP#`, vous pouvez manipuler individuellement chacune des 132 colonnes. Souvenez-vous que les caractères représentant chaque colonne dans `GDISP#` ne sont pas les mêmes que les caractères normalement représentés par les variables alphanumériques. Chacun de ces caractères "normaux", tels qu'affichés par un ordre `DISP`, utilise six colonnes d'octets que vous ne pouvez pas contrôler individuellement.

Affichage de graphiques (`GDISP`)

Après la création d'une configuration de points avec `CHARSET` ou sa lecture avec `GDISP#`, vous pouvez l'afficher avec l'ordre `GDISP`. Cet ordre affiche des caractères d'une colonne de large et offre donc une plus grande souplesse d'affichage que `DISP`.

```
GDISP configuration de bits
```

La *configuration de bits* est la configuration de points exprimée sous forme d'une chaîne telle que celle créée par `CHARSET#` ou lue par `GDISP#`. Cette configuration de bits comporte toujours 132 caractères et chaque caractère représente la configuration de bits pour une colonne de points. Si la chaîne spécifiée contient plus ou moins de 132 caractères, le HP-71 la tronque ou lui ajoute des caractères nuls (`CHR#(0)`) représentant une colonne de points vierges.

L'ordre `GDISP` établit une configuration spécifiée de points dans l'affichage mais ne modifie pas le tampon d'affichage ou la zone de stockage contenant les 96 caractères de la ligne d'affichage.

La configuration de points définie par `GDISP` reste affichée jusqu'à modification de l'affichage par le clavier ou un programme. Parmi les actions retirant une configuration de points de l'affichage se trouvent:

- L'envoi d'un caractère à l'affichage (y compris un espace).
- La pression d'une des touches de flèches (`◀▶▲▼`).

L'ordre `GDISP` vous permet aussi d'afficher une configuration de points dans la zone protégée de l'affichage. Cette configuration se trouve alors protégée dès son affichage.

Exemple: Entrez le programme `CINEMA` suivant et exécutez-le. Lorsque le HP-71 vous demande une chaîne, entrez une chaîne entre 1 et 22 caractères. Veillez à exécuter `EDIT CINEMA` avant d'entrer les lignes de programme suivantes.

10 DISP "Programme CINEMA"	
20 DIM D\$[132],N\$[132],X,S\$	
30 N\$=CHR\$(0)	Le caractère nul.
40 FOR X=1 TO 131	
50 N\$=N\$ & CHR\$(0)	
60 NEXT X	N\$ contient maintenant 132 colonnes de points vierges.
70 INPUT "ENTREZ UNE CHAINE: ";S\$	
80 DELAY 0,8	Ces deux lignes rendent l'action de la ligne 100 moins visible.
90 CONTRAST 0	
100 DISP S\$	Prépare l'affichage pour la fonction <code>GDISP</code> .
110 D\$=GDISP\$	Lit la configuration de points entrée en ligne 70.
120 DISP	Efface l'affichage.
130 CONTRAST 9	Ré-active le contraste normal.
140 FOR X=1 TO 132	Cette boucle déroule le message à l'affichage,
150 GDISP N\$[X,132]&D\$	colonne par colonne.
160 NEXT X	
170 FOR X=1 TO 132	Cette boucle déroule le message hors de
180 GDISP D\$[X,132]	l'affichage par la gauche.
190 NEXT X	
200 DELAY .5,.125	Ré-active le délai standard.
210 DISP TAB(8);"Fin" @ END	

Lors de la première exécution de la boucle comprenant les lignes 140 à 160, l'ordre `DISP#` de la ligne 150 affiche les 132 colonnes de points vierges stockés dans `M#`. Lors de la seconde exécution, la ligne 150 affiche le contenu de `M#` mais remplace la colonne de gauche par la première colonne de points stockés dans `D#` (données lues par l'instruction de la ligne 110). Les exécutions suivantes de la boucle déroulent le contenu de `M#` hors de l'affichage et le remplace petit à petit par celui de `D#`. Le programme affiche donc, colonne par colonne, le message entré en ligne 70.

La boucle comprenant les lignes 170 à 190 fait donc disparaître `D#` comme la boucle des lignes 140 à 160 a fait disparaître `M#`.

Restriction de l'utilisation du HP-71 (LOCK)

Vous pouvez utiliser `LOCK` pour définir un mot de passe limitant l'accès à votre HP-71.

```
LOCK mot de passe
```

Le *mot de passe* peut être une expression quelconque donnant une chaîne de huit caractères ou moins. Ce paramètre ne peut pas être une chaîne sans guillemets. Toute personne voulant utiliser votre HP-71 devra frapper le mot de passe avant de pouvoir effectuer des opérations.

Vous pouvez exécuter `LOCK` à tout moment. Son exécution n'éteint pas le HP-71 et n'affecte pas son fonctionnement. Lors de la mise sous tension suivante, le HP-71 affiche le message `password?` et s'éteint automatiquement si l'utilisateur n'entre pas le mot de passe correct (sans guillemets).

Vous pouvez annuler le mot de passe en exécutant l'ordre `LOCK` suivi d'une chaîne vide: `LOCK ""`. La ré-initialisation de la mémoire annule aussi le mot de passe.

Exécution automatique d'instructions (STARTUP)

L'ordre `STARTUP` vous permet de définir une instruction ou un groupe d'instructions que le HP-71 exécute automatiquement lors de chaque mise sous tension.

```
STARTUP "chaîne d'instructions"
```

La *chaîne d'instructions* peut être une instruction ou un groupe d'instructions quelconques exécutables au clavier, reliées par `;` et le tout placé entre guillemets.

Lors de l'exécution de `STARTUP`, le HP-71 ne vérifie pas la syntaxe des instructions de la chaîne. Cette vérification n'a lieu que lors de la première mise sous tension suivante du HP-71.

Vous pouvez annuler cette chaîne d'instructions en exécutant `STARTUP` suivi d'une chaîne vide: `STARTUP ""`.

Contrôle de l'affichage (LC)

Vous pouvez contrôler à partir du clavier ou d'un programme:

- Les vitesses de déroulement horizontale et verticale.
- L'angle d'affichage donnant le meilleur contraste.
- la longueur de chaque ligne affichée, 96 caractères maximum.
- Le type de clavier: majuscules ou minuscules en position principale des touches.

Les vitesses de déroulement sont contrôlées par l'ordre `DELAY` (voir chapitre 1, page 26). L'angle d'affichage est contrôlé par l'ordre `CONTRAST` (voir chapitre 1, page 29). La longueur de ligne est contrôlée par l'ordre `WIDTH` (voir chapitre 13, page 232). Le type de clavier est décrit ci-après.

```
LC
LC ON
LC OFF
```

L'ordre `LC` change le type de clavier: si les majuscules sont en positions primaires, cet ordre affecte les minuscules aux positions primaires et vice-versa. Cet ordre n'affecte que les touches alphabétiques.

L'exécution de l'ordre `LC ON` active le clavier de minuscules. La pression d'une touche alphabétique affiche alors une minuscule et la pression de la même touche précédée de `[9]` affiche la même lettre en majuscule.

L'exécution de l'ordre `LC OFF` active le clavier de majuscules. La pression d'une touche alphabétique affiche alors une majuscule et la pression de la même touche précédée de `[9]` affiche la même lettre en minuscule.

Deuxième partie
Programmation du HP-71

Rédaction et exécution de programmes

Table des matières

Présentation	143
Entrée d'un nouveau programme	143
Création d'un nouveau fichier de programme (EDIT)	143
Utilisation du fichier de travail <code>workfile</code> (NAME)	144
Entrée d'une ligne de programme	145
Entrée de lignes en série (AUTO)	149
Exécution d'un programme	149
Exécution du fichier courant (RUN, GOSUB)	149
Exécution d'un programme quelconque (RUN, CALL, CHAIN)	150
Arrêt d'un programme	152
Arrêt à partir du clavier	153
Arrêt à partir d'un programme (PAUSE, STOP, END)	154
Relance de l'exécution (<code>F</code> <code>CONT</code>), <code>CONT</code>)	155
Modifications d'un programme	156
Affichage de lignes de programme (FETCH, LIST, GOTO)	156
Ajout de lignes	158
Modification de lignes existantes	158
Suppression de lignes (DELETE)	158
ReNumérotation des lignes (RENUMBER)	159
Utilisation des fichiers <code>BIN</code> et <code>LEX</code>	160
Programmes binaires	160
Fichiers d'extension langage	160
Transformation de fichiers (TRANSFORM)	160

Présentation

Les chapitres précédents ont utilisés des petits programmes pour illustrer certaines des caractéristiques du HP-71. Si vous avez entré en mémoire certains de ces programmes, vous avez déjà une petite expérience de la programmation de cet ordinateur. Le présent chapitre décrit la rédaction et l'exécution de programmes sur le HP-71:

- Création d'un fichier de programme.
- Entrée de lignes dans un fichier de programme.
- Exécution d'un programme en utilisant la touche `RUN` ou les ordres BASIC RUN, GOSUB, CALL et CHAIN.
- Modification d'un programme.
- Recherche des erreurs dans un programme.
- Interruption de l'exécution d'un programme.

Entrée d'un nouveau programme

L'entrée d'un nouveau programme dans le HP-71 se passe en deux étapes:

1. Création d'un nouveau fichier de programme.
2. Entrée des lignes de programme dans le fichier.

Vous pouvez aussi entrer un programme en mémoire à partir de cartes magnétiques en utilisant l'ordre `COPY` comme décrit en annexe C.

Création d'un nouveau fichier de programme (EDIT)

Avant d'entrer des lignes de programme dans le HP-71, placez l'ordinateur en mode BASIC (vous ne pouvez pas entrer de lignes de programme en mémoire lorsque le HP-71 est en mode CALCUL). Utilisez ensuite `EDIT` pour créer un nouveau fichier de programme BASIC dans lequel le HP-71 pourra stocker les lignes de programme entrées.

```
EDIT nom de fichier
```

Le paramètre spécifie le nom du fichier que vous créez. Ce fichier devient le fichier courant et est stocké dans la mémoire principale. S'il existe déjà un fichier portant ce nom, le HP-71 déclare ce dernier comme fichier courant (voir chapitre 6, «Opérations sur les fichiers», pour plus d'informations sur les noms de fichier).

Exemple, `EDIT RESULTAT` crée un fichier nommé `RESULTAT` et le déclare comme fichier courant. Le HP-71 stocke dans ce fichier toutes les lignes de programme entrées ensuite.

Lors de la création de ce fichier, il n'est pas nécessaire de spécifier sa taille. Le HP-71 ajuste automatiquement la taille des fichiers BASIC lors de l'entrée de nouvelles lignes. La taille des fichiers n'est limitée que par la quantité de mémoire vive disponible.

Vous pouvez spécifier dans quelle unité de mémoire vous désirez stocker le fichier créé en ajoutant le nom de l'unité après le nom du fichier tel que décrit au chapitre 6. Si vous omettez le nom d'unité, le HP-71 crée le fichier en mémoire principale.

Exemples:

<code>EDIT RADAR</code>	Crée un fichier BASIC en mémoire vive.
<code>EDIT NOYAU:PORT(0)</code>	Crée un fichier BASIC dans le logement 0.

Utilisation du fichier de travail `workfile` (`NAME`)

Au lieu de créer un nouveau fichier pour votre programme, vous pouvez utiliser le fichier de travail `workfile` comme fichier de brouillon lors de l'entrée des lignes de programme. Pour cela vous devez déclarer `workfile` comme fichier courant. Nous vous recommandons en outre de vérifier que le fichier `workfile` est vide (consultez le chapitre 6 pour plus d'informations sur le fichier `workfile`).

Pour déclarer `workfile` comme fichier courant, frappez `EDIT` sans nom de fichier.

Entrée/Résultat

<code>EDIT</code> <code>END LINE</code>	Le nom de fichier par défaut est <code>workfile</code> .
---	--

<code>workfile</code> <code>BASIC nn</code>	Affiche l'étiquette du fichier <code>workfile</code> . Si <code>nn</code> est nul, le fichier <code>workfile</code> est vide. <code>workfile</code> est maintenant le fichier courant.
---	--

Si `nn` n'est pas nul, le fichier `workfile` contient des lignes de programme. Vous devez alors soit stocker le contenu de `workfile` dans un autre fichier soit le supprimer.

Vous pouvez sauvegarder le contenu de `workfile` de deux façons:

1. En nommant `workfile`.
2. En copiant `workfile`.

Changement du nom de workfile. Vous pouvez changer le nom du fichier workfile en utilisant l'ordre NAME.

— syntaxe simplifiée —

```
NAME nouveau nom de fichier
```

L'exemple suivant illustre la sauvegarde du contenu de workfile en utilisant NAME. Une fois le changement de nom effectué, vous pouvez créer un nouveau fichier de travail et le déclarer comme fichier courant en utilisant l'ordre EDIT.

```
NAME PROPEL1 END LINE
```

Nomme le fichier de travail workfile.

```
EDIT END LINE
```

Crée un nouveau fichier de travail workfile et le déclare comme fichier courant.

Copie du fichier workfile. Pour copier le fichier workfile, vérifiez qu'il est déclaré comme fichier courant et exécutez COPY TO *fichier destination* sans spécifier de *fichier source* (l'ordre COPY est décrit au chapitre 6 dans la section «Copie de fichiers», page 112).

Après la copie de workfile, nous vous recommandons d'en effacer le contenu avant d'y entrer de nouvelles lignes. Vous pouvez, pour cela, utiliser l'ordre DELETE ALL:

```
EDIT END LINE
```

Déclare le fichier workfile comme fichier courant.

```
COPY TO FILE1 END LINE
```

Copie le contenu de workfile dans un fichier nommé FILE1 en mémoire principale.

```
DELETE ALL END LINE
```

Efface le contenu de workfile.

Entrée d'une ligne de programme

Format des lignes de programme. Une ligne de programme BASIC commence toujours par un *numéro de ligne* et se compose d'une ou plusieurs instructions BASIC. Les numéros de lignes sont des entiers entre 1 et 9999 qui définissent la position de la ligne dans le programme. Le HP-71 stocke les lignes de programme dans l'ordre de leurs numéros quel que soit l'ordre dans lequel vous les entrez.

Lignes multi-instructions (@). IL est souvent utile de pouvoir placer plusieurs instructions sur une même ligne de programme, en particulier dans les instructions IF...THEN...ELSE (voir chapitre 10, «*Branchement, boucle et exécution conditionnelle*»). Vous pouvez placer plusieurs instructions sur une même ligne en les *concaténant* avec le symbole @. Exemple:

```
10 A = B
20 B = C
```

peut être écrit sous la forme

```
10 A = B @ B = C
```

Les deux instructions étant sur la même ligne, vous économisez deux octets de mémoire. Le HP-71 exécute les instructions de gauche à droite.

Labels. Vous pouvez identifier chaque instruction d'un programme par un *label*. Vous pouvez utiliser ces labels dans les instructions de branchements (GOSUB et GOTO) pour transférer l'exécution à une instruction sur une ligne de programme quelconque. Exemple, dans la ligne de programme:

```
250 'INVERSE': X=A @ A=B @ B=X
```

le label INVERSE identifie l'instruction X=A de la même façon que le numéro de ligne 250 identifie la ligne de programme.

Les labels vous permettent d'identifier des sous-routines avec des mnémoniques significatives, vous aidant ainsi à vous souvenir du rôle de la sous-routine. Lors de la rédaction d'un programme, il vous sera parfois nécessaire d'inclure une instruction de branchement transférant l'exécution à une sous-routine que vous n'avez pas encore écrite. Dans la plupart des cas vous ne connaissez pas le numéro de la première ligne de la sous-routine; l'utilisation d'un label vous permet de compléter l'instruction de branchement dès maintenant. Exemple:

```
40 GOSUB 'RETEST'
```

provoque un branchement à l'instruction portant le label RETEST. On dit que cette instruction *fait référence* au label RETEST. L'utilisation de label au lieu de numéro de ligne vous permet de rédiger l'ensemble du programme principal sans avoir à vous inquiéter des numéros de premières lignes des sous-routines. Exemple:

```
530 'RETEST': IF A=B OR C=D THEN A=D
```

Lors de l'utilisation de labels, souvenez-vous des règles suivantes:

- Un label se compose de huit lettres ou chiffres maximum et doit commencer par une lettre. Lors de l'entrée d'une ligne, il n'est pas nécessaire de placer le label entre apostrophes mais le HP-71 ajoute toujours des apostrophes de chaque côté des labels.

```
110 'SORT12': FOR I=1 TO 100      SORT12 identifie l'instruction
                                FOR I=1 TO 100
```

- Lorsqu'un label *identifie* une instruction, vous devez placer un signe : après le dernier caractère du label. N'utilisez pas le signe : lorsque le label *référence* une instruction. Exemple:

```
90 'RETEST': IF A=B THEN B=D     RETEST identifie l'instruction.
40 GOTO 'RETEST'                RETEST référence l'instruction.
```

- Un label peut se trouver seul sur une ligne.

```
120 'VOILIER':                  'VOILIER': identifie une ligne.
```

- Vous pouvez placer un label dans une ligne multi-instruction après @.

```
350 IF A=B THEN B=C @           'COMPLEX1': identifie l'instruction
    'COMPLEX1': A=A/C           A=A/C.
25 N=SIN(X) @ 'QUADN':          'QUADN': et 'SIGNN': identifient deux ins-
    Q=RED(SIN(X),360) @ 'SIGNN': tructions différentes sur la même ligne.
    S=SIGN(SIN(X))
```

- Vous pouvez placer plusieurs labels sur une même ligne et plusieurs labels peuvent identifier la même instruction.

```
9990 'VERIF1': 'VERIF2':        'VERIF1': et 'VERIF2': identifient la
    IF X#0 THEN Y=1             même instruction. Cette situation apparaît
                                lorsque vous consolidez plusieurs sous-routines
                                mais ne voulez pas changer toutes les références
                                de label pour ces sous-routines.
```

- Si vous utilisez des labels identiques pour identifier différentes instructions, le HP-71 ne reconnaît que la première instruction. Vous ne pourrez jamais transférer l'exécution à des labels identiques après le premier.

```
10 'DEBUT': DESTROY A,B,C       ligne porte le label 'DEBUT':.
20 'DEBUT': DIM S(25)           Cette ligne contient aussi le label 'DEBUT':.
                                Néanmoins, ce label étant déjà utilisé en ligne 10,
                                vous ne pourrez jamais effectuer un transfert
                                d'exécution au label 'DEBUT': de la ligne 20.
```

Entrée d'une ligne. Pour entrer une ligne de programme, frappez un numéro de ligne suivi d'une ou plusieurs instructions. Le HP-71 stocke cette ligne dans le fichier de programme lorsque vous appuyez sur `END LINE` (si une erreur de syntaxe existe, le HP-71 ne stocke pas la ligne. Les erreurs de syntaxe sont décrites en page 163).

Le HP-71 interprète toute ligne affichée comme une ligne de programme si elle commence par un numéro de ligne. Exemple:

```
10A=B
```

interprétée comme une ligne de programme alors que

```
A=B
```

est exécutée immédiatement.

Vérification d'erreur. Lorsque vous appuyez sur `END LINE` pour entrer une ligne de programme en mémoire, le HP-71 vérifie que la ligne ne contient pas d'erreur de syntaxe: orthographe, paramètres incorrects et utilisation erronée d'un mot-clé.

Si le HP-71 ne trouve pas d'erreur de syntaxe:

- Il stocke la ligne dans le fichier de programme.
- Il déclare cette ligne comme ligne courante (consultez la section «Affichage de lignes de programme» en page 156).
- Il efface l'affichage.

Si le HP-71 trouve une erreur de syntaxe:

- Il ne stocke pas la ligne dans le fichier de programme.
- Il émet un bip.
- Il affiche un message d'erreur pendant la durée définie par l'ordre `DELAY`.
- Il affiche la ligne.
- Il place le curseur sur le caractère de la ligne où il a détecté l'erreur.

Si le HP-71 vous indique qu'il a trouvé une erreur, corrigez la ligne de programme et ré-entrez-la (voir «Recherche et corrections d'erreurs» page 165).

Entrée de lignes en série (AUTO)

Lorsque vous entrez de nouvelles lignes dans un fichier, la taille de ce dernier augmente de façon à accepter les nouvelles lignes. Si vous utilisez toute la mémoire de l'ordinateur et essayez d'entrer une nouvelle ligne, le HP-71 affiche un message vous indiquant qu'il n'y a pas assez de mémoire.

Lors de l'entrée d'un programme, l'ordre `AUTO` vous permet d'afficher automatiquement les numéros des lignes successives lors de leur frappe. Vous pouvez spécifier en option le numéro de la première ligne et l'incrément.

Exemples:

```
AUTO
```

Commence la numérotation automatique des lignes à la ligne 10 avec un incrément de 10.

```
AUTO 100,20
```

Commence la numérotation automatique des lignes à la ligne 100 avec un incrément de 20.

Exécution d'un programme

Exécution du fichier courant (RUN, COSUE)

Vous pouvez exécuter le fichier courant en appuyant simplement sur `RUN`. Cette touche lance l'exécution du fichier courant à partir de la ligne de plus petit numéro.

Vous pouvez lancer l'exécution du fichier courant en exécutant l'ordre `RUN` au clavier.

— syntaxe simplifiée —

```
RUN [numéro de ligne]
```

```
RUN [, label]
```

Le *numéro de ligne* ou *label* représente la ligne ou l'instruction où l'exécution doit commencer. Si vous omettez ce paramètre, le HP-71 commence l'exécution à la ligne de plus petit numéro (si vous spécifiez une ligne inexistante, le HP-71 commence l'exécution à la ligne de numéro supérieur. Si vous spécifiez un label inexistante, le HP-71 affiche un message d'erreur).

Exemples:

```
RUN 120
```

Exécute le fichier courant en commençant à la ligne 120.

```
RUN ,UNITE1
```

Exécute le fichier courant en commençant à l'instruction de label `UNITE1`.

RUN exécuté dans un programme lance l'exécution de ce programme à la ligne ou au label spécifié ou à la ligne de plus petit numéro par défaut. Lors de l'exécution de RUN, seules les variables et tableaux de l'environnement principal restent inchangés (l'environnement principal est décrit au chapitre 12, «Sous-programmes et fonctions définies par l'utilisateur»).

L'ordre GOSUB, utilisé au clavier, vous permet de lancer l'exécution du fichier courant à partir d'une ligne ou d'un label quelconque.

— syntaxe simplifiée —

```
GOSUB numéro de ligne
GOSUB label
```

Exemples:

```
GOSUB 240
```

Exécute le fichier courant en commençant à la ligne 240.

```
GOSUB "TERM"
```

Exécute le fichier courant en commençant au label TERM.

Si vous exécutez GOSUB au clavier, le programme s'arrête dès qu'il rencontre PAUSE, STOP, END ou RETURN. Les instructions éventuellement concaténées après GOSUB ne sont exécutées que si le programme est arrêté par RETURN (l'exécution de GOSUB dans un programme est décrit dans le chapitre 10, «Branchement, boucle et exécution conditionnelle»). L'ordre RETURN arrêtant un programme lancé par GOSUB, vous pouvez exécuter une sous-routine au clavier sans exécuter le programme principal complet.

Exécution d'un programme quelconque (RUN, CALL, CHAIN)

Vous pouvez exécuter un programme quelconque en mémoire (ou sur un support de stockage de masse) à partir du clavier ou à partir d'un autre programme. Il n'est pas nécessaire qu'un fichier soit le fichier courant pour exécuter le programme qu'il contient. Néanmoins, le HP-71 déclare le fichier de programme comme fichier courant juste avant de commencer l'exécution.

Exécution d'un programme. L'instruction RUN vous permet d'exécuter un programme quelconque en commençant au *numéro de ligne* ou au *label* spécifié.

— syntaxe simplifiée —

```
RUN nom de fichier [, numéro de ligne]
RUN nom de fichier [, label]
```

Exemples:

<code>RUN PROG1</code>	Exécute le fichier <code>PROG1</code> en commençant à la première ligne.
<code>RUN TEST3:PORT(1),3500</code>	Exécute le fichier <code>TEST3</code> du logement 1 en commençant à la ligne 3500.
<code>RUN PROG2,COMPTE</code>	Exécute le fichier <code>PROG2</code> en commençant au label <code>COMPTE</code> .

Appel d'un programme comme sous-programme. L'ordre `CALL` exécute un programme d'une façon similaire à `RUN`. Cependant, lors de l'exécution de `CALL`, le HP-71 sauvegarde l'environnement courant et crée un nouvel environnement pour le programme appelé. Le HP-71 traite le programme appelé comme un sous-programme (les sous-programmes sont décrits au chapitre 12).

```
CALL nom de fichier [: unité]
```

Exemples:

<code>CALL ORBITE</code>	Exécute le programme du fichier <code>ORBITE</code> .
<code>CALL APOGEE:PORT(0)</code>	Exécute le programme du fichier <code>APOGEE</code> dans le logement 0.
<code>CALL</code>	Exécute le fichier courant.

Bien que `CALL` soit conçu pour exécuter des sous-programmes, vous pouvez l'utiliser pour exécuter des programmes tel qu'indiqué ci-dessus à condition qu'il n'y ait pas en mémoire de sous-programme portant le même nom. Lorsque vous exécutez `CALL`, le HP-71 cherche d'abord un sous-programme portant ce nom. S'il ne trouve pas de sous-programme, le HP-71 cherche un programme portant le nom spécifié. Si vous essayez d'exécuter avec `CALL` un programme portant le même nom qu'un sous-programme en mémoire, le HP-71 exécute le sous-programme plutôt que le programme.

Enchaînement de programmes. L'ordre `CHAIN` charge un programme en mémoire principale à partir de cartes magnétiques, d'une unité de mémoire de masse ou de la mémoire et l'exécute. Avant de copier le nouveau fichier en mémoire principale, le HP-71 supprime le fichier courant (celui dans lequel se trouve l'instruction `CHAIN` ayant provoqué l'enchaînement).

```
CHAIN nom de fichier [: unité]
```

`CHAIN` s'avère particulièrement utile lorsque vous voulez exécuter un programme trop grand pour la mémoire de votre HP-71. Vous pouvez alors diviser le programme en plusieurs petits programmes et enchaîner leurs exécutions. Chaque petit programme doit se terminer par une instruction `CHAIN` qui charge et exécute le segment suivant.

Exemple: supposons que vous ayez un grand programme et que la mémoire de votre HP-71 ne permette d'en exécuter qu'un tiers à la fois. Vous pouvez simplement découper le programme en trois segments et stocker séparément ces derniers en mémoire vive, sur des cartes magnétiques ou sur une unité de mémoire de masse.



Comme illustré ci-dessus, chaque segment est stocké comme un fichier séparé. Si vous nommez ces fichiers PROG1, PROG2 et PROG3, la dernière instruction de PROG1 est CHAIN PROG2 et la dernière instruction de PROG2 est CHAIN PROG3. Lorsque PROG1 se termine, le HP-71 le supprime de la mémoire principale et le remplace par PROG2. A la fin de PROG2, le HP-71 supprime ce fichier et le remplace par PROG3.

Bien que CHAIN soit conçu principalement pour enchaîner des fichiers stockés sur cartes magnétiques ou sur des unités de mémoire de masse, vous pouvez l'utiliser pour exécuter des programmes stockés en mémoire principale (pour plus d'informations sur l'utilisation de CHAIN avec des fichiers sur cartes magnétiques, consultez l'annexe C).

Arrêt de programme

L'exécution d'un programme s'arrête automatiquement à la fin du programme mais peut aussi s'arrêter pour d'autres raisons: erreur pendant l'exécution par exemple. Dans ce cas, le HP-71 arrête l'exécution et affiche un message d'erreur.

Vous pouvez aussi arrêter vous-même l'exécution avant la fin du programme soit à partir du clavier soit par l'intermédiaire d'une instruction dans le programme. Ces arrêts vous permettent par exemple d'afficher des valeurs ou de corriger des erreurs.

Un programme arrêté peut assumer un des deux états suivants:

- *Interrompu* (ou *suspendu*). Le témoin **SUSP** est allumé, indiquant que vous pouvez relancer le programme à l'endroit de l'interruption. Toutes les informations de contrôle restent intactes.
- *Terminé*. Le témoin **SUSP** est éteint. Toutes les informations de contrôle du programme sont effacées. Vous ne pouvez pas relancer le programme.

Un programme est interrompu lorsque vous appuyez sur **[ATTN]**, lorsque vous exécutez PAUSE ou lorsqu'une erreur a lieu.

Arrêt à partir du clavier

Interruption de programme. La touche **[ATTN]** vous permet d'interrompre un programme de telle façon qu'il conserve l'état suspendu. Le HP-71 affiche alors le témoin **SUSP**, indiquant que vous pouvez relancer le programme à l'endroit de l'interruption (*l'instruction où l'exécution est interrompue est appelée instruction d'interruption*).

Le HP-71 conserve l'environnement existant au moment de l'interruption (les environnements sont décrits au chapitre 12, «*Sous-programmes et fonctions définies par l'utilisateur*»). Lorsque l'exécution d'un programme est interrompue, vous pouvez effectuer les opérations suivantes sur le HP-71 sans affecter l'état interrompu:

- Affichage et modification de variables.
- Calculs au clavier en mode BASIC ou CALCUL.
- Affichage du contenu du fichier courant.
- Copie de fichiers.
- Obtention de catalogues.
- Mises sous et hors tension du HP-71.

En général, toutes les opérations qui ne modifient pas le fichier courant ou ne déclarent pas un autre fichier comme fichier courant, n'affectent pas l'état interrompu du programme. Les opérations qui affectent l'état interrompu d'un programme sont décrites ci-après sous le titre «*Fin de programme*».

Lorsque vous voulez relancer l'exécution d'un programme interrompu, vous pouvez appuyer sur **[CONT]** ou exécuter CONT. L'exécution reprend à *l'instruction d'interruption*, instruction suivant la dernière instruction exécutée. Vous pouvez afficher l'instruction d'interruption en exécutant FETCH au clavier.

Dans certaines situations, un programme ne s'arrête pas lorsque vous appuyez sur **[ATTN]**. Si cela se produit, appuyez sur **[ON]**/**[/]** simultanément et choisissez le niveau 1 (ce processus est décrit en annexe A page 273). Lorsque vous interrompez l'ordinateur avec **[ON]**/**[/]**, le HP-71 termine le programme et peut effectuer une ré-initialisation de la mémoire. Nous vous recommandons donc de n'utiliser cette séquence que si vous ne pouvez pas reprendre le contrôle du HP-71.

Fin de programme. Les instructions et opérations suivantes terminent un programme et annulent l'état interrompu:

- EDIT.
- END, END ALL, STOP.*
- MERGE (sur un fichier BASIC).
- PURGE (sur le fichier courant).
- FREE PORT, CLAIM PORT.
- TRANSFORM (du fichier courant).
- RUN, CHAIN, GOSUB.
- Modification d'une ligne du programme.

Une fois un programme terminé, vous ne pouvez pas le remettre dans l'état interrompu précédent.

Arrêt à partir d'un programme (PAUSE, WAIT, STOP, END)

Interruption de programme. L'exécution de PAUSE permet à un programme de s'interrompre lui-même.

```
PAUSE
```

Lorsqu'un programme exécute cet ordre, il s'interrompt comme si vous aviez appuyé sur `[ATTN]`. La *ligne courante* est celle contenant l'instruction suivant PAUSE. Si vous voulez relancer l'exécution, il vous suffit d'appuyer sur `[f][CONT]` ou d'exécuter CONT (décrit ci-après sous le titre «*Relance de l'exécution*»).

Pour permettre à un utilisateur de consulter des résultats intermédiaires, vous pouvez utiliser WAIT.

```
WAIT secondes
```

L'ordre WAIT ordonne au programme d'attendre sans rien faire pendant le nombre de *secondes* spécifié. Toute information présente à l'affichage lors de l'exécution de l'ordre WAIT reste affichée pendant l'attente. WAIT n'interrompt pas le programme.

* Si le programme a été interrompu pendant l'exécution d'un sous-programme, l'exécution de END ou STOP termine le sous-programme seulement, le programme principal reste en état interrompu. END ALL termine tous les niveaux de sous-programme et le programme principal (pour plus d'informations sur les sous-programmes, consultez le chapitre 12 «*Sous-programmes et fonctions définies par l'utilisateur*»).

Fin de programme. Un programme termine son exécution s'il rencontre un ordre `STOP` ou `END`.*

```
STOP
```

```
END
```

Ces deux ordres terminent l'exécution d'un programme et en annulent l'environnement. Bien que ces ordres soient généralement situés à la fin d'un programme, ils peuvent être placés sur une ligne quelconque du programme.

Relance de l'exécution d'un programme (`f` `CONT`, `CONT`)

Lorsque le témoin **SUSP** est allumé, le programme dans le fichier courant est simplement interrompu. Vous pouvez en relancer l'exécution en appuyant sur `f` `CONT` ou en exécutant `CONT`. L'exécution reprend à l'instruction d'interruption.

Pression de `f` `CONT`. La pression de `f` `CONT` relance l'exécution d'un programme interrompu à l'instruction d'interruption si le témoin **SUSP** est allumé. Si le témoin **SUSP** est éteint, la pression de `f` `CONT` commence l'exécution du fichier courant à la ligne de plus petit numéro (identique à la pression de `RUN`).

Nota: Vous ne pouvez pas relancer l'exécution en appuyant sur `RUN` ou en exécutant `RUN`. `RUN` efface les informations de contrôle de programme et recommence le programme.

Exécution de `CONT`. L'ordre `CONT` vous permet de spécifier le numéro de ligne ou le label d'instruction où vous voulez relancer l'exécution.

```
CONT
```

Exemples:

```
CONT 100
```

Relance l'exécution à la ligne 100.

```
CONT "PHASE1"
```

Relance l'exécution à l'instruction de label `PHASE1`.

L'ordre `CONT` n'est pas programmable. Vous ne pouvez l'utiliser qu'au clavier.

* Si un programme lance l'exécution d'un sous-programme, l'exécution de `STOP` ou `END` termine le sous-programme uniquement. Pour terminer un programme pendant l'exécution d'un sous-programme, vous devez exécuter `END ALL` (pour plus d'informations sur les sous-programmes, consultez le chapitre 12, «Sous-programmes et fonction définies par l'utilisateur»).

Si vous ne spécifiez pas de numéro de ligne ou de label d'instruction, l'exécution reprend à l'instruction d'interruption.

Si vous exécutez `CONT` pour un programme terminé (et non seulement interrompu), le HP-71 lance l'exécution du programme courant (identique à l'exécution de `RUN`).

Modification de programme

Les modifications de programme se composent généralement d'une combinaison des opérations suivantes:

- Affichage de lignes de programme.
- Ajout de lignes.
- Modification de lignes existantes.
- Suppression de lignes.
- Renumérotation de lignes.

Avant de modifier un fichier, vérifiez que le HP-71 est en mode BASIC et que le fichier à modifier est le fichier courant.

Pour modifier un programme qui n'est pas dans le fichier courant, utilisez:

```
EDIT [nom de fichier]
```

Si vous ne spécifiez pas le *nom de fichier*, le HP-71 déclare le fichier `workfile` comme fichier courant.

Affichage de lignes de programme (FETCH, LIST, PLIST, GOTO)

Déroulement de programme. Vous pouvez utiliser les touches `▲`, `▼`, `g▲` et `g▼` pour afficher successivement les différentes lignes d'un programme BASIC.

Le HP-71 n'affichant qu'une seule ligne à la fois, il définit cette ligne comme ligne courante. Vous pouvez afficher la ligne courante en exécutant `FETCH` (comme décrit ci-après).

Pour afficher la ligne précédente, appuyez sur `▲`. Pour afficher la ligne suivante, appuyez sur `▼`. La nouvelle ligne affichée devient la ligne courante. Ces touches vous permettent de dérouler le fichier ligne par ligne.

Si vous maintenez la touche `▲` ou `▼` enfoncée, le HP-71 déroule successivement toutes les lignes du fichier dans le sens indiqué.

La séquence `g▲` permet d'afficher la première ligne d'un fichier et `g▼`, la dernière.

Affichage d'une ligne. L'ordre `FETCH` vous permet d'afficher une ligne quelconque d'un programme.

```
FETCH [numéro de ligne]
FETCH [label]
```

Vous pouvez spécifier la ligne à afficher soit par son numéro soit par le label d'une instruction qui s'y trouve. Si vous omettez le paramètre, le HP-71 affiche la ligne courante.

Exemples:

<code>FETCH</code>	Affiche la ligne courante.
<code>FETCH 100</code>	Affiche la ligne 100.
<code>FETCH GRAPH</code>	Affiche la ligne contenant le label <code>GRAPH</code> .
<code>FETCH A#</code>	Affiche la ligne contenant le label indiqué par la variable <code>A#</code> .

Listage de fichier. Vous pouvez lister le contenu d'un fichier en utilisant les ordres `LIST` et `PLIST`. Si le HP-71 est connecté à une imprimante, `PLIST` liste le programme spécifié sur l'imprimante, sinon il liste le programme sur l'affichage du HP-71*.

— syntaxe simplifiée —

```
LIST [numéro de première ligne [, numéro de dernière ligne]
LIST [nom de fichier] [, numéro de première ligne [, numéro de dernière ligne]]
```

— syntaxe simplifiée —

```
LIST [numéro de première ligne [, numéro de dernière ligne]
PLIST [nom de fichier] [, numéro de première ligne [, numéro de dernière ligne]]
```

Les ordres `LIST` et `PLIST` listent les lignes de programme dans l'ordre croissant de leurs numéros.

L'exécution de `LIST` sans paramètre liste le fichier courant de la première à la dernière ligne. Si le HP-71 ne trouve pas le fichier spécifié, il affiche le message d'erreur:

```
ERR: File Not Found
```

* Vous devez utiliser l'interface HP-IL HP 82401A pour connecter une imprimante au HP-71.

Changement de ligne courante. L'ordre GOTO exécuté au clavier vous permet de définir une ligne quelconque comme ligne courante sans l'afficher.

```
GOTO numéro de ligne
GOTO label
```

GOTO doit être suivi d'un *numéro de ligne* ou d'un *label*.

Exemples:

```
GOTO 100
```

Déclare la ligne 100 comme ligne courante.

```
GOTO VALENCE
```

Déclare la ligne contenant le label VALENCE comme ligne courante.

Ajout de lignes

Vous pouvez ajouter une ligne à un programme en frappant un numéro de ligne suivi d'une ou plusieurs instructions et de `END LINE`. De même que lors de l'entrée du programme, le HP-71 place la ligne dans le programme selon son numéro. Si ce numéro est déjà utilisé par une autre ligne, la nouvelle ligne remplace l'ancienne. Un programme ne peut pas contenir deux lignes de même numéro.

Modification de lignes existantes

Pour modifier une ligne de programme existante, vous devez tout d'abord l'afficher (FETCH, `▼`, `▲`, `⏎` ou `⏏`) comme décrit ci-avant). Vous pouvez alors effectuer tous les changements désirés puis appuyer sur `END LINE` pour la stocker dans le programme (la simple pression de `▼`, `▲`, `⏎` ou `⏏` après avoir modifié une ligne ne stocke pas la nouvelle ligne dans le programme).

Suppression de lignes (DELETE)

L'ordre DELETE vous permet de supprimer une ou plusieurs lignes d'un programme:

- Une seule ligne:

```
DELETE numéro de ligne
```

- Un bloc de lignes:

```
DELETE numéro de première ligne, numéro de dernière ligne
```

- Toutes les lignes d'un fichier:

```
DELETE ALL
```

Si vous exécutez `DELETE ALL`, le fichier reste en mémoire mais il est vide. Si vous voulez supprimer le fichier courant de la mémoire, exécutez plutôt `PURGE`.

Vous pouvez aussi supprimer une ligne d'un programme en frappant son numéro suivi de `END LINE`. `50 END LINE`, par exemple, supprime la ligne 50 du fichier courant.

Renumérotation des lignes (`RENUMBER`)

Bien souvent, après plusieurs ajouts, suppressions ou modifications de lignes de programme, les numéros de lignes ne vous permettent plus d'insérer une ligne supplémentaire. Vous ne pouvez pas ajouter de ligne entre les lignes 10 et 11 d'un programme par exemple. Pour pallier cela, vous devez changer les numéros d'une ou plusieurs lignes. Cependant, lorsque vous changez un numéro de ligne, vous devez veiller à changer en outre toutes les références à cette ligne (telles que les ordres `GOTO`, `GOSUB` et `PRINT USING`). Si vous changez de nombreux numéros de ligne, il est facile d'oublier une référence et d'introduire ainsi une erreur dans le programme.

L'ordre `RENUMBER` du HP-71 peut effectuer tous ces changements pour vous sans introduire d'erreurs. Cet ordre renumérote le fichier courant en utilisant les paramètres spécifiés.

```
RENUMBER [nouvelle première ligne [, incrément [, ancienne première ligne [, ancienne dernière ligne]]]]
```

Les numéros d'anciennes lignes définissent le bloc de lignes que vous voulez renuméroter. *L'incrément* spécifie le nombre de numéros entre deux nouvelles lignes successives et le premier paramètre spécifie la position du bloc renuméroté dans le programme.

Exemples:

```
RENUMBER
```

Renumérote le fichier courant de façon que la première ligne ait le numéro 10 et que l'incrément soit 10 (la deuxième ligne aura pour numéro 20). Ces valeurs de paramètres sont les valeurs par défaut.

```
RENUMBER 100,10,200,300
```

Renumérote les lignes 200 à 300 du fichier courant de façon que la ligne 200 devienne la ligne 100 et que l'incrément soit 10.

Utilisation des fichiers BIN et LEX

Le HP-71 possède un grand nombre d'ordres, de fonctions et d'opérateurs. Vous pouvez néanmoins accroître ses performances en utilisant les fichiers BIN et LEX: les programmes BIN sont plus rapides que les programmes BASIC et les fichiers LEX ajoutent des mots-clés au langage du HP-71.

Programmes binaires

Les programmes binaires sont des fichiers de programme codés d'une façon particulière et peuvent être exécutés comme des programmes BASIC. Ils se présentent généralement sous la forme de fichiers sur cartes magnétiques ou unité de stockage de masse ou peuvent se trouver dans une ROM enfichable.

Vous pouvez exécuter un fichier BIN de la même façon que vous exécuteriez un fichier BASIC: vous pouvez utiliser les ordres RUN, CHAIN ou CALL. Les fichiers BIN n'ont pas de numéro de ligne ou de label, vous ne pouvez donc pas les exécuter avec GOSUB. Le type de codage des fichiers BIN ne vous permet pas de les modifier. Vous ne pouvez que les exécuter.

Fichiers d'extension de langage

Les fichiers d'extension de langage (LEX) sont des fichiers binaires particuliers qui ajoutent des mots-clés BASIC au HP-71. Ils se trouvent généralement dans les logiciels d'applications et les modules ou extensions enfichables. Vous ne pouvez pas exécuter ou modifier un fichier LEX. Il n'est pas nécessaire qu'un fichier LEX soit le fichier courant pour que vous puissiez l'utiliser. Dès qu'un fichier LEX est dans la mémoire (RAM ou ROM) du HP-71 vous pouvez utiliser les mots-clés qu'il contient.

Vous pouvez utiliser les mots-clés BASIC définis dans un fichier LEX de la même façon que vous utiliseriez les autres mots-clés BASIC. La documentation fournie avec un fichier LEX présente la syntaxe et l'utilisation des mots-clés qu'il contient. Lorsque vous exécutez un programme BASIC contenant des mots-clés de fichier LEX, ce dernier doit être présent en mémoire.

Transformation de fichiers (TRANSFORM)

L'ordre TRANSFORM vous permet de transformer un fichier de programme BASIC en un fichier TEXT de façon que vous puissiez le transférer sur un autre ordinateur Hewlett-Packard.

— syntaxe simplifiée

```
TRANSFORM [[nom de fichier] ;unité] INTO type de fichier [nom de fichier [:unité]]
```

L'ordre TRANSFORM peut aussi transformer un fichier TEXT en un fichier de programme BASIC. Les fichiers TEXT résultant d'une transformation comportent un enregistrement par ligne de programme (l'ordre TRANSFORM est décrit plus en détails dans le *Manuel de référence* du HP-71).

Exemples:

```
TRANSFORM PROG1 INTO TEXT
  TPROG1:PORT(0)
```

Transforme le fichier de programme BASIC PROG1 en un fichier TEXT nommé TPROG1 dans le logement 0.

```
TRANSFORM TPROG1:PORT(0)
  INTO BASIC
```

Transforme le fichier TEXT TPROG1 du logement 0 en un fichier BASIC.

Cet ordre est particulièrement utile pour exécuter sur le HP-71 des programmes du HP-75.

Les fichiers TEXT du HP-71 utilisent le format de transfert logique Hewlett-Packard, type 1 (LIF1), utilisé par plusieurs ordinateurs HP et permettant de transférer des informations entre différents ordinateurs. Vous pouvez charger dans la mémoire du HP-71 des fichiers HP-75 de type LIF1 en utilisant le Lecteur de cartes magnétiques optionnel HP 82400A (le fonctionnement du lecteur de cartes est décrit en annexe C). Vous pouvez transformer un fichier BASIC en un fichier TEXT (LIF1) sur un ordinateur, le stocker sur cartes magnétiques puis le charger dans la mémoire d'un autre ordinateur où vous pouvez le retransformer en un fichier BASIC.

Exemple: Pour transformer un fichier de programme BASIC du HP-75 en un fichier LIF1 et le transférer sur le HP-71:

Sur le HP-75:

```
transform 'prog1' into lif1

copy 'prog1' to card
```

Transforme un fichier BASIC du HP-75 en un fichier LIF1.

Copie le fichier transformé sur une carte magnétique en utilisant le lecteur de cartes incorporé au HP-75.

Sur le HP-71:

```
COPY CARD TO PROG1

TRANSFORM PROG1 INTO BASIC
```

Copie le contenu du fichier sur carte dans le fichier TEXT PROG1.

Transforme PROG1 en un fichier BASIC.

Nota: Vous ne pouvez transformer un fichier TEXT en un fichier BASIC que si chaque enregistrement (ligne) commence par un numéro de ligne valide. Le HP-75 accepte le numéro de ligne 0 mais le HP-71 ne l'accepte pas. Si le HP-71 essaie de transformer un fichier contenant une ligne de numéro 0, il affiche une erreur et ne termine pas la transformation. Nous vous recommandons donc de vérifier que les fichiers transférés ne contiennent pas de ligne ayant le numéro 0.

Lorsque vous transformez un fichier TEXT écrit sur une carte par un HP-75, le HP-71 transforme en remarques toutes les lignes qu'il ne peut pas interpréter. C'est-à-dire que le HP-71 insère les caractères ! ? après le numéro de toute ligne qu'il ne peut pas interpréter. Vous devrez ensuite ré-écrire la ligne de façon qu'elle soit compatible avec le BASIC du HP-71.

Conditions d'erreur

Table des matières

Présentation	162
Types d'erreurs	163
Messages d'erreurs	163
Messages des erreurs de syntaxe	163
Messages des erreurs d'exécution (<code>(g)ERRM</code>)	164
Opérations de traitement d'erreurs	165
Pistage de l'exécution (<code>TRACE FLOW</code>)	166
Pistage des affectations de variables (<code>TRACE VARS</code>)	167
Annulation des opérations de pistage (<code>TRACE OFF</code>)	168
Exécution pas-à-pas (<code>(f)SST</code>)	168
Contrôle des erreurs par programme	171
Branchement pour une erreur (<code>ON ERROR,OFF ERROR</code>)	172
Détermination du numéro d'un message d'erreur (<code>ERRH</code>)	173
Rappel d'un message d'erreur (<code>(g)ERRM</code> , <code>ERRM#</code>)	175
Localisation d'une erreur (<code>ERRL</code>)	175
Messages d'avertissement	175
Exceptions mathématiques	176
Exceptions comme erreurs	176
Exceptions comme avertissements	177

Présentation

Ce chapitre traite des sujets suivants:

- Types d'erreurs.
- Notification des erreurs.
- Réponse aux messages d'erreur et d'avertissement.
- Localisation et correction des erreurs.
- Traitement des erreurs par programme.

Types d'erreurs

Lors de la rédaction d'un programme, de l'exécution d'opérations au clavier ou de l'exécution d'un programme, le HP-71 peut afficher des messages d'erreur ou d'avertissement. Un *message d'erreur* indique qu'une opération ne peut pas être effectuée si vous ne corrigez pas l'erreur. Un *message d'avertissement* indique que le HP-71 a utilisé une valeur par défaut comme résultat d'une opération ou qu'une certaine condition demande votre attention.

Vous pouvez rencontrer trois type d'erreurs sur le HP-71:

- Erreurs de syntaxe.
- Erreurs d'exécution.
- Erreurs de logique.

Une erreur de syntaxe est une erreur dans la construction d'une instruction, telle qu'une faute d'orthographe ou l'utilisation d'un paramètre incorrect. Le HP-71 vérifie la syntaxe des instructions dès leur entrée au clavier.

Une erreur d'exécution ne peut être détectée que lors de l'exécution de l'instruction. Ces erreurs apparaissent lorsque vous utilisez un argument non valide pour une fonction ou lorsque vous spécifiez un branchement à une ligne inexistante.

Une erreur de logique résulte d'une faute de conception du programme. Ce type d'erreur apparaît généralement lorsque le programme ne donne pas les résultats attendus. Le HP-71 ne détecte pas les erreurs de logique mais possède néanmoins des fonctions qui vous permettent de pister ces erreurs.

Messages d'erreurs

Le HP-71 affiche un message d'erreur et interrompt l'exécution lorsqu'il ne peut pas effectuer correctement une opération. Un message d'erreur indique la nature de l'erreur et, dans le cas d'une erreur dans un programme, la ligne sur laquelle le HP-71 a détecté l'erreur.

Messages des erreurs de syntaxe

Lorsque le HP-71 détecte une erreur de syntaxe:

- Il refuse la ligne entrée.
- Il affecte le numéro de l'erreur à `ERRN` (décrit page 173).
- Il émet un bip et affiche temporairement le message (la durée de l'affichage dépend de l'ordre `DELAY`).
- Il affiche la ligne frappée.
- Il place le curseur sur le caractère où il a détecté l'erreur.

Un message d'erreur de syntaxe se présente de la façon suivante:

```
ERR: message
```

Exemple:

```
ERR:Invalid Expr
```

Indique que vous avez frappé une expression de façon incorrecte.

Si le message affiché ne vous suffit pas pour déterminer la cause de l'erreur, consultez le chapitre «Messages d'erreur, d'avertissement et de système» dans *Manuel de référence* du HP-71. Ce chapitre liste les causes principales des erreurs affichées par le HP-71.

Messages des erreurs d'exécution (ERRM)

Lorsqu'une erreur d'exécution a lieu:

- Le HP-71 arrête l'exécution (si l'erreur a eu lieu dans un programme en cours d'exécution, le HP-71 interrompt l'exécution).
- Le HP-71 affecte à ERRL et à ERRN les numéros de ligne et de message respectivement (voir section «Contrôle des erreurs par programme» en page 171).
- Le HP-71 émet un bip et affiche le message.

Un message d'erreur d'exécution se présente de la façon suivante:

```
ERR: message
```

Pour une instruction exécutée au clavier.

```
ERR Ln: message
```

Pour instruction exécutée dans un programme.

Le mot ERR indique qu'il s'agit d'un message d'erreur, Ln indique le numéro de ligne, n, où le HP-71 a détecté l'erreur et message décrit la cause de l'erreur.

Exemple:

```
ERR L30:String Ovfl
```

Indique qu'une instruction de la ligne 30 a essayé d'affecter une chaîne à une variable trop petite.

Vous pouvez afficher le dernier message d'erreur ou d'avertissement en appuyant sur ERRM ou en exécutant ERRM#. Exemple:

Entrée/Résultat

```
DIM A#[5] [END LINE]
```

```
A#="TROP GRAND" [END LINE]
```

```
ERR:String Ovfl
```

```
[9] [ERRM] (maintenu)
```

```
String Ovfl
```

```
ERRM# [END LINE]
```

```
String Ovfl
```

Dimensionne la variable alphanumérique A# à 5 caractères.

Essaie d'affecter une chaîne de 10 caractères à la variable A#.

Le HP-71 émet un bip et affiche un message indiquant que la chaîne affectée à la variable est trop grande.

Affiche le message le plus récent.

La fonction ERRM# donne le message le plus récent.

Opérations de traitement d'erreur

Les messages d'erreur sont très concis et généralement faciles à interpréter. Si vous désirez une explication en Français ou plus d'informations sur une erreur, consultez le chapitre «*Messages d'erreur, d'avertissement et de système*» dans le *Manuel de référence* du HP-71. Ce chapitre contient une liste de tous les messages d'erreur du HP-71 et de leurs causes les plus courantes. Vous pouvez en outre consulter le dictionnaire des mots-clés dans le *Manuel de référence* pour plus d'informations sur la syntaxe et l'utilisation des mots-clés du HP-71.

Si vous ne pouvez pas déterminer la cause d'une erreur à l'aide du manuel de référence, vous pouvez utiliser TRACE et [SST] pour pister l'exécution du programme.

Pistage de l'exécution (TRACE FLOW)

L'ordre TRACE FLOW vous permet de pister les branchements d'un programme et facilite ainsi la localisation des erreurs. Pour pister les branchements d'un programme, exécutez TRACE FLOW au clavier puis lancez le programme. Cet ordre peut aussi être exécuté dans un programme mais ne peut pas l'être au clavier pendant l'exécution d'un programme.

```
TRACE FLOW
```

Lorsque vous pistez l'exécution d'un programme, le HP-71 affiche un message indiquant chaque branchement effectué (consultez le chapitre suivant «*Branchement, boucle et exécution conditionnelle*» pour plus d'informations sur les branchements).

Ce message est de la forme:

```
Trace line numéro de ligne TO numéro de ligne
```

Si l'exécution d'un programme suit exactement l'ordre des lignes, le HP-71 n'affiche aucun message. Mais lorsqu'un branchement a lieu (y compris un appel de routine), le HP-71 affiche le numéro de la ligne provoquant le branchement et celui de la ligne où l'exécution est transférée.

Exemple:

```
Trace line 100 to 40
```

Un branchement à une routine apparaît de la façon suivante:

```
Trace line numéro de ligne CALL nom de routine
```

et un retour de routine apparaît comme suit:

```
Trace line numéro de ligne ENDSUB
```

Exemple: La ligne 100 d'un programme appelle la routine TEST1. La routine commence en ligne 400 et se termine en ligne 450.

```
100 CALL TEST1
```

```
400 SUB TEST1
```

```
450 END SUB
```

Lors de l'exécution de ces lignes, le HP-71 affiche:

```
Trace line 100 CALL T1
```

Branchement à la routine T1.

```
Trace line 450 ENDSUB
```

Retour au programme appelant.

Pistage des affectations de variables (TRACE VARS)

L'ordre TRACE VARS vous permet de pister les changements de valeur des variables d'un programme en cours d'exécution. Cette ordre peut être actif en même temps que TRACE FLOW.

```
TRACE VARS
```

Lorsque le programme affecte une valeur à une variable, le HP-71 affiche un message indiquant le numéro de la ligne où l'affectation a eu lieu et:

- Le nom et la valeur affectée pour une variable numérique simple.

```
Trace line 10 B=2
```

- Le nom pour une variable alphanumérique.

```
Trace line 20 S$
```

- Le nom, le ou les indices et la valeur affectée pour un élément de tableau.

```
Trace line 30 C(2,3)=45
```

- Le nom et le ou les indices pour un élément de tableau alphanumérique.

```
Trace line 40 A$(3)[1,10]
```

TRACE VARS vous permet de trouver l'endroit où les affectations de variable sont erronées.

Annulation des opérations de pistage (TRACE OFF)

Vous pouvez annuler les opérations de pistage en exécutant TRACE OFF.

```
TRACE OFF
```

Cet ordre annule toute condition TRACE VARS et TRACE FLOW active et peut être exécuté au clavier ou dans un programme. Il ne peut pas être exécuté au clavier pendant l'exécution d'un programme.

Exécution pas à pas (f SST)

La fonction SST vous permet d'exécuter un programme pas à pas et de voir le résultat de chaque opération. Vous pouvez exécuter chaque ligne d'un programme pour déterminer où et pourquoi une erreur d'exécution apparaît.

La fonction SST a les résultats suivants:

- Elle affiche l'instruction suivante à exécuter lorsque la touche SST est enfoncée.
- Elle exécute l'instruction lorsque vous relâchez SST.

Exemple: Entrez un programme qui affiche une par une les lettres de l'alphabet puis exécutez ce programme pas à pas (avant d'utiliser SST, vérifiez que le programme se trouve dans le fichier courant et que le HP-71 est en mode BASIC). L'exécution pas à pas vous permet de vérifier que les résultats des différentes lignes sont corrects.

Entrée/Résultat

```
10 FOR I=65 TO 90 END LINE
```

Début de la boucle.

```
20 CHR$(I) END LINE
```

Affiche un caractère.

```
30 NEXT I END LINE
```

Fin de la boucle.

f SST (maintenu)

```
10 FOR I=65 TO 90
```

Affiche la première ligne tant que vous maintenez SST enfoncée (il n'est pas nécessaire de maintenir f enfoncée).

(relâchez la touche)

```
> █ SUSP
```

Exécute la ligne.

Affiche le curseur et le témoin **SUSP**.

f **SST** (maintenu)

```
20 DISP CHR#(I) SUSP
```

Affiche une ligne.

(relâchez la touche)

```
  A SUSP
```

Exécute la ligne.

Affiche le caractère de code décimal 65.

I **END LINE**

Quelle est la valeur du compteur d'incrément?
(vous pouvez effectuer des opérations au clavier pendant l'exécution pas à pas d'un programme.)

```
 65 SUSP
```

I égale 65 (code décimal de A).

f **SST**

```
30 NEXT I SUSP
```

Affiche et exécute la ligne suivante.

f **SST** (maintenu)

```
20 DISP CHR#(I) SUSP
```

Du fait de la boucle, l'exécution retourne à l'instruction 20.

(relâchez la touche)

```
  B SUSP
```

Affiche le caractère suivant.

I

66	SUSP
----	------

Valeur du compteur d'incrément?

Code décimal du caractère E.

Relance l'exécution normale du programme.

Vous pouvez continuer l'exécution pas à pas jusqu'à la fin du programme ou bien appuyer sur (voir ci-dessus) pour lancer l'exécution normale de la fin du programme.

Lorsque vous exécutez un programme pas à pas, le HP-71 affiche et exécute une instruction à la fois. Si vous exécutez des lignes multi-instructions avec , le HP-71 affiche et exécute une par une chaque instruction de la ligne avec les symboles @.

Exemple: Changez la ligne 20 du programme précédent de façon à afficher le code décimal du caractère, son équivalent en majuscule puis en minuscule. Exécutez ensuite le programme pas à pas.

Entrée/Résultat

20 I @ CHR\$(I) @ CHR\$(I+32)

Change la ligne 20 (exemple précédent).

 (maintenu)

Affiche la première ligne.

10 FOR I=65 TO 90	
-------------------	--

(relâchez la touche)

Exécute l'instruction et interrompt le programme. Le HP-71 allume le témoin **SUSP**. (maintenu)

Affiche l'instruction suivante.

20 DISP I @	SUSP
-------------	------

(relâchez la touche)

Le symbole @ indique qu'il y a une autre instruction sur la même ligne.

65	SUSP
----	------

Affiche la valeur de I.

 (maintenu)

Affiche l'instruction suivante sur la ligne.

20@ DISP CHR\$(I) @	SUSP
---------------------	------

Le 1^{er} symbole @ indique qu'une instruction précède celle-ci sur la ligne et le 2^{me} indique qu'une instruction suit.

(relâchez la touche)

```
A SUSP
```

Exécute l'instruction.

Affiche un caractère.

f **SST** (maintenu)

```
200@ DISP CHR$(I+32) SUSP
```

Affiche la dernière instruction de la ligne.

(relâchez la touche)

```
@ SUSP
```

Exécute l'instruction.

f **CONT**

Continue l'exécution de façon normale.

Pour les longs programmes, vous pouvez exécuter pas à pas une petite partie du programme et poursuivre le reste du programme en appuyant sur **f** **CONT**. Si vous voulez commencer l'exécution pas à pas ailleurs qu'au début du programme, vous pouvez placer le pointeur d'exécution à l'endroit où vous voulez commencer en exécutant `GOTO` au clavier. La ligne spécifiée dans l'instruction `GOTO` devient alors la ligne courante.

Vous pouvez aussi placer une instruction `PAUSE` dans le programme à l'endroit où vous voulez commencer l'exécution pas à pas et appuyer sur **f** **SST** lorsque l'exécution s'interrompt à cause de l'instruction `PAUSE`. Lorsqu'un programme est interrompu, vous pouvez utiliser **f** **SST** pour exécuter pas à pas les instructions du programme à partir de l'instruction d'interruption. Vous pouvez aussi relancer l'exécution normale du programme à tout moment en appuyant sur **f** **CONT**.

Les conditions `TRACE VARS` et `TRACE FLOW` peuvent être actives pendant l'exécution pas à pas d'un programme.

Contrôle des erreurs par programme

Lors de la plupart des erreurs d'exécution, le HP-71 arrête l'exécution du programme et affiche un message. Dans certains cas, si vous anticipez des erreurs, il peut être préférable que le programme ne s'arrête pas. Le HP-71 vous permet d'écrire des routines de traitement pour les erreurs que vous pouvez anticiper. Lorsqu'une telle erreur apparaît, le HP-71 transfère l'exécution à la routine de traitement sans arrêter l'exécution. Le HP-71 possède plusieurs ordres vous permettant d'écrire ces routines et de les utiliser.

Branchement pour une erreur (ON ERROR, OFF ERROR)

L'ordre ON ERROR provoque un branchement à une ligne spécifiée lorsqu'une erreur apparaît.

Les deux formes de ON ERROR sont:

— syntaxe simplifiée —

```
ON ERROR GOTO numéro de ligne
ON ERROR GOTO label
```

— syntaxe simplifiée —

```
ON ERROR GOSUB numéro de ligne
ON ERROR GOSUB label
```

L'ordre ON ERROR GOTO provoque un branchement à une instruction et ON ERROR GOSUB provoque un branchement à une routine. Dans ce dernier cas, lorsque la routine est terminée, l'exécution retourne automatiquement à l'instruction suivant celle où l'erreur a eu lieu.

Lors de l'exécution de ON ERROR, le HP-71 active une condition ON ERROR qui reste active jusqu'à ce qu'elle soit explicitement désactivée, modifiée ou jusqu'à la fin du programme*. Pour modifier une condition ON ERROR, il suffit d'exécuter ON ERROR à nouveau.

Pour annuler une condition ON ERROR, exécutez:

```
OFF ERROR
```

Les branchements pour erreurs n'auront plus lieu sauf si vous exécutez un nouvel ordre ON ERROR.

* Les conditions ON ERROR ne sont pas globales. Si vous déclarez une telle condition dans le programme principal, elle n'existe pas automatiquement pour les sous-programmes. Si vous la déclarez dans un sous-programme, elle n'est pas automatiquement active dans le programme principal ou dans les autres sous-programmes. Consultez le chapitre 12, «Sous-programmes et fonctions définies par l'utilisateur».

Détermination du numéro de message d'erreur (ERRN)

Chaque message d'erreur du HP-71 possède un numéro d'identification unique (les numéros d'identification des messages d'erreur sont listés dans le chapitre «*Messages d'erreur, d'avertissement et de système*» du *Manuel de référence* du HP-71). Les messages sont groupés par type pour faciliter leur recherche. Les numéros d'identification des messages d'erreurs mathématiques par exemple sont tous compris entre 1 et 21.

Dans certaines applications, un programme peut déterminer le type de la dernière erreur commise en utilisant la fonction ERRN.

```
ERRN
```

ERRN donne le numéro d'identification du message d'erreur le plus récent.

Exemple: Générez une erreur puis déterminez-en le numéro de message.

Entrée/Résultat

```
DEFAULT OFF  END LINE
```

Traite les exceptions mathématiques comme des erreurs.

```
LOG(-5)  END LINE
```

Exécute une fonction en utilisant un argument non valide de façon à provoquer une erreur.

```
ERR:LOG(neg)
```

Le HP-71 affiche le message d'erreur.

```
ERRN  END LINE
```

Affiche le numéro du message.

```
13
```

Affiche le numéro de l'erreur la plus récente.

Si vous connaissez à l'avance les erreurs que vous voulez traiter dans une routine de traitement, vous pouvez inclure un test du numéro d'erreur de façon à déterminer le type d'opération à effectuer.

Exemple: Rédigez un programme affichant le carré du logarithme d'un nombre entré au clavier. Placez dans ce programme une routine de traitement d'erreur pour les nombre négatif ou nuls.

```
10 DEFAULT OFF @ DESTROY N
```

```
20 INPUT "CARRE DU LOG DE ?"; N
```

```
30 ON ERROR GOSUB 70
```

```
40 DISP LOG(N)^2
```

```
50 OFF ERROR
```

```
60 GOTO 20
```

```
70 IF ERRN=12 THEN DISP "N'ACCEPTÉ  
PAS 0"
```

```
80 IF ERRN=13 THEN DISP "N'ACCEPTÉ  
PAS <0"
```

```
90 RETURN
```

Traite toutes les conditions d'erreur comme des erreurs.

Entrez un nombre.

Transfère l'exécution en ligne 70 en cas d'erreur.

Calcule et affiche le carré du logarithme du nombre.

Désarme la condition ON ERROR.

Retourne au début de la boucle pour attendre un autre nombre.

Affiche un message si le numéro de l'erreur est 12.

Affiche un message si le numéro de l'erreur est 13.

ERRN indique en outre le type d'unité, la ROM enfichable ou le fichier LEX qui a généré le message*. Les numéros de message ont le format suivant:

iiimmm

où *iii* est un numéro d'identification LEX à trois chiffres et *mmm* est un numéro d'identification de message. Le HP-71 supprime tous les zéros d'en-tête de ce nombre. Exemple: ERRN donne simplement un numéro d'identification de message pour les erreurs générées par l'ordinateur car il a un numéro d'identification LEX nul.

Le numéro d'identification LEX indique l'unité ou le fichier LEX ayant généré l'erreur. Le manuel d'utilisation de chaque extension ou ROM enfichable indique son numéro d'identification LEX.

Par exemple, le module d'interface HP-IL HP 82401A possède le numéro d'identification LEX 255 et peut générer ses propres messages d'erreur. Si vous utilisez une interface HP-IL et provoquez l'erreur 7 de ce module, ERRN donne la valeur 255007. Les trois premiers chiffres (255) indiquent que l'unité est l'interface HP-IL et les trois derniers qu'il s'agit de l'erreur 7.

* Un fichier LEX est un fichier d'extension de langage, c'est-à-dire un programme binaire qui ajoute des mots-clés au HP-71. Les fichiers LEX peuvent se trouver en mémoire utilisateur ou dans une extension enfichable. Pour plus d'informations sur les fichiers LEX, consultez le chapitre 8, «*Rédaction et exécution de programmes*».

Rappel d'un message d'erreur (`9 ERRM`, `ERRM#`)

Certaines applications demandent de conserver un message d'erreur ou de le combiner avec d'autres messages. Pour conserver ou manipuler un message, vous devez le rappeler. Vous pouvez rappeler les messages d'erreur du HP-71 de deux façons:

- Appuyez sur `9 ERRM` et maintenez la touche enfoncée. Cette séquence affiche le message d'erreur le plus récent.
- Exécutez `ERRM#`. Cette fonction donne le message d'erreur le plus récent sous forme d'une chaîne alphanumérique.

```
ERRM#
```

`ERRM#` est particulièrement utile lorsque vous désirez personnaliser les routines de traitement d'erreur. Exemple: une erreur peut transférer l'exécution à une routine qui affecte un message d'erreur à une variable, y ajoute du texte et affiche la variable comme un message personnalisé.

Localisation d'une erreur (`ERRL`)

La fonction `ERRL` vous permet de déterminer la ligne de programme de l'erreur la plus récente (si elle a eu lieu dans un programme en cours d'exécution). Cette fonction donne toujours un numéro de ligne. Exemple: si une erreur a lieu dans la ligne 50 de votre programme, l'instruction

```
DISP "Erreur en ligne"; ERRL
```

donne l'affichage de

```
Erreur en ligne 50
```

Messages d'avertissement

Les messages d'avertissement indiquent des conditions qui ne requièrent pas l'arrêt de l'exécution du programme et qui peuvent être directement traitées par le HP-71. Ces avertissements couvrent l'état de la batterie, des informations du lecteur de cartes, des informations sur les fichiers et les dépassements de capacités (décrits dans la section «*Exceptions mathématiques*» ci-dessous).

Lorsqu'un avertissement apparaît:

1. Le HP-71 affecte à `ERRN` le numéro du message d'avertissement. Si l'avertissement apparaît pendant l'exécution d'un programme, le HP-71 affecte à `ERRL` le numéro de la ligne.
2. Le HP-71 affiche le numéro de ligne (si l'avertissement apparaît dans un programme) et le message, accompagné d'un bip. Exemple:

```
WRN L50:Invalid TAB
```

Le message reste affiché pendant la durée définie par l'ordre `DELAY`.

3. Le HP-71 substitue une valeur par défaut et relance l'exécution (vous pouvez choisir le type de valeur par défaut utilisé pour certaines expressions. Consultez pour cela la section «*Exceptions mathématiques*» ci-dessous).

L'affichage des avertissements n'est pas affecté par une condition `ON ERROR`. C'est-à-dire que vous ne pouvez pas définir de branchement `ON ERROR` pour une condition d'avertissement.

Vous pouvez supprimer l'affichage des messages d'avertissement en armant l'indicateur `-1` (consultez le chapitre 11, «*Indicateurs d'état*», pour plus d'informations sur la manipulation des indicateurs). Lorsque cet indicateur est armé, le HP-71 substitue une valeur par défaut pour toute expression provoquant un avertissement. Le HP-71 n'affiche pas de message d'avertissement et n'affecte pas de valeurs à `ERRL` et `ERRN`, Vous n'avez alors aucune indication qu'une condition d'avertissement est apparue. Cette procédure est utile lorsque vous voulez limiter au maximum les interruptions du programme.

Exceptions mathématiques

Les exceptions mathématiques sont des conditions d'erreur que le HP-71 peut traiter comme des erreurs ou comme des avertissements (les exceptions mathématiques sont décrites au chapitre 2). Le HP-71 possède cinq indicateurs associés aux cinq exceptions mathématiques.

Exceptions comme erreurs

Lorsque la valeur de trappe d'un indicateur d'exception est nulle, le HP-71 traite l'exception associée comme une erreur. Lorsque l'exception apparaît:

- L'exécution s'arrête et le HP-71 affiche un message d'erreur.
- Si vous avez exécuté `ON ERROR`, le HP-71 transfère l'exécution du programme à la ligne spécifiée dans l'instruction `ON ERROR`.

Exceptions comme avertissements

Lorsque la valeur de trappe de l'exception est 1 ou 2, le HP-71 traite généralement l'exception comme un avertissement et utilise une valeur par défaut dans l'expression ayant provoqué l'exception*. La valeur par défaut utilisée dépend de la valeur de trappe.

Dans de nombreuses applications, il est parfois utile de traiter les exceptions mathématiques comme des avertissements. Dans ce cas, le HP-71 affecte les valeurs par défaut choisies aux expressions provoquant les exceptions. Comme vous pouvez anticiper ces conditions, il n'est pas toujours utile d'afficher les messages d'avertissement. Pour supprimer l'affichage de la plupart des message d'avertissement, exécutez

```
SFLAG -1
```

Le HP-71 n'affichera pas la plupart des messages d'avertissement jusqu'à ce que vous exécutiez

```
CFLAG -1
```

Pour plus d'informations sur ces ordres, consultez le chapitre 11, «Indicateurs d'état».

* L'indicateur d'exception IWL constitue une exception à cette procédure. Lorsque la valeur de trappe pour IWL vaut 1, le HP-71 traite l'exception IWL comme une erreur. Pour plus d'informations sur les exceptions mathématiques, consultez le chapitre 2.

Branchement, boucle et exécution conditionnelle

Table des matières

Présentation	178
Branchements inconditionnels	179
Branchement à une ligne ou à un label (GOTO, GOSUB, RETURN, POP) ..	179
Branchement à un sous-programme (CALL)	180
Branchement à un autre programme (CHAIN)	180
Branchements calculés (ON...GOSUB, ON...GOTO)	181
Branchements par l'horloge	182
Branchements par l'horloge avec GOTO (ON TIMER #...GOTO)	182
Branchements par l'horloge avec GOSUB (ON TIMER #...GOSUB)	183
Registres d'horloge	183
Désactivation d'un registre (OFF TIMER #)	184
Boucles	185
Boucles simples (FOR...NEXT)	185
Boucles emboîtées	186
Exécution conditionnelle (IF...THEN, IF...THEN...ELSE)	187
Branchement conditionnel	187
Option ELSE	188

Généralités

Le HP-71 possède plusieurs ordres de branchement et de bouclage qui vous permettent de contrôler l'ordre d'exécution des instructions des programmes.

Ce chapitre décrit:

- Les branchements inconditionnels.
- Les branchements calculés.
- Les branchements par l'horloge.
- Les boucles.
- L'exécution conditionnelle.

Un autre type de branchement, le branchement pour erreur, est traité au chapitre 9, «Conditions d'erreur».

Branchements inconditionnels

Branchement à une ligne ou à un label (GOTO, GOSUB, RETURN, POP)

L'ordre GOTO transfère l'exécution d'un programme à une ligne ou un label spécifié.

— syntaxe simplifiée —

```
GOTO numéro de ligne
GOTO label
```

Les branchements avec GOTO sont dits inconditionnels car ils se produisent lors de chaque exécution de l'instruction. GOTO ne peut pas transférer l'exécution à une fonction définie par l'utilisateur ou à un sous-programme (ces derniers sont décrits au chapitre 12).

L'ordre GOSUB provoque un branchement conditionnel à une sous-routine. Le HP-71 garde en mémoire l'emplacement de chaque ordre GOSUB exécuté de façon à pouvoir renvoyer l'exécution à l'instruction suivante lorsque la sous-routine se termine (par RETURN).

— syntaxe simplifiée —

```
GOSUB numéro de ligne
GOSUB label
```

De même que GOTO, l'ordre GOSUB ne peut pas transférer l'exécution à un sous-programme ou à une fonction définie par l'utilisateur.

L'ordre RETURN marque la fin d'une sous-routine et renvoie l'exécution à l'instruction suivant le dernier GOSUB exécuté.

```
RETURN
```

Une sous-routine peut contenir un GOSUB qui transfère l'exécution à une autre sous-routine. Lorsque le HP-71 rencontre un ordre RETURN, il renvoie l'exécution à l'instruction suivant le GOSUB de la sous-routine précédente. Le deuxième ordre RETURN renvoie l'exécution à l'instruction suivant le premier GOSUB exécuté. Les sous-routines peuvent ainsi provoquer des branchements à d'autres sous-routines: ces sous-routines sont alors dites *emboîtées*. Le nombre d'emboîtements n'est limité que par la taille de la mémoire vive de l'ordinateur. Les sous-routines emboîtées se terminent généralement dans l'ordre inverse des branchements qui les ont lancées: une instruction RETURN termine l'exécution de la dernière sous-routine commencée.

Dans certaines applications, vous pouvez utiliser POP pour changer la séquence selon laquelle les sous-routines se terminent.

```
POP
```

Lorsqu'un branchement à une sous-routine a lieu, le HP-71 garde en mémoire l'emplacement du GOSUB ayant provoqué le branchement. Si vous exécutez un ordre POP, le HP-71 efface l'adresse du GOSUB du dernier branchement. Vous pouvez ainsi «sauter» un retour de sous-routine.

Branchement à un sous-programme (CALL)

L'ordre CALL permet de transférer l'exécution à un sous-programme (ou à un autre programme) et, une fois l'exécution terminée, de retourner à l'instruction suivant l'instruction d'appel.

— syntaxe simplifiée —

```
CALL nom de sous-programme [paramètres]
```

Cet ordre est similaire à GOSUB car, à la fin du sous-programme, l'exécution retourne à l'instruction suivant le CALL. La différence réside dans le fait que GOSUB provoque un branchement à une *sous-routine* alors que CALL provoque un branchement à un *sous-programme*. De plus, CALL peut transférer l'exécution à un sous-programme situé dans un autre fichier que le fichier courant.

La rédaction et l'utilisation des sous-programmes sont décrites au chapitre 12, «*Sous-programmes et fonctions définies par l'utilisateur*».

Branchement à un autre programme (CHAIN)

Vous pouvez provoquer un branchement inconditionnel à un autre programme en utilisant l'ordre CHAIN.

— syntaxe simplifiée —

```
CHAIN nom de fichier
```

L'ordre CHAIN supprime le fichier courant de la mémoire vive et y copie le fichier spécifié puis exécute ce dernier. Le fichier enchaîné devient le fichier courant. Le programme à chaîner doit se trouver dans la mémoire de l'ordinateur ou dans une unité de mémoire externe telle qu'une carte magnétique ou une cassette numérique.

Nota: L'ordre CHAIN effaçant le programme courant de la mémoire, veillez à ne pas utiliser une copie unique d'un programme pour exécuter cet ordre!

Grace à l'ordre CHAIN, vous pouvez diviser en petites unités les programmes trop grands pour tenir en mémoire et stocker les morceaux sur un support de mémoire de masse dans des fichiers différents. Lors de l'exécution, chaque portion du programme enchaîne l'exécution de la suivante.

CHAIN conserve l'état de toutes les variables, les modes, les trappes, les indicateurs et les fichiers de données ouverts entre les programmes. Il supprime cependant tous les environnements et informations de contrôle de programme locaux (les environnements de programmes sont décrits au chapitre 12, «Sous-programmes et fonctions définies par l'utilisateur»). Lors de l'enchaînement d'un programme, son exécution commence à la ligne de plus petit numéro.

Exemples:

CHAIN LISTE

Enchaîne un programme, LISTE, et en commence l'exécution à la ligne de plus petit numéro.

CHAIN LISTE: CARD

Enchaîne le programme LISTE à partir du lecteur de cartes magnétiques.

Branchements calculés (ON...GOTO, ON...GOSUB)

Les ordres ON...GOSUB et ON...GOTO permettent d'effectuer des branchements dépendant de la valeur d'une expression numérique.

— syntaxe simplifiée —

```
ON expression GOTO numéro de ligne, numéro de ligne...
                    label           label
```

— syntaxe simplifiée —

```
ON expression GOTO numéro de ligne, numéro de ligne...
                    label           label
```

Lorsque le HP-71 rencontre l'un de ces ordres dans un programme, il évalue l'expression numérique et en arrondit la valeur à un entier. Si l'expression vaut 1, le HP-71 transfère l'exécution au premier numéro de ligne ou label suivant le GOTO ou GOSUB. Si l'expression vaut deux, le HP-71 transfère l'exécution au deuxième numéro de ligne ou label etc.

Exemple: L'instruction suivante transfère l'exécution selon la valeur de $(T=Z) + (A=Z) + 1$.

```
ON (T=Z)+(A=Z)+1 GOTO PREMIER, 200, DERNIER
```

Si $(T=Z) + (A=Z) + 1$ vaut 1, le HP-71 transfère l'exécution au label PREMIER. Si l'expression vaut 2, le HP-71 transfère l'exécution à la ligne 200. Et si l'expression vaut 3, le HP-71 transfère l'exécution au label DERNIER.

Branchements par l'horloge

Le HP-71 possède trois registres d'horloge que vous pouvez utiliser dans un programme pour provoquer un branchement à une ligne ou à un label. Une des applications principales des registres d'horloge consiste à exécuter des routines à intervalles réguliers. Les ordres `ON TIMER #...GOSUB` et `ON TIMER #...GOTO` vous permettent de synchroniser des branchements à des sous-routines ou à des programmes.

Branchements par horloge avec GOTO (`ON TIMER #...GOTO`)

L'ordre `ON TIMER #...GOTO` transfère l'exécution à une ligne ou à un label lorsque le registre d'horloge spécifié atteint la valeur programmée. Le HP-71 termine l'instruction en cours avant d'effectuer le branchement.

— syntaxe simplifiée —

```
ON TIMER # numéro de registre , secondes GOTO numéro de ligne
                                     label
```

Le *numéro de registre* est une expression numérique dont la valeur doit s'arrondir à 1, 2 ou 3. Cette valeur indique le registre d'horloge à utiliser. Le paramètre *secondes* est aussi une expression numérique et définit le nombre de secondes entre le moment où le registre est déclenché et celui où il expire. Vous pouvez définir les intervalles de temps avec une précision du 1/10 de seconde. L'intervalle des valeurs pour le paramètre *secondes* est 0,1 à 134.217.727,0 secondes, soit un peu plus de quatre ans. Si vous spécifiez une valeur inférieure à 0,1 seconde, le HP-71 utilise 0,1 seconde. Si vous spécifiez une valeur supérieure au maximum, le HP-71 utilise la valeur maximum.

Une fois le registre déclaré, il reste actif jusqu'à la fin du programme ou son annulation (voir ci-après sous le titre «*Annulation d'un registre d'horloge*»). Les registres restent actifs lorsque les programmes sont suspendus mais ils ne provoquent le branchement spécifié que lors de la relance du programme.

Exemple:

```
10 ON TIMER # 1, 10 GOTO 50
20 GOTO 20
50 DISP "DIX SECONDES"
60 BEEP 100
70 DISP "CONTINUE"
```

Le registre #1 transfère l'exécution à la ligne 50.

L'exécution continue comme après tout autre branchement.

Lorsqu'un registre déclaré par `ON TIMER #...GOTO` expire, le HP-71 le redéclare automatiquement en utilisant le même intervalle. Un registre déclaré avec un intervalle de 30 secondes, expire donc toutes les 30 secondes en provoquant un branchement.

Branchements par horloge avec GOSUB (ON TIMER #...GOSUB)

L'ordre ON TIMER #...GOSUB provoque un branchement à la sous-routine spécifiée lorsque le registre spécifié expire.

— syntaxe simplifiée —

```
ON TIMER # numéro de registre , secondes GOSUB numéro de ligne
                                     label
```

Avec ON TIMER #...GOSUB, le HP-71 transfère l'exécution à la sous-routine spécifiée lorsque le registre spécifié expire. Lorsque le HP-71 rencontre ensuite un ordre RETURN, il redéclare le registre et renvoie l'exécution à l'instruction suivant celle en cours d'exécution lorsque le registre a provoqué le branchement.

Exemple:

<pre>10 ON TIMER #2, 15 GOSUB 100</pre>	<p>Déclare un branchement à la sous-routine de la ligne 100 à intervalles de 15 secondes.</p>
<pre>20 DISP "ATTENTE"</pre>	
<pre>30 GOTO 20</pre>	
<pre>100 DISP "REGISTRE EXPIRE"</pre>	<p>Lorsque le registre expire, le HP-71 transfère l'exécution à cette ligne.</p>
<pre>110 RETURN</pre>	<p>Lorsque le HP-71 rencontre cet ordre, il renvoie l'exécution à l'instruction suivant celle en cours d'exécution lors du branchement.</p>

Registres d'horloge

Registres dans des sous-programmes. Vous pouvez déclarer des registres d'horloge dans les sous-programmes comme dans le programme principal. Néanmoins, leurs effets sont *locaux*, c'est-à-dire qu'un registre ne peut provoquer un branchement que lorsque le programme ou sous-programme qui l'a déclaré est en cours d'exécution. De plus les registres ne peuvent provoquer que des branchements à une ligne ou un label dans le programme ou sous-programme qui l'a déclaré.

Si une routine déclare un registre puis appelle un sous-programme, le registre ne peut pas provoquer de branchement jusqu'à ce que l'exécution retourne à la routine d'appel. De la même façon, si un sous-programme déclare un registre puis se termine, la déclaration reste active mais ne peut pas provoquer de branchement sauf si l'exécution retourne dans ce sous-programme.

Lorsque l'ordinateur est éteint. Si un programme déclare un registre puis exécute `BYE` ou `OFF`, la déclaration reste active. Lorsque l'intervalle défini expire, le HP-71 se rallume et relance l'exécution à la ligne ou au label spécifié dans l'instruction `ON TIMER #`. Cette technique permet par exemple au HP-71 de lire une mesure sur une unité externe (telle qu'un voltmètre) à intervalle spécifié sans devoir rester allumé entre-temps. Le programme peut déclarer un registre, éteindre le HP-71 puis le rallumer lorsque l'intervalle expire, lire la mesure et éteindre à nouveau le HP-71.

Exemple: Le programme suivant déclare un registre, exécute `BEEP`, éteint le HP-71 puis le rallume, exécute un `BEEP` et le réteint toutes les trois secondes.

Le HP-71 exécute les lignes 40 à 80 à chaque fois que le registre expire. Lorsque le HP-71 a émis 10 bips, le programme désactive le registre (voir ci-après sous le titre «*Désactivation d'un registre*»).

10 DESTROY I	Assure que la variable I est disponible.
20 I=1	Définit la valeur initiale de I.
30 ON TIMER # 1,3 GOTO 40	Déclare le registre 1 avec un intervalle de 3 secondes.
40 DISP I	Affiche la valeur de I.
50 BEEP	Emet un bip.
60 I=I+1	Incrémente I.
70 IF I>10 THEN OFF TIMER # 1	Désactive le registre si le HP-71 a déjà émis 10 bips (valeur de I).
80 BYE	Eteint le HP-71.

Désactivation d'un registre (`OFF TIMER #`)

L'ordre `OFF TIMER #` vous permet de désactiver un registre.

```
OFF TIMER # numéro de registre
```

Exemple: l'instruction `OFF TIMER #3` désactive le registre 3.

La fin d'un programme désactive automatiquement les trois registres (pour plus d'informations sur les fins de programme, consultez la section «*Arrêt d'un programme*», page 155).

Boucles

L'exécution répétitive d'une séquence d'instructions est appelée bouclage. Une boucle simple commence par une instruction `FOR` qui initialise la boucle et se termine par une instruction `NEXT`. Les boucles simples peuvent contenir d'autres boucles pour former des *boucles emboîtées*. Les boucles emboîtées sont particulièrement utiles pour traiter les tableaux et les fichiers de données.

Boucles simples (FOR...NEXT)

Les ordres `FOR` et `NEXT` encadrent une série d'instructions exécutées plusieurs fois.

```
FOR compteur = valeur initiale TO valeur finale [STEP incrément]
```

```
NEXT compteur
```

L'ordre `FOR` définit le début de la boucle et initialise le *compteur* qui détermine le nombre d'exécutions. Le compteur doit être une variable numérique simple. Les *valeur initiale*, *valeur finale* et *incrément* sont des expressions numériques. Ces expressions définissent les valeurs initiale et finale du compteur et l'incrément entre des valeurs successives. Si vous ne spécifiez pas l'incrément (la partie `STEP` de l'instruction est optionnelle), le HP-71 utilise la valeur 1.

Exemples:

```
FOR I = 255-X TO 255
:
NEXT I
```

```
FOR J = (A AND B) TO X^2 + X STEP Y
:
NEXT J
```

```
FOR L1 = 100 TO 200 STEP 25
:
NEXT L1
```

L'ordre `FOR` effectue quatre opérations:

- Il affecte au compteur sa valeur initiale.
- Il stocke la valeur finale du compteur.

- Il stocke la valeur de l'incrément.
- Il marque le début de la boucle.

L'ordre `NEXT` effectue trois opérations:

- Il définit la fin de la boucle.
- Il incrémente le compteur de la valeur spécifiée.
- Il teste la valeur courante du compteur par rapport à la valeur finale. Si elle lui est égale ou la dépasse, le programme sort de la boucle et exécute l'instruction suivant `NEXT`. Si la valeur courante ne dépasse pas la valeur finale, le HP-71 transfère l'exécution à l'instruction suivant `FOR`.

Il y a deux règles gouvernant les branchements dans et hors des boucles `FOR...NEXT`:

- L'exécution d'une boucle `FOR...NEXT` doit toujours commencer à l'instruction `FOR`. Les branchements en milieu de boucle donnent des erreurs lorsque le HP-71 rencontre une instruction `NEXT` sans avoir exécuté une instruction `FOR` correspondante.
- Il est possible de sortir d'une boucle sans la terminer. Lorsque le HP-71 sort d'une boucle, le compteur conserve sa valeur pour une éventuelle utilisation ultérieure.

Boucles emboîtées

Le HP-71 vous permet d'emboîter des boucles `FOR...NEXT` c'est-à-dire de placer une boucle dans une autre. Remarquez cependant que les boucles ne peuvent pas se chevaucher: une boucle emboîtée doit être entièrement contenue dans la boucle supérieure. Le nombre de boucles emboîtées n'est limité que par la quantité de mémoire vive disponible.

Exemple: Le segment de programme suivant, sans numéro de ligne, illustre l'utilisation de boucles emboîtées dans des applications telles que des multiplications de matrices. La matrice `A` est multipliée par `B` pour donner `C`.

```
OPTION BASE 1
M=3@N=4@P=2
DIM A(M,N),B(N,P),C(M,P)
FOR I=1 TO M
  FOR J=1 TO P
    S=0
    FOR K=1 TO N
      S=S+A(I,K)*B(K,J)
    NEXT K
    C(I,J)=S
  NEXT J
NEXT I
```

Lors de l'utilisation de boucles FOR...NEXT emboîtées, veillez à suivre les recommandations ci-après:

- Les instructions FOR et NEXT doivent aller de paire. C'est-à-dire, pour chaque instruction FOR il doit y avoir une instruction NEXT utilisant la même variable de compteur.
- Les boucles emboîtées ne peuvent pas utiliser une même variable pour leurs compteurs.
- Les boucles ne doivent pas se chevaucher.

Exécution conditionnelle (IF...THEN, IF...THEN...ELSE)

L'instruction IF...THEN permet à un programme d'exécuter une ou plusieurs instructions selon la valeur d'une expression numérique.

— syntaxe simplifiée —

```
IF expression numérique THEN instructions
```

Si l'expression vaut zéro (fausse), l'exécution va à la ligne suivante. Si l'expression prend une valeur différente de zéro, le HP-71 exécute les instructions suivant THEN.

Vous pouvez placer une ou plusieurs instructions après THEN en les concaténant avec le symbole @. Certains ordres, cependant, ne sont pas autorisés dans une instruction IF...THEN. Le dictionnaire des mots-clés du *Manuel de référence* indique pour chaque mot-clé si vous pouvez l'utiliser ou non dans une instruction IF...THEN.

Vous ne pouvez pas, par exemple, placer une instruction IF...THEN après THEN mais vous pouvez la placer après ELSE.

Branchements conditionnels

Les instructions IF...THEN sont souvent utilisées pour effectuer des branchements conditionnels. L'utilisation de GOTO ou GOSUB après THEN permet de transférer l'exécution selon le résultat d'un test conditionnel.

— syntaxe simplifiée —

```
IF expression THEN [GOTO] numéro de ligne  
IF expression THEN [GOTO] label
```

— syntaxe simplifiée —

```
IF expression THEN GOSUB numéro de ligne  
IF expression THEN GOSUB label
```

Remarquez que le mot-clé `GOTO` est facultatif immédiatement après `THEN`. Si vous placez un *numéro de ligne* ou un *label* immédiatement après `THEN`, le HP-71 reconnaît cela comme un `GOTO implicite`.
Exemple:

```
IF P1=Q1 THEN 100
```

transfère l'exécution à la ligne 100 si P1 égale Q1.

Par contre, si vous désirez un branchement conditionnel à une sous-routine, vous devez utiliser le mot-clé `GOSUB`. Exemple:

```
IF P1=Q1 THEN GOSUB NVFORM
```

transfère l'exécution à la sous-routine `NVFORM` si P1 égale Q1.

Option ELSE

`ELSE` est un mot-clé optionnel que vous pouvez utiliser avec `IF...THEN`. Ce mot-clé vous permet de spécifier une ou plusieurs instructions à exécuter si l'expression entre `IF` et `THEN` prend une valeur nulle.

— syntaxe simplifiée —

```
IF expression THEN instructions ELSE instructions
```

Lorsque l'expression entre `IF` et `THEN` prend une valeur non nulle (vraie), le HP-71 exécute les instructions situées entre `THEN` et `ELSE`. Lorsque l'expression prend une valeur nulle (fausse), le HP-71 exécute les instructions situées après `ELSE`. L'exécution continue ensuite à la ligne suivante.

Exemples:

```
IF I=0 THEN X=X+Y @ Z=1 ELSE X=X-Y @ Z=2
```

```
IF FLAG(QVF) THEN M$="DEPASSEMENT" ELSE M$="PAS D'ERREUR"
```

Vous pouvez utiliser les ordres `GOSUB` et `GOTO` après `ELSE` de la même façon qu'après `THEN`.

Exemples:

```
IF L THEN "SORTIE" ELSE GOSUB
"REFORMAT"
```

Si L est non nul, le HP-71 transfère l'exécution au label SORTIE (GOTO implicite). Si L est nul, le HP-71 transfère l'exécution au label REFORMAT.

```
IF X>Y THEN 100
```

Si X>Y est vrai, le HP-71 transfère l'exécution en ligne 100.

```
IF T#=A# THEN 245 ELSE FRM1
```

Si T#=A# est vrai, le HP-71 transfère l'exécution en ligne 245 et sinon au label FRM1.

```
IF NOT Y THEN X=T @ GOSUB 50
ELSE 150
```

Si NOT Y est vrai, le HP-71 affecte la valeur de T à X et transfère l'exécution à la sous-routine commençant en ligne 50. Sinon, il transfère simplement l'exécution en ligne 150.

Indicateurs d'état

Table des matières

Présentation	190
Introduction aux indicateurs	191
Tests des indicateurs (FLAG)	191
Armement et effacement des indicateurs	192
Armement des indicateurs (SFLAG)	192
Effacement des indicateurs (CFLAG, RESET)	192
Indicateurs utilisateur	193
Indicateurs système	196
Indicateur de message d'avertissement (-1)	196
Indicateurs de témoin sonore (-2, -25)	197
Indicateur d'affichage continu (-3)	197
Indicateurs d'exceptions mathématiques (-4 à -8)	197
Indicateur de clavier Utilisateur (-9)	197
Indicateur d'unité d'angle (-10)	197
Indicateurs d'arrondis (-11, -12)	198
Indicateurs de format d'affichage (-13, -14)	198
Indicateur de minuscules (-15)	199
Indicateur de limite basse de tableau (-16)	199
Indicateurs de nombre de chiffres (-17 à -20)	199
Indicateur de symbole BASIC (-26)	200
Indicateur EXACT (-46)	200
Indicateurs des témoins affichés (-57, -60 à -64)	201

Présentation

Le HP-71 possède 128 indicateurs binaires d'état. Vous pouvez tous les tester mais vous ne pouvez armer ou effacer explicitement que 96 d'entre-eux. Ce chapitre présente:

- Les types d'indicateurs.
- Les tests des indicateurs.
- L'armement des indicateurs.
- L'effacement des indicateurs.

Introduction aux indicateurs

Le chapitre précédent a décrit les instructions d'exécution conditionnelle qui dirigent le flux d'un programme selon le résultat d'un test conditionnel. Parmi les tests que vous pouvez effectuer se trouvent les *tests d'indicateur*. Un indicateur est soit armé (vrai) soit effacé (faux). Un test d'indicateur est une fonction qui donne l'état d'un indicateur, armé ou désarmé.

Les indicateurs numérotés de -64 à -1 sont des *indicateurs système*. Ils servent au système d'exploitation de l'ordinateur pour indiquer différents états. Les indicateurs système utiles au clavier et dans les programmes BASIC sont décrits ci-après sous le titre «*Indicateurs système*».

Les indicateurs numérotés 0 à 63 sont des *indicateurs utilisateur*. Ces indicateurs n'ont pas de signification particulière pour l'ordinateur. Vous pouvez arbitrairement leur affecter une signification dans un programme pour indiquer une condition non représentée par les indicateurs système (vous trouverez un exemple de programmation avec les indicateurs utilisateur dans la section «*Indicateurs utilisateur*»).

Tests des indicateurs (FLAG)

Vous pouvez tester tous les indicateurs grâce à la fonction FLAG.

```
FLAG(numéro d'indicateur)
```

Cette fonction donne la valeur 1 si l'indicateur spécifié est armé et 0 s'il est effacé.

Vous pouvez tester les 129 indicateurs du HP-71 avec cette fonction. Le *numéro d'indicateur* doit se trouver dans l'intervalle -64 à 63 . Si vous spécifiez un nombre non entier, le HP-71 arrondit cette valeur avant de l'utiliser. Vous pouvez utiliser les mnémoniques des indicateurs d'exceptions mathématiques, telles que IVL et DVZ au lieu des *numéros d'indicateur* pour tester les indicateurs -4 à -8 .

Exemples:

```
FLAG(5)
```

Teste l'indicateur 5.

```
FLAG(IVL)
```

Teste l'indicateur de calcul invalide.

```
FLAG(X)
```

Teste l'indicateur spécifié par X.

```
IF FLAG(-15) THEN 100
```

Transfère l'exécution en ligne 100 si l'indicateur -15 est armé.

```
A=FLAG(12)*5
```

Exécute $A=5$ si l'indicateur 12 est armé et $A=0$ s'il est effacé.

Armement et effacement des indicateurs

Vous pouvez armer et effacer les indicateurs en utilisant les ordres SFLAG, CFLAG et FLAG.

Armement des indicateurs (SFLAG)

L'ordre SFLAG vous permet d'armer un indicateur, tous les indicateurs ou uniquement les indicateurs des exceptions mathématiques. Cet ordre agit sur les indicateurs système -32 à 63 .

```
SFLAG numéro d'indicateur , numéro d'indicateur , ...
SFLAG ALL
SFLAG MATH
```

Exemples:

SFLAG 4,5,25,OVF	Arme les indicateurs 4, 5, 25 et OVF.
SFLAG I,J,K	Arme les indicateurs spécifiés par I, J et K.
SFLAG ALL	Arme les indicateurs 0 à 63.
SFLAG MATH	Arme les cinq indicateurs d'exceptions mathématiques.

Vous pouvez aussi armer un indicateur lorsque vous le testez en utilisant FLAG.

```
FLAG(numéro d'indicateur , nouvelle valeur )
```

Lorsque vous utilisez FLAG pour armer un indicateur, le *numéro d'indicateur* doit se trouver dans l'intervalle -32 à 63 . La *nouvelle valeur* peut être un nombre quelconque y compris Inf et NaN. Si la nouvelle valeur est nulle, le HP-71 efface l'indicateur sinon, pour toute autre valeur, il l'arme. Cette fonction ne vous permet d'armer qu'un seul indicateur à la fois.

Exemple:

FLAG(5,1)	Teste l'indicateur 5 et l'arme.
-----------	---------------------------------

Effacement des indicateurs (CFLAG, RESET)

Vous pouvez effacer les indicateurs avec CFLAG ou FLAG.

```
CFLAG numéro d'indicateur , numéro d'indicateur , ...
CFLAG ALL
CFLAG MATH
```

Les paramètres sont identiques à ceux de SFLAG.

Exemples:

CFLAG 3,5,UNF	Efface les indicateurs 3, 5 et UNF.
CFLAG L,I#J	Efface les indicateurs spécifiés par les expressions numériques.

FLAG vous permet d'effacer des indicateurs lorsque vous les testez.

Exemple:

FLAG(5,0)	Teste l'indicateur 5 et l'efface.
FLAG(5,J)	Teste l'indicateur 5 puis l'arme si J#0 ou l'efface si J=0.
FLAG(5,FLAG(5)-1)	Teste l'indicateur 5 et change son état (l'efface s'il est armé ou l'arme s'il est effacé).

L'exécution de RESET ou l'initialisation de la mémoire (INIT: 3) efface les indicateurs -32 à 63.

RESET

Indicateurs utilisateur

Les indicateurs utilisateur sont numérotés de 0 à 63. Vous pouvez tous les armer, les effacer ou les tester. L'ordinateur n'utilise pas directement ces indicateurs qui n'ont pas d'autres significations que celles que vous leurs affectez.

Les indicateurs 0 à 4 correspondent aux témoins **0, 1, 2, 3, 4** de l'affichage. Le HP-71 allume ces témoins lorsque les indicateurs correspondants sont armés.

Exemple: Dans certains pays étrangers, les distances sont comptées en pouces, pieds et miles et non pas en centimètres. Si vous essayez de lire une carte, il est bien souvent nécessaire de pouvoir convertir les distances sur la carte en distance vraie sur le sol. Vous pouvez écrire un programme utilisant des indicateurs qui accepte des distances sur carte en pouces ou en centimètres et les convertit, selon l'échelle de la carte, en distance réelles en miles ou en kilomètres. Un indicateur représente l'unité sur la carte et l'autre l'unité de distance réelle.

Supposons que vous prépariez une randonnée dans la chaîne des Cascades dans l'Orégon (U.S.A.) et que vous vouliez connaître la distance en ligne droite entre le mont Jefferson et la pointe Grizzly. Vous disposez pour cela d'une carte topographique à l'échelle 1:62500. La distance sur la carte entre les deux sommets est 8,8 centimètres. Combien de miles séparent les deux sommets en ligne droite?

Les formules permettant de convertir les pouces et centimètres en miles et kilomètres sont:

miles = e (pouces / 63360)

miles = e (centimètres / 160934,4)

kilomètres = e (pouces) (,0000254)

kilomètres = e (centimètres) (,00001)

où e est le dénominateur de la fraction d'échelle (dans cet exemple, e vaut 62500).

Le programme suivant vous permet de choisir le type d'entrée, pouces ou centimètres, et le type de résultat, miles ou kilomètres. Le programme utilise les indicateurs 1 et 2 pour représenter les unités utilisées.

Dans le programme, l'indicateur 1 représente le type d'unité pour les distances sur la carte: armé pour les distances en pouces et effacé pour celles en centimètres. L'indicateur 2 représente le type d'unités des distances réelles: armé pour les distances en miles et effacé pour celles en kilomètres.

Le programme arme ou efface l'indicateur 1 en ligne 60 et l'indicateur 2 en ligne 90. IL y a quatre combinaisons possibles des états des indicateurs 1 et 2. Le programme teste l'état de chaque indicateur par les lignes 120 à 150 pour déterminer quelle formule de conversion il doit utiliser. Exemple, si vous mesurez les distances en centimètres sur la carte et voulez connaître les distances réelles en miles, l'indicateur 1 est effacé et l'indicateur 2 est armé. Le programme utilise alors la formule de conversion de la ligne 130.

```
10 'INIT': DELAY 1,1 @ FIX 2
```

```
20 DESTROY S,M$,R$,M,R
```

Initialise les variables utilisées.

```
30 INPUT "ECHELLE?"; S
```

Entre l'échelle de la carte.

```
40 DISP "MESURES"
```

```
50 INPUT " pouces OU cm (P/C)?"; M$
```

Entre l'unité des distances sur la carte.

```
60 IF UPRC$(M$)="P" THEN SFLAG 1
   ELSE IF UPRC$(M$)="C" THEN CFLAG
   1 ELSE 40
```

Arme l'indicateur 1 pour les distances en pouces et l'efface pour celles en centimètres.

```
70 DISP "RESULTATS"
```

```
80 INPUT " mi OU km (M/K)?"; R$
```

Entre l'unité des distances réelles.

```
90 IF UPRC$(R$)="M" THEN SFLAG
   2 @ ELSE IF UPRC$(R$)="K"
   THEN CFLAG 2 ELSE 70
```

Arme l'indicateur 2 pour les distances en miles et l'efface pour celles en kilomètres.

```

110 INPUT "MESURE? "; M @
    IF M <= 0 THEN 170
120 IF FLAG(1) AND FLAG(2)
    THEN R=S*M/
    63360 @ DISP R; " mi"
130 IF NOT FLAG(1) AND
    FLAG(2) THEN R=S*M/
    160934.4 @ DISP R; " mi"
140 IF FLAG(1) AND NOT
    FLAG(2) THEN R=S*M*.0000254
    @ DISP R; " km"
150 IF NOT FLAG(1) AND NOT
    FLAG(2) THEN R=S*M*.00001
    @ DISP R; " km"
160 GOTO CONVERT
170 END
    
```

Entre une distance de carte et branche en fin programme si elle est inférieure ou égale à zéro. Conversion de pouces en miles si les indicateurs 1 et 2 sont armés.

Conversion de centimètres en miles si l'indicateur 1 est effacé et 2 est armé.

Conversion de pouces en kilomètres si l'indicateur 1 est armé et 2 est effacé.

Conversion de centimètres en kilomètres si les indicateurs 1 et 2 sont effacés.

Transfère au label CONVERT en ligne 100.

Exécutez maintenant le programme pour connaître la distance entre les deux sommets.

Entrée/Résultat

ECHELLE ?

Vous demande le dénominateur de la fraction d'échelle.

62500

Entrez l'échelle.

MESURE

pouces OU cm (P/C)?

Vous demande l'unité des distances sur la carte.

Frappez pour centimètres.

RESULTATS

mi OU km (M/K)?

Vous demande l'unité des distances que vous cherchez.

M

Frappez M pour miles.

```
MESURE?
```

Vous demande une mesure.

8.8

Entrez la distance entre le mont Jefferson et la pointe Grizzly mesurée en centimètres sur la carte.

```
3.42 mi
```

Affiche la distance en ligne droite en miles.

```
MESURE?
```

Vous demande une autre mesure.

Ø

Termine le programme.

Le programme continue à vous demander des mesures jusqu'à ce que vous entriez une valeur négative ou nulle. Si vous voulez utiliser d'autres unités de mesure, recommencez le programme.

Indicateurs système

Les indicateurs du HP-71 se répartissent en deux groupes: les indicateurs *système*, dont la signification est déterminée par l'ordinateur, et les indicateurs *utilisateur*, dont la signification est déterminée par l'utilisateur.

Tous les indicateurs du HP-71 sont *globaux*, c'est-à-dire qu'ils peuvent être utilisés par des sous-programmes (même ceux situés dans d'autres fichiers) et par des fonctions définies par l'utilisateur. Aucun indicateur ne peut être local à un fichier ou à un sous-programme.

Les indicateurs système représentent l'état de l'ordinateur. Vous pouvez tester, armer et effacer les indicateurs -32 à -1 mais vous ne pouvez que tester les indicateurs -64 à -33.

La section ci-dessous décrit les indicateurs que vous pouvez utiliser. Les autres indicateurs servent uniquement au fonctionnement interne de l'ordinateur.

Indicateur de message d'avertissement (-1)

Vous pouvez supprimer l'affichage de la plupart des messages d'avertissement en armant l'indicateur -1. Vous ne pouvez pas supprimer les messages de système, les messages d'erreur et certains messages d'avertissement (la section «*Messages d'erreur, d'avertissement et de système*» dans le *Manuel de référence* du HP-71 liste ces messages d'avertissement).

Indicateurs de témoin sonore (−2, −25)

Lorsque l'indicateur −2 est armé, le HP-71 ne pourra pas activer le témoin sonore lors de l'exécution de BEEP. Lorsque l'indicateur −2 est effacé, le témoin sonore fonctionne normalement.

L'indicateur −25 vous permet de choisir entre deux niveaux sonores lors de l'exécution de BEEP. Lorsque l'indicateur est armé, le niveau sonore est élevé et lorsque l'indicateur est effacé le niveau est bas.

Indicateur d'affichage continu (−3)

De façon à économiser la batterie, le HP-71 s'éteint automatiquement après 10 minutes d'inactivité. Si vous désirez laisser l'affichage allumé en permanence, vous pouvez armer l'indicateur −3. L'effacement de l'indicateur −3 ré-active l'extinction automatique.

Indicateurs d'exceptions mathématiques (−4 à −8)

Le HP-71 arme certains de ces indicateurs lorsqu'il rencontre une exception mathématique (pour plus d'informations sur les exceptions mathématiques, consultez la page 57).

Une fois armés, vous pouvez effacer *individuellement* ces indicateurs en exécutant CFLAG ou les effacer *collectivement* en exécutant CFLAG MATH.

Indicateur de clavier Utilisateur (−9)

L'indicateur −9 armé signifie que le clavier Utilisateur est actif. Vous pouvez activer le clavier Utilisateur par programme en armant cet indicateur (équivalent à l'exécution de USER ON). L'effacement de cet indicateur désactive le clavier Utilisateur (équivalent à l'exécution de USER OFF). Cet indicateur est particulièrement utile dans les programmes lorsque vous utilisez des fichiers KEYS.

Indicateur d'unités d'angle (−10)

L'indicateur −10 signifie que les arguments et les résultats des fonctions trigonométriques sont exprimés en radians lorsqu'il est armé et en degrés lorsqu'il est effacé.

Exemple: La routine suivante calcule l'arcsinus d'une valeur entrée et affiche le résultat en degrés ou en radians selon l'état de l'indicateur −10.

```
10 INPUT X
20 IF FLAG(−10) THEN ASIN(X);
   "RADIANS" ELSE ASIN(X);
   "DEGRES"
```

Lorsque l'indicateur −10 est armé, le HP-71 allume le témoin **RAD** à l'affichage. Vous pouvez aussi changer l'unité d'angle en utilisant les ordres RADIANS ou DEGREES.

Indicateurs d'arrondis (-11, -12)

Les indicateurs -11 et -12 déterminent le type d'arrondi courant selon le tableau suivant:

Indicateurs d'arrondis

Type d'arrondi	Indicateur -11	Indicateur -12
Valeur la plus proche	effacé	effacé
Vers zéro	effacé	armé
Valeur supérieure	armé	effacé
Valeur inférieure	armé	armé

Exemple: La routine suivante affiche le type d'arrondi courant.

```

10 A=FLAG(-12)+2*FLAG(-11)
20 IF A=0 THEN R$="PROCHE"
30 IF A=1 THEN R$="ZERO"
40 IF A=2 THEN R$="SUPERIEURE"
50 IF A = 3 THEN R$="INFERIEURE"
60 "ARRONDI:"; R$

```

Vous pouvez choisir le type d'arrondi en armant ou effaçant les indicateurs appropriés ou en utilisant l'ordre `OPTION ROUND`.

Indicateurs de format d'affichage (-13, -14)

Les indicateurs -13 et -14 déterminent le format d'affichage courant conformément au tableau suivant:

Indicateurs de format d'affichage

Format d'affichage	Indicateur -13	Indicateur -14
STD	effacé	effacé
FIX	armé	effacé
SCI	effacé	armé
ENG	armé	armé

Ces indicateurs sont particulièrement utiles pour les routines numériques pour lesquelles le HP-71 doit connaître la façon dont les nombres sont affichés.

Exemple: La routine suivante affiche le format d'affichage courant.

```

10 A=FLAG(-13)+2*FLAG(-14)
20 IF A=0 THEN R$="STD"
30 IF A=1 THEN R$="FIX"
40 IF A=2 THEN R$="SCI"
50 IF A=3 THEN R$="ENG"
60 "FORMAT: "; R$

```

Vous pouvez aussi définir le format d'affichage en utilisant `STD`, `FIX`, `SCI` et `ENG`.

Indicateur de minuscules (-15)

L'indicateur -15 armé signifie que le clavier est en mode minuscule. C'est-à-dire que la pression d'une touche alphabétique donne une lettre minuscule. Lorsque le clavier est en mode majuscule, l'indicateur -15 est effacé.

La pression de `[g]` avant une touche alphabétique, affiche une lettre en majuscule si le clavier est en mode minuscule et vice-versa. Par exemple, lorsque l'indicateur -15 est effacé, `[g][D]` affiche `d` et lorsque l'indicateur -15 est armé, `[g][D]` affiche `D`.

Vous pouvez aussi placer le clavier dans le mode désiré en utilisant les ordres `LC ON` et `LC OFF`.

Indicateur de limite inférieure de tableau (-16)

L'indicateur -16 signifie que la limite inférieure courante des dimensions de tableaux est 1 lorsqu'il est armé et 0 lorsqu'il est effacé. Vous pouvez aussi définir la limite inférieure des dimensions de tableaux en utilisant l'ordre `OPTION BASE`.

Indicateurs de nombre de chiffres (-17 à -20)

Les indicateurs -17 à -20 représentent le nombre de chiffres significatifs dans le format d'affichage courant: `FIX`, `SCI` ou `ENG`. Le tableau suivant indique les nombres de chiffres représentés par les différentes combinaisons de ces quatre indicateurs:

Nombre de chiffres affichés

Nombre de chiffres	Indicateur -17	Indicateur -18	Indicateur -19	Indicateur -20
0	effacé	effacé	effacé	effacé
1	armé	effacé	effacé	effacé
2	effacé	armé	effacé	effacé
3	armé	armé	effacé	effacé
4	effacé	effacé	armé	effacé
5	armé	effacé	armé	effacé
6	effacé	armé	armé	effacé
7	armé	armé	armé	effacé
8	effacé	effacé	effacé	armé
9	armé	effacé	effacé	armé
10	effacé	armé	effacé	armé
11	armé	armé	effacé	armé

Vous pouvez afficher un maximum de 11 chiffres après le point décimal. Ces indicateurs permettent aux programmes de déterminer le nombre de chiffres significatifs affichés.

Exemple: L'instruction suivante donne le nombre de chiffres significatifs courant.

```
FLAG(-17)+2*FLAG(-18)+4*FLAG(-19)+8*FLAG(-20)
```

Vous pouvez définir le nombre de chiffres significatifs avec les ordres `FIX`, `SCI` et `ENG`.

Indicateur de symbole BASIC (-26)

Lorsque l'indicateur -26 est armé, le HP-71 supprime l'affichage du symbole BASIC. Toute pression de `ENDLINE` efface cet indicateur.

Indicateur EXACT (-46)

L'indicateur -46 armé signifie que l'ordre `EXACT` a été exécuté. Le HP-71 efface cet indicateur lors de la ré-initialisation de l'horloge ou de la mémoire. Vous ne pouvez que tester cet indicateur.

Indicateurs des témoins affichés (–57, –60 à –64)

Les indicateurs de témoin représentent l'état des témoins affichés et des conditions associées. Un indicateur armé signifie que le témoin correspondant est allumé. Le tableau suivant résume ces indicateurs et les témoins correspondants.

Indicateurs de témoins

Témoin	Indicateur
AC	–57
Alarme ((•))	–60
BAT	–61
PRGM	–62
SUSP	–63
CALC	–64

Sous-programmes et fonctions définies par l'utilisateur

Table des matières

Présentation	202
Sous-programmes	203
Formes d'un sous-programme (END SUB)	204
Appel de sous-programme (CALL)	205
Environnements de sous-programmes	210
Sous-programmes récursifs	214
Fonctions définies par l'utilisateur	218
Formes d'un fonction définie par l'utilisateur (DEF FN, END DEF)	218
Référence à une fonction définie par l'utilisateur	220
Environnement d'une fonction définie par l'utilisateur	220
Fonctions récursives	222

Présentation

Vous pouvez bien souvent réduire la complexité d'un programme en utilisant des boucles et des sous-routines pour effectuer certaines tâches répétitives. Un programme doit cependant souvent pouvoir transférer l'exécution autour des sous-routines et des boucles lorsque celles-ci ne doivent pas être exécutées. De plus, des routines qui opèrent sur différents jeux de variables utilisent souvent leurs propres jeux de variables pour contenir les valeurs intermédiaires. Pour un grand programme, les opérations sur de nombreuses variables peuvent devenir difficiles à suivre et à contrôler.

Les *sous-programmes* et les *fonctions définies par l'utilisateur*, que vous pouvez utiliser en tout point d'un programme, permettent de surmonter certains des désavantages des sous-routines et des boucles. Ces sous-programmes et fonctions définies par l'utilisateur sont des structures de programmation indépendantes que le HP-71 n'exécute que lorsqu'un programme leur fait explicitement référence. Contrairement à une sous-routine ou à une boucle, il n'est pas nécessaire de contourner les sous-programmes et les fonctions définies par l'utilisateur lorsque vous ne voulez pas les exécuter.

Les sous-programmes et les fonctions définies par l'utilisateur peuvent contenir leurs propres jeux de variables. Lorsque vous rédigez un sous-programme, il n'est pas nécessaire que vous vous souveniez des noms des variables utilisées pas les programmes ou sous-programmes appelant. Le nom d'une variable dans un sous-programme peut être identique à celui d'une variable dans un autre programme ou sous-programme, le HP-71 les considère comme deux variables indépendantes.

Les structures des sous-programmes et des fonctions définies par l'utilisateur vous permettent de pister aisément les valeurs qui y entrent et en sortent. Les sous-programmes et fonctions définies par l'utilisateur offrent les avantages suivants:

- Ils ont des noms par lesquels vous pouvez y faire référence.
- Ils vous permettent de réduire de grands programmes en une série d'unités plus petites, plus simples et indépendantes.
- Ils ne sont exécutés que lorsque vous y faites explicitement référence.
- Les noms de variables dans un sous-programme ou une fonction définie par l'utilisateur peuvent être identiques à ceux utilisés dans d'autres programmes ou sous-programmes.

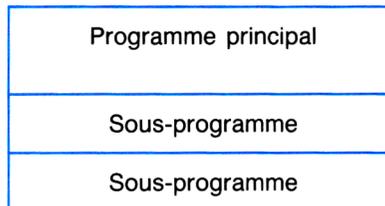
Ce chapitre vous décrit comment incorporer des sous-programmes et des fonctions définies par l'utilisateur dans vos programmes. Il vous présente plus précisément:

- Comment écrire, stocker et utiliser un sous-programme.
- Comment écrire, stocker et utiliser une fonction définie par l'utilisateur.
- Comment utiliser les récurrences.
- Comment le HP-71 maintient différents «environnements» pour les sous-programmes et les fonctions définies par l'utilisateur.

Sous-programmes

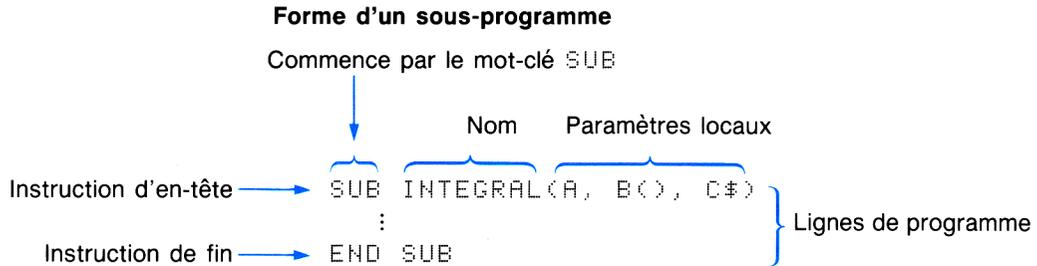
De façon à simplifier les grands programmes, les programmeurs les réduisent souvent à des collections de routines, effectuant chacune une des tâches du programme. Il est alors plus simple et plus rapide de créer une série de courtes routines. Les unités de programme disponibles sur les ordinateurs en BASIC sont généralement limitées aux sous-routines et aux fonctions définies par l'utilisateur. Le HP-71 met à votre disposition un troisième type d'unité: le sous-programme.

Un sous-programme est un groupe de lignes de programme distinct qui peut se trouver dans un fichier BASIC séparé du programme principal (voir illustration ci-dessous). Les programmes (ou sous-programmes) appellent les sous-programmes d'une façon similaire à l'exécution d'une sous-routine.



Forme d'un sous-programme (SUB, END SUB)

Les sous-programmes sont des portions de programme indépendantes, petites mais complètes. Les sous-programmes sont délimités par une *instruction d'en-tête*, qui contient le nom du sous-programme, et une *instruction de fin*. Vous pouvez lancer l'exécution d'un sous-programme en le spécifiant dans une instruction CALL (décrite ci-après).



Ordre SUB. L'ordre SUB doit se trouver sur la première ligne d'un sous-programme. Il identifie le sous-programme et déclare ces paramètres.

— syntaxe simplifiée —

```
SUB nom de sous-programme [(liste de paramètres locaux)]
```

De même que les noms de fichiers, un nom de sous-programme peut se composer d'un maximum de huit lettres ou chiffres et doit commencer par une lettre. Le nom est suivi d'une *liste de paramètres locaux* optionnelle qui spécifient les noms de variables et de tableaux, appelés *paramètres*, qui contiendront les informations transférées au sous-programme par le programme appelant.

Liste de paramètres locaux. Un sous-programme obtient des valeurs du programme appelant par l'intermédiaire de la *liste de paramètres locaux*. Un sous-programme peut en outre renvoyer des valeurs au programme appelant grâce à cette liste. La liste doit se trouver entre parenthèses et les paramètres doivent être séparés par des virgules.

Exemple:

```
SUB SOLUTION(A, B, C1)
```

Le nom du sous-programme est SOLUTION et les paramètres locaux sont A, B et C1.

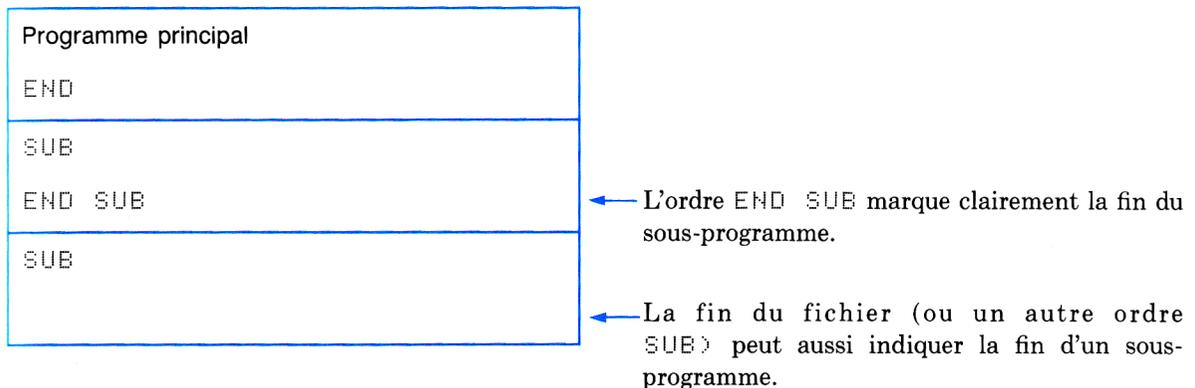
Fin de sous-programme. Un sous-programme se termine par l'ordre END SUB.

```
END SUB
```

La fonction de l'ordre `END SUB` est similaire à celle de `RETURN` car il renvoie l'exécution au programme appelant. Cet ordre efface en outre l'espace mémoire utilisé pour exécuter le sous-programme, le rendant ainsi disponible pour d'autres utilisations (voir section intitulée «*Environnements de sous-programme*» plus loin dans ce chapitre).

Vous pouvez utiliser `END` au lieu de `END SUB` pour terminer un sous-programme. En fait, il n'est même pas nécessaire d'utiliser l'un ou l'autre pour terminer un sous-programme. Tout sous-programme se termine automatiquement lorsque l'ordinateur rencontre un autre ordre `SUB` ou la fin du fichier.

Stockage d'un sous-programme. Les sous-programmes peuvent être stockés dans des fichiers avec le programme principal ou seuls. Un fichier peut contenir plusieurs sous-programmes; néanmoins, tous les sous-programmes doivent se trouver après le programme principal (s'il y en a un) et ne doivent pas être séparés par des lignes de programme (les lignes de programme isolées entre des sous-programmes ne seront jamais exécutées).



Appel de sous-programme (CALL)

Ordre CALL. L'ordre `CALL` appelle un sous-programme et l'exécute.

— syntaxe simplifiée —

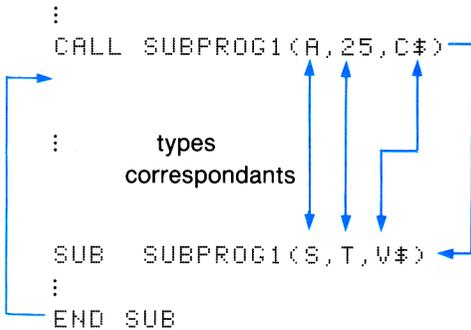
```
CALL nom de sous-programme [⟨liste de paramètres réels⟩] [IN fichier de programme]
```

Les éléments de cet ordre correspondent à ceux de l'ordre `SUB`. Le *nom de sous-programme* est celui du sous-programme à exécuter. La *liste de paramètres réels* est un jeu d'expressions et de variables transférées au sous-programme. Le *fichier de programme* définit le fichier dans lequel le sous-programme se trouve.

`CALL` est similaire à `GOSUB` car il transfère l'exécution à un sous-programme jusqu'à ce que ce dernier se termine puis la retourne à l'instruction suivant le `CALL` ayant provoqué l'appel.

Paramètres réels. Plusieurs ordres `CALL` peuvent appeler le même sous-programme à condition que leurs listes de paramètres réels correspondent à la liste de paramètres locaux de l'ordre `SUB`, élément pour élément, type pour type.

L'illustration ci-dessous montre la correspondance entre les ordres `CALL` et `SUB`.



L'ordre `CALL` appelle le sous-programme et lui transfère des informations.

Renvoie l'exécution à l'instruction suivant `CALL`.

Dans cette illustration, les paramètres locaux `S` et `T` sont des variables numériques et `V#` est une variable alphanumérique. Tout ordre `CALL` appelant `SUBPROG1` doit avoir des valeurs ou expressions numériques comme deux premiers paramètres et une valeur ou expression alphanumérique comme dernier paramètre.

Types de paramètres réels. Dans certaines applications, un programme doit récupérer des informations à la fin de l'exécution d'un sous-programme. Vous pouvez effectuer cela en transférant une variable à un sous-programme qui en modifie la valeur.

Un ordre `CALL` peut transférer des paramètres à un sous-programme de trois façon:

- *Transfert par valeur.* Le sous-programme ne peut pas modifier la variable dans le programme appelant. Les paramètres transférés par valeur doivent être entre parenthèses dans la liste de paramètres réels.
- *Transfert par référence.* Le sous-programme peut modifier la valeur de tout paramètre transféré par référence, c'est-à-dire de toute variable non placée entre parenthèses dans la liste de paramètres réels. Vous ne pouvez transférer par référence que des variables simples ou en tableaux.
- *Transfert par canal.* Les canaux sont identifiés par des numéros associés à des fichiers de données. Les paramètres transférés par canal sont précédés du symbole `#` dans une liste de paramètres locaux (l'utilisation des fichiers de données est décrite au chapitre 14, «Stockage et rappel de données»).

Exemple: Un programme appelle un sous-programme qui calcule l'équivalent octal d'un nombre décimal. Le programme transfère par valeur le paramètre `T` qui contient le nombre décimal et par référence le paramètre `E#` auquel le sous-programme affecte l'équivalent octal.

<pre> : INPUT T : CALL OCTAL(T), E# : SUB OCTAL(A, B#) : B# = <i>équivalent octal</i> : END SUB </pre>	<p><code>T</code> contient le nombre décimal et est transféré par valeur.</p> <p><code>E#</code> contient l'équivalent octal lorsque le sous-programme se termine et est transféré par référence.</p> <p><code>A</code> contient la valeur transférée par <code>T</code>. Les changements de <code>B#</code> résultent dans des changements identiques de <code>E#</code>.</p> <p>Le sous-programme affecte la représentation octale à <code>B#</code>. <code>E#</code> étant le paramètre correspondant, il contient aussi la représentation octale.</p>
---	---

Lors de l'exécution de `CALL`, le HP-71 affecte les valeurs des paramètres transférés par valeur aux paramètres locaux correspondants de l'ordre `SUB`. Le HP-71 «lie» les paramètres transférés par référence à leurs paramètres locaux correspondants de telle façon que tout changement du paramètre local se reflète dans le paramètre de référence. Vous pouvez de cette façon transférer des valeurs du sous-programme au programme appelant.

Toutes les variables déclarées dans un sous-programme sont *locales* à ce sous-programme. C'est-à-dire qu'elles ne sont pas communes au programme appelant. De ce fait, une variable de sous-programme peut avoir le même nom qu'une variable du programme appelant sans que le HP-71 les confonde. De plus, il n'est pas nécessaire qu'un paramètre ait le même nom de variable que son paramètre local correspondant. Un sous-programme opère dans son propre *environnement local* (consultez la section «*Environnements de sous-programme*»).

Les paramètres peuvent être des variables de type quelconque: numérique ou alphanumérique, simple ou en tableau. Les paramètres transférés par valeur peuvent être des expressions numériques ou alphanumériques mais ne peuvent pas contenir de référence à des fonctions définies par l'utilisateur. Les paramètres transférés par référence doivent être des variables.

Les *canaux*, associés à des fichiers de données, peuvent aussi être transférés comme paramètres réels à un sous-programme. Un canal dans une liste de paramètres locaux d'un sous-programme devient associé au même fichier que le paramètre réel correspondant.

Si un sous-programme ouvre un fichier de données en utilisant un numéro de canal qui ne se trouve pas dans la liste de paramètres locaux, ce canal reste local au sous-programme et le HP-71 ferme le fichier de données lorsque le sous-programme se termine. Si un sous-programme ne possède pas de liste de paramètres locaux, les canaux qu'il utilise sont *globaux*. Dans ce cas, le fichier de données associé au canal reste ouvert à la fin du sous-programme (pour plus d'informations sur les canaux, consultez la section «*Fichiers de données*», page 247).

Exemples:

```
CALL SOLUTION(2*X+3+5X-12,A1)
```

Le premier paramètre, $2*X+3+5X-12$, est transféré par valeur et le second, A1, par référence.

```
CALL EDITEUR(DISP#,L#&T#, (X))
```

Tous les paramètres sont transférés par valeur.

Exemple de sous-programme. Maintenant que nous avons décrit certains des éléments d'un sous-programme, nous allons étudier un sous-programme complet et son fonctionnement.

Exemple: La série de Fibonacci est une série mathématique dans laquelle chaque terme est la somme des deux précédents. Les valeurs des deux premiers termes sont définies égales à 1. Le troisième terme est la somme des deux premiers, soit 2. Le quatrième est la somme du second et du troisième, $1 + 2 = 3$, etc.

Le programme suivant, FIBONAC, utilise un sous-programme pour calculer le n^{me} terme de la série de Fibonacci, pour $0 < n < 61$. Il demande une valeur au clavier, appelle un sous-programme pour déterminer le nombre de Fibonacci correspondant et affiche le résultat.

Pour entrer ce programme, frappez EDIT FIBONAC END LINE et les lignes suivantes.

```

10 DESTROY N, A
20 INPUT "TERME? ";N
30 CALL FIBO (N,A)
40 DISP "LE TERME ";N;" VAUT ";A
50 ! FIN DU PROGRAMME PRINCIPAL
60 SUB FIBO(B,C)
70 IF FP(B) OR B<?1 OR B>?60
THEN C=0 @ GOTO 170
80 IF B=1 OR B=2 THEN
C=1 @ GOTO 170
90 X=1
100 Y=1
110 FOR I=3 TO B
120 Z=X
130 X=X+Y
140 Y=Z
150 NEXT I
160 C=X
170 END SUB
    
```

E correspond à N et C correspond à A.
Vérifie qu'il n'y a pas d'erreur.

Affecte 1 à C pour les cas simples. A étant un paramètre transféré par référence, le HP-71 lui affecte la valeur de C.

Y et X sont des termes consécutifs.
Calcule la valeur du terme d'ordre B (N).
Z est une variable temporaire pour stocker X.
X prend la valeur du terme suivant
et Y prend la valeur de X.

X contient la valeur du terme d'ordre B.

FIBO est le nom du sous-programme. Le nombre entré au clavier doit être un entier positif non nul (le programme donne 0 pour toute entrée incorrecte, c'est-à-dire inférieure à 0, supérieure à 60, NaN ou INF) et est affecté à N, qui est ensuite transféré par référence au sous-programme. Le sous-programme affecte la valeur du N^{me} terme de la série au paramètre de référence, A.

Pour utiliser le programme, appuyez sur **RUN** et, lorsque le HP-71 vous le demande, entrez un nombre suivi de **END LINE**. Le programme calcule et affiche alors la valeur du n^{me} terme de la série de Fibonacci.

Entrée/Résultat

RUN

Exécute FIBONAC (en supposant qu'il se trouve dans le fichier courant).

TERME?

Le HP-71 vous demande un nombre.

6 **END LINE**

Choisissez par exemple le sixième terme de la série.

LE TERME 6 VAUT 8

Affiche la valeur du sixième terme.

Appel d'un sous-programme dans un autre fichier. Vous pouvez appeler un sous-programme dans un autre fichier en spécifiant ce dernier dans l'ordre `CALL`.

Exemples:

```
CALL FIBO1((S),T) IN
  MATH:PORT(2)
```

Appelle `FIBO1` dans le fichier `MATH` dans le logement 2.

```
CALL DATE1 IN DATES
```

Appelle `DATE1` dans le fichier `DATES`.

Le HP-71 peut trouver un sous-programme dans un autre fichier même si vous ne spécifiez pas le nom du fichier. Il n'est donc pas nécessaire de spécifier le nom du fichier dans l'ordre `CALL` sauf si un autre sous-programme portant le même nom se trouve en mémoire.

Vous pouvez aussi appeler un programme comme sous-programme mais vous ne pouvez pas lui transférer de paramètres.

— syntaxe simplifiée —

```
CALL [nom de fichier]
```

Lors de l'appel d'un programme, le HP-71 cherche d'abord dans les fichiers en mémoire s'il existe un sous-programme portant le nom spécifié. S'il ne trouve pas de sous-programme, le HP-71 cherche alors un programme et l'exécute comme un sous-programme.

Exemples:

```
CALL
```

Appelle le fichier courant comme un sous-programme.

```
CALL VITESSE
```

Appelle `VITESSE`. Le HP-71 cherche d'abord un sous-programme portant ce nom. S'il n'en trouve pas, il cherche ensuite un programme et l'exécute.

Environnements de sous-programmes

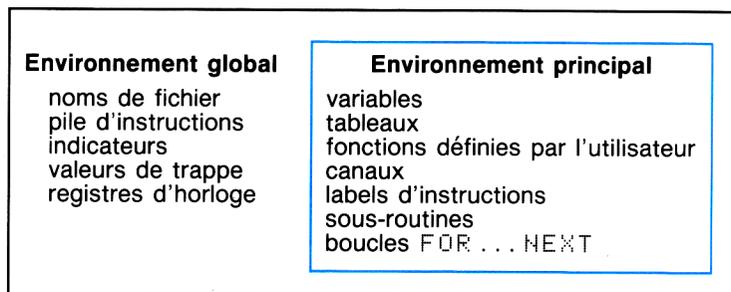
Lorsque vous effectuez des opérations au clavier, vous avez accès à des variables, des fichiers, la pile d'instructions et de nombreuses autres structures de fonctionnement du HP-71. Les programmes ont aussi accès à ces structures. Certaines de ces structures sont disponibles pour tous les programmes et sous-programmes et d'autres ne sont disponibles que pour le programme ou sous-programme dans lequel elles ont été définies. Ces structures forment un *environnement* pour le programme ou sous-programme.

Les structures disponibles pour tous les programmes et sous-programmes forment *l'environnement global* et celles dont l'utilisation est restreinte à un programme ou sous-programme forment *l'environnement local* de ce programme ou sous-programme. A partir du clavier, vous avez toujours accès à l'environnement global et, à un instant donné, à *un* environnement local.

Lorsque vous exécutez `RUN`, le programme utilise le même environnement local que celui disponible au clavier. Cet environnement est appelé *environnement principal* et constitue l'environnement local par défaut de l'ordinateur à cet instant.

L'illustration suivante montre la relation entre l'environnement principal et l'environnement global. Un programme en cours d'exécution a accès aux éléments listés dans l'environnement principal plus ceux listés dans l'environnement global.

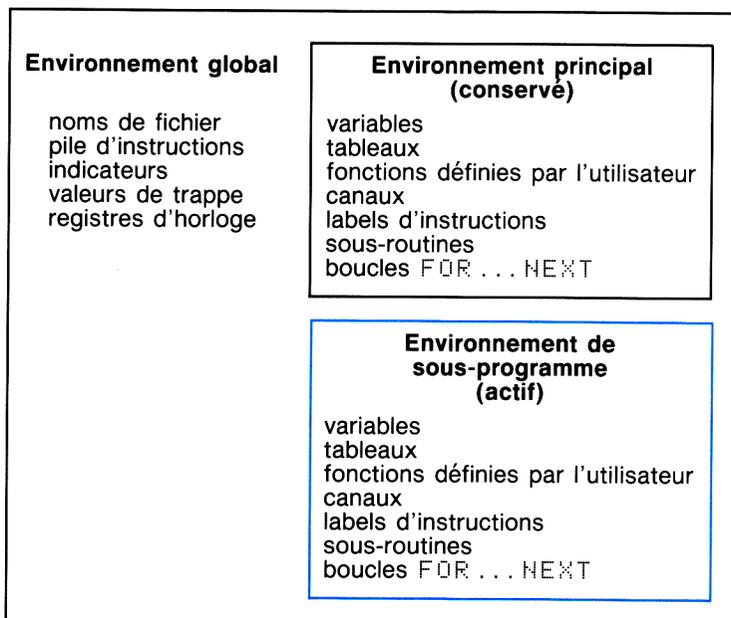
Environnements principal et global



Environnements conservés. Lorsque vous appelez un sous-programme (par un programme ou au clavier), le HP-71 conserve l'environnement principal et crée un nouvel environnement pour le sous-programme. Ce dernier a accès aux éléments de son environnement local et à ceux de l'environnement global mais il n'a pas accès aux éléments de l'environnement conservé. Le nouvel environnement local existe tant que le sous-programme correspondant est actif. Lorsque le sous-programme se termine, le HP-71 efface l'environnement local et ré-active l'environnement principal.

Lorsqu'un environnement local est actif, il devient *l'environnement actif*. Si un sous-programme est interrompu, vous ne pouvez accéder au clavier qu'à cet environnement local et à l'environnement global.

Environnement de programme et de sous-programme

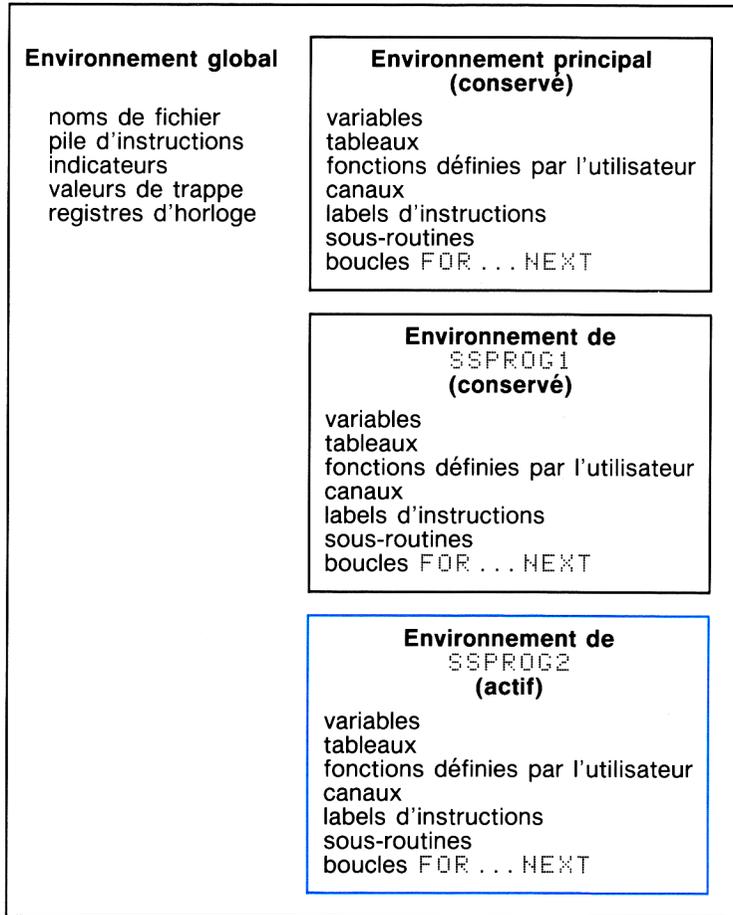


Un environnement ne peut pas accéder aux éléments d'un autre environnement local. Exemple, un programme ou un sous-programme ne peut pas utiliser une fonction définie par l'utilisateur définie dans un autre sous-programme. Par contre tout environnement a accès à tous les éléments de l'environnement global.

Un programme peut appeler un sous-programme, celui-ci peut appeler un autre sous-programme; il n'y a pas de limite au nombre d'appels de sous-programmes emboîtés. Mais chaque appel impose la conservation de l'environnement actif et la création d'un nouvel environnement, utilisant ainsi rapidement la mémoire vive. Le nombre d'appels de sous-programme emboîtés se trouve donc limité par la taille de la mémoire vive.

Ré-activation des environnements. Lorsqu'un sous-programme se termine, l'exécution retourne au programme ou sous-programme appelant. De plus le HP-71 efface l'environnement du sous-programme terminé et ré-active *l'environnement d'appel*, l'environnement du programme ou sous-programme appelant.

Par exemple, si le PROG1 appelle le sous-programme SSSPROG1, le HP-71 conserve l'environnement de PROG1 (principal) et en crée un nouveau pour SSSPROG1. Si ce dernier appelle un autre sous-programme, SSSPROG2, le HP-71 conserve l'environnement de SSSPROG1 et en crée un autre pour SSSPROG2.

Environnement de programme et de sous-programme

Pendant l'exécution de SSPROG2, son environnement devient l'environnement actif. L'environnement d'appel est celui de SSPROG1 et l'environnement d'appel de SSPROG1 est l'environnement principal. Lorsque vous interrompez l'exécution d'un programme, l'environnement auquel vous avez accès au clavier est l'environnement actif au moment de l'interruption. Pour accéder à un environnement conservé, vous devez terminer le sous-programme actif, et éventuellement les sous-programmes d'appel, de façon à activer l'environnement désiré. Vous pouvez effectuer cette opération au clavier en utilisant l'ordre END.

Fin d'un environnement de sous-programme. Lorsqu'un sous-programme exécute `END` ou `END SUB`, le sous-programme se termine et l'exécution retourne au programme ou sous-programme d'appel. Le HP-71 ré-active alors l'environnement d'appel.

Plus précisément, lors de l'exécution de `END` ou `END SUB` dans un sous-programme, le HP-71 effectue les opérations suivantes:

- Renvoie l'exécution au programme ou sous-programme d'appel.
- Ferme tous les fichiers associés aux canaux locaux.
- Efface la mémoire associée à l'environnement local du sous-programme.
- Efface toute condition `ON ERROR` définie dans le sous-programme.
- Ré-active toute condition `ON ERROR` définie dans l'environnement d'appel.
- Ré-active le pointeur `DATA`.

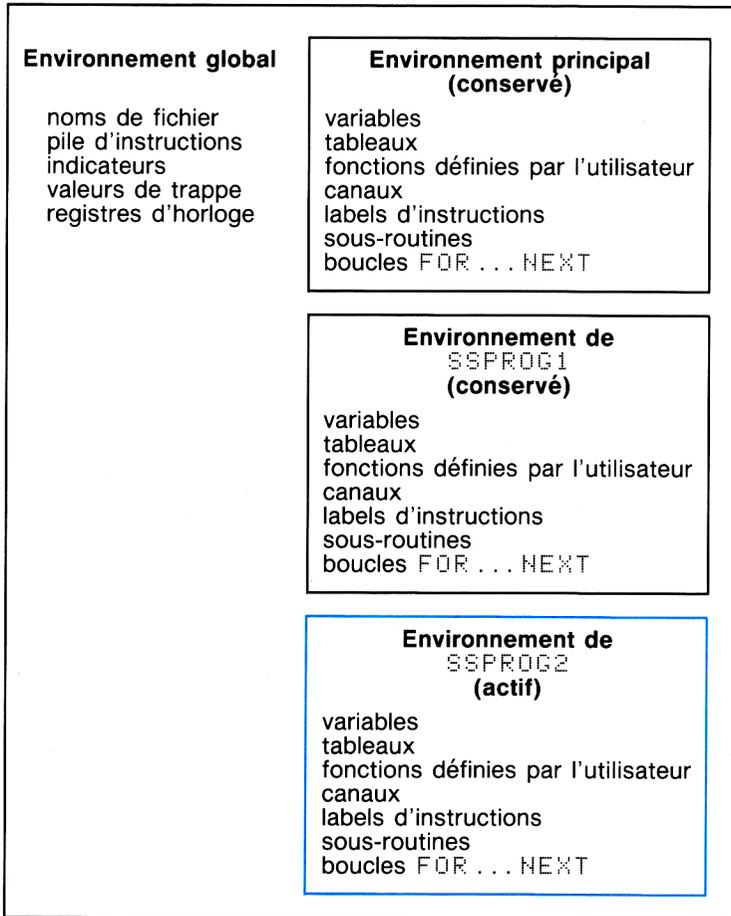
Vous pouvez aussi terminer un programme et effacer l'environnement actif et tous les environnements conservés en utilisant l'ordre `END ALL`. Cet ordre termine l'exécution du programme mais n'affecte pas les variables déclarées dans l'environnement principal.

Si vous exécutez `END` ou `END SUB` au clavier pendant l'interruption d'un sous-programme, l'instruction suivant le `CALL` ayant appelé le sous-programme devient l'instruction d'interruption.

Sous-programmes récursifs

Qu'est-ce que la récurrence? Un sous-programme peut appeler un autre sous-programme de la même façon qu'un programme peut appeler un sous-programme. Un sous-programme peut aussi s'appeler comme sous-programme. Un sous-programme qui s'appelle lui-même est dit *récursif*. Lors de chaque appel récursif, le HP-71 crée un nouvel environnement local, comme pour tout autre sous-programme (voir illustration ci-dessous).

Environnement de programme et de sous-programme



Les appels récursifs facilitent, entre-autres opérations, les recherches et les tris. Les sous-programmes récursifs sont généralement utilisés pour des applications complexes mais peuvent néanmoins simplifier des programmes plus petits. Par exemple, au lieu d'utiliser l'algorithme itératif du programme Fibonacci précédent, vous pouvez simplement définir la valeur du n^{me} terme de la série de Fibonacci égale à la valeur du terme $(n-1)$ plus celle du terme $(n-2)$. Le sous-programme peut ensuite se transférer $n-1$ et $n-2$ comme paramètres pour en déterminer les valeurs.

Exemple: Le programme Fibonacci suivant utilise un sous-programme récursif. Entrez ce programme et exécutez-le comme le précédent (frappez EDIT FIBONAC2).

```

10 DESTROY N,A
20 INPUT "TERME? ";N
30 CALL FIBO(N,A)
40 DISP "LE TERME ";N;" VAUT ";A
50 SUB FIBO(B,C)
60 IF FP(B) OR B<?1 OR B>?60
   THEN C=0 @ GOTO 110
70 IF B=1 OR B=2 THEN C=1 @
   GOTO 110
80 CALL FIBO(B-1,S)
90 CALL FIBO(B-2,T)

100 C=S+T
110 END SUB

```

Donne 0 si la valeur est erronée.

Donne 1 pour les cas simples.

Calcule la valeur du terme d'ordre (B-1) et l'affecte à S.

Calcule la valeur du terme d'ordre (B-2) et l'affecte à T.

Affecte la valeur du terme d'ordre B, $S + T$, à C.

La différence entre cet algorithme et l'algorithme itératif de l'exemple précédent est que ce dernier commence par la valeur du terme le plus simple et continue la série jusqu'au terme d'ordre n alors que l'algorithme récursif commence par le terme d'ordre n et continue jusqu'au terme le plus simple.

Exemple: Exécutez le nouveau programme.

Entrée/Résultat

```
RUN FIBONAC2 
```

Lance le programme.

```
TERME? █
```

Demande un entier.

```
5 
```

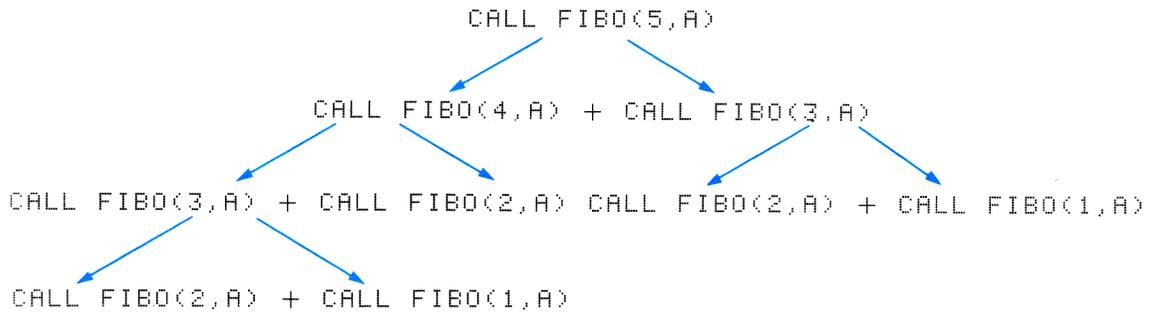
Calcule la valeur du 5^{me} terme.

```
LE TERME 5 VAUT 5
```

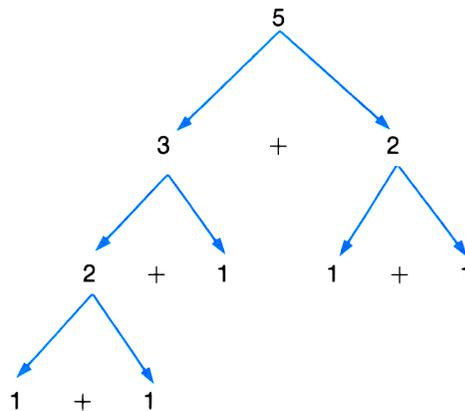
Affiche le 5^{me} terme.

A l'aide de l'algorithme récursif, le sous-programme détermine la valeur d'un terme en s'appelant successivement pour chaque terme d'ordre inférieur. Pour éviter que le sous-programme s'appelle indéfiniment, le programme donne une valeur discrète pour les 1^{er} et 2^{me} termes. Pour tout autre terme le sous-programme s'appelle lui-même. Les différents niveaux d'appel se terminent après le calcul des valeurs pour leurs termes respectifs.

Le sous-programme calcule la valeur d'un terme en calculant récursivement les valeurs des deux termes précédent, comme illustré ci-dessous.



Cette illustration montre les différents niveaux d'appel de sous-programme nécessaires pour déterminer la valeur du 5^{me} terme de la série de Fibonacci. Le sous-programme s'appelle récursivement jusqu'à ce qu'il donne une valeur pour un cas simple. Les appels de sous-programme en attente sont alors évalués jusqu'au premier niveau. En utilisant des nombres au lieu des appels de sous-programme, le diagramme devient:



Vous vous êtes sans doute aperçu que l'algorithme récursif est beaucoup plus lent que l'algorithme itératif. La raison en est que l'algorithme récursif calcule plusieurs fois plusieurs des valeurs intermédiaires et prend donc plus de temps pour arriver au même résultat.

Pour de nombreux problèmes de programmation, un algorithme itératif est la solution la plus efficace mais ce n'est pas toujours le cas. Les sous-programmes récursifs peuvent être beaucoup plus faciles à écrire et s'avérer plus rapides. Néanmoins, les sous-programmes récursifs utilisent beaucoup d'espace mémoire pour stocker toutes les variables de chaque appel.

Fonctions définies par l'utilisateur

Le HP-71 contient de nombreuses fonctions préprogrammées qui doivent répondre à une grande part de vos besoins en programmation. Dans certains cas, vous aurez néanmoins besoin de créer vos propres fonctions. Le HP-71 vous permet de créer des *fonctions définies par l'utilisateur* et de les utiliser comme les fonctions préprogrammées.

Une fonction définie par l'utilisateur peut se trouver en un endroit quelconque d'un programme ou d'un sous-programme mais elle ne peut être utilisée que par le programme ou sous-programme dans lequel elle a été définie. Si le HP-71 rencontre une définition de fonction pendant l'exécution d'une séquence d'instructions, il l'ignore. Vous ne pouvez exécuter une fonction définie par l'utilisateur qu'en la référant dans une expression numérique ou alphanumérique, dans un programme ou au clavier.

Formes d'une fonction définie par l'utilisateur (DEF FN, END DEF)

Vous pouvez créer des fonctions simples ou complexes en utilisant des définitions contenant une ou plusieurs instructions.

Fonctions mono-instructions. Les fonctions mono-instructions commencent par le mot-clé DEF FN et contiennent une expression qui affecte une valeur à la fonction.

— syntaxe simplifiée —

```
DEF FN nom de fonction [liste de paramètres locaux] = expression
```

Un *nom de fonction* se compose d'une lettre ou d'une lettre suivie d'un chiffre pour une fonction numérique. Pour une fonction alphanumérique, le symbole # doit suivre la lettre ou la combinaison lettre-chiffre. Les variables dans la *liste de paramètres locaux* sont des noms de variables qui peuvent être identiques à d'autres noms de variables dans d'autres environnements. Vous pouvez utiliser des paramètres locaux dans l'*expression* pour affecter une valeur à la fonction.

Un ordre DEF FN peut contenir un maximum de 14 paramètres dans sa liste de paramètres locaux. Ces paramètres locaux doivent être des variables numériques ou alphanumériques simples.

Exemples:

```
DEF FNT(A,B) = (A-B) / (A+B)
DEF FNA1#(S2,F#) = F#[S2,S2+6] & F#[S2+12,S2+18]
DEF FNP = 3*P^3+2*P^2+P+5
```

Fonctions multi-instructions. Les fonctions multi-instructions sont encadrées par les ordres DEF FN et END DEF.

— syntaxe simplifiée —

```
DEF FN nom de fonction [(liste de paramètres locaux)]
```

```
END DEF
```

Une fonction multi-instruction suit la forme générale ci-dessous:

DEF FNx(<i>liste de paramètres locaux</i>)	Première ligne.
:	Lignes de programme.
FNx = <i>expression</i>	Instruction d'affectation.
:	Lignes de programme.
END DEF	Ordre de fin.

Exemples:

100 DEF FNT(A,B)	Définit une fonction nommée FNT.
110 A=A+B	
120 B=A+B	
130 FNT=A+B	Affecte une valeur à la fonction.
140 END DEF	Termine la fonction.

100 DEF FNA\$(R\$,T\$,U\$)	Définit une fonction nommée FNA\$.
110 FOR I=1 TO LEN(R\$)	Affecte une valeur à FNA\$.
120 U\$[I,I+1]=R\$[I,I] & U\$[I,I]	
130 NEST I	
140 FNA\$=U\$	
150 END DEF	Termine la fonction.

Référence à une fonction définie par l'utilisateur

Lorsque vous utilisez une fonction dans une expression, vous lui faites *référence*. Pour faire référence à une fonction définie par l'utilisateur, vous pouvez la placer dans une expression comme toute autre fonctions de l'ordinateur. Vous ne pouvez faire référence au clavier à une fonction définie par l'utilisateur que si elle est définie dans le fichier courant (il n'est pas nécessaire que le programme du fichier courant soit interrompu).

Exemples:

```
A# = B# & FNT#(A, B#)
DISP T#FNU(T)/PI
H = FNA2(FNW(X, Y) + FNR)
```

Les paramètres dans la référence de fonction sont similaires aux paramètres réels dans un appel de sous-programme. Ils doivent correspondre aux paramètres locaux de la fonction dans l'ordre de la liste, type pour type. Vous pouvez utiliser des expressions numériques et alphanumériques comme paramètres réels et, contrairement aux paramètres réels des ordres `CALL`, les expressions peuvent contenir des références à des fonctions définies par l'utilisateur. Tous les paramètres réels sont transférés par *valeur*. Une fonction définie par l'utilisateur ne donne pas de valeurs par l'intermédiaire des paramètres réels.

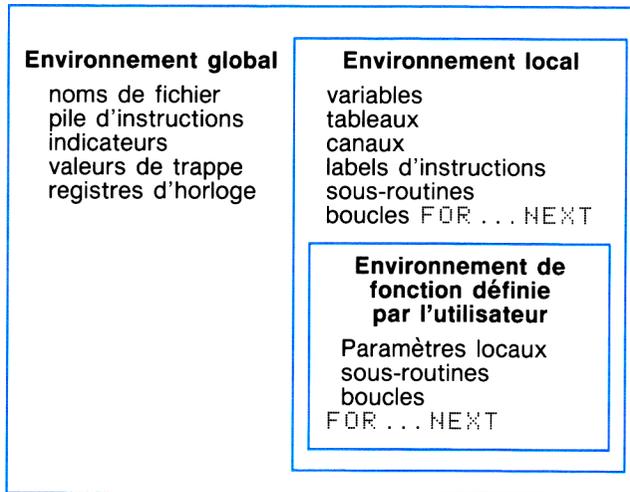
Environnement d'une fonction définie par l'utilisateur

Les fonctions définies par l'utilisateur ne sont disponibles que dans le programme ou sous-programme dans lequel elles ont été définies. Un programme principal ou un sous-programme ne peut pas faire référence à une fonction définie dans un autre programme ou sous-programme.

Une fonction définie par l'utilisateur peut accéder aux variables et canaux de l'environnement dans lequel elle est définie. Mais les variables déclarées dans la liste de paramètres locaux d'une fonction sont locales à celle-ci. De ce fait, une fonction définie par l'utilisateur ne peut pas accéder à une variable de l'environnement dans lequel elle est définie si sa liste de paramètres locaux contient une variable de même nom.

L'illustration suivante montre qu'une fonction définie par l'utilisateur possède son propre environnement comprenant ses paramètres locaux, des sous-routines et des boucles `FOR...NEXT`. Les éléments de l'environnement de la fonction définie par l'utilisateur ne sont pas accessibles par le programme ou sous-programme mais les éléments de l'environnement local du programme ou sous-programme ainsi que les éléments de l'environnement global sont accessibles par la fonction.

Environnements de programme et de fonction définie par l'utilisateur



Une fonction multi-instruction se termine généralement par `END DEF` ou par `END`. Lorsqu'aucun de ces ordres ne termine une fonction définie par l'utilisateur, le HP-71:

- Renvoie l'exécution à l'expression d'appel.
- Efface la mémoire associée à l'environnement de la fonction.

Une fonction définie par l'utilisateur se termine aussi lorsque le HP-71 rencontre `SUB`, `END SUB` ou la fin du fichier.

Lors d'une recherche d'erreur dans une fonction définie par l'utilisateur, il est souvent utile de placer un ordre `PAUSE` dans la définition de la fonction puis de faire référence à la fonction à partir du clavier. L'ordre `PAUSE` interrompt l'exécution et vous permet de consulter les valeurs des variables, des indicateurs ou d'autres aspects de l'environnement global. Vous pouvez ensuite relancer l'exécution en appuyant sur `f` `CONT`. Si vous faites référence à une fonction dans une expression au clavier et interrompez son exécution, le HP-71 ne complète *que* la fonction lorsque vous relancez l'exécution; il n'évalue *pas* la partie de l'expression située à droite de la référence à la fonction. Si vous terminez une fonction interrompue référencée à partir du clavier, le HP-71 ne donne pas de valeur pour la fonction.

Si vous terminez une fonction interrompue référencée par un programme avant qu'une valeur ne soit affectée à la fonction, celle-ci donne la valeur zéro pour une expression numérique et une chaîne vide pour une expression alphanumérique.

Fonctions récursives

De même que les sous-programmes, les fonctions définies par l'utilisateur peuvent être récursives. Le programme de Fibonacci, par exemple, peut être écrit en utilisant une fonction récursive.

Exemple: Entrez le programme suivant et exécutez-le comme les précédents.

La partie principale du programme est la ligne 50 dans la fonction FNF. Dans cette fonction, le HP-71 donne une valeur discrète pour les cas simples (entrée égale à 1 ou 2). Si l'entrée est supérieure à 2, la fonction s'appelle répétitivement. La ligne 50 indique simplement que la valeur de la fonction est 1 si l'entrée est 1 ou 2. Si l'entrée est supérieure à 2, le valeur de la fonction est la somme des valeurs de Fibonacci des deux termes précédents de la série.

La ligne 40 du programme vérifie que les entrées sont valides. Si le HP-71 détecte une entrée non valide, la fonction affiche un message et se termine sans affecter de valeur. Lorsqu'une fonction numérique se termine sans affecter de valeur, l'expression vaut zéro.

10 INPUT "TERME? ";N	Le programme vous demande d'entrer un terme.
20 DISP "LE TERME ";N;" VAUT ";FNF(N)	Appelle la fonction définie par l'utilisateur et affiche sa valeur.
30 DEF FNF(B)	Définit le début de la fonction.
40 IF FP(B) OR B<?1 OR B>?60 THEN 60	Vérifie que l'entrée est valide. En cas d'erreur la fonction donne 0.
50 IF B=1 OR B=2 THEN FNF=1 ELSE FNF=FNF(B-1) + FNF(B-2)	Affecte une valeur à la fonction.
60 END DEF	Termine la définition de la fonction.

L'algorithme pour ce programme est similaire à celui utilisé précédemment pour le sous-programme récursif. Néanmoins, cette fonction est plus courte et plus facile à suivre.

Format d'impression et d'affichage

Table des matières

Présentation	224
Formatage simple	225
Affichage et impression d'informations (DISP, PRINT)	225
Ordre DISP implicite	226
Espace des sorties (TAB)	226
Formatage par chaîne de format	228
Qu'est-ce qu'une chaîne de format?	229
Utilisation d'une chaîne de format (IMAGE, DISP USING, PRINT USING)	230
Contrôle de l'affichage et de l'imprimante	232
Longueur de ligne (WIDTH, PWIDTH)	232
Modification de la séquence de fin de ligne (ENDLINE)	234
Contrôle du curseur et de l'affichage	234

Présentation

Lorsque vous rédigez des programmes en BASIC, il est souvent nécessaire de contrôler la façon dont les informations sont affichées ou imprimées. Vous pouvez pour cela utiliser quelques techniques simples ou, si vous avez déjà une bonne connaissance de la programmation, des techniques plus avancées. Le présent chapitre décrit les formats d'affichage et d'impression simples et complexes:

- Envoi d'informations à l'affichage ou à une imprimante.
- Codage d'informations à afficher ou à imprimer.
- Utilisation des chaînes de format et de l'ordre IMAGE pour formater les sorties.
- Contrôle de la largeur de ligne de l'affichage et de l'imprimante.
- Contrôle de la séquence de fin-de-ligne utilisée par l'affichage et l'imprimante.
- Contrôle de l'affichage et de l'imprimante en utilisant les codes de contrôle et les séquences d'échappement.
- Création et utilisation des champs protégés à l'affichage.

Formatage simple

Affichage et impression d'informations (DISP, PRINT)

Vous pouvez utiliser les ordres `DISP` et `PRINT` pour afficher des informations sur le HP-71. Si vous disposez d'une imprimante, l'ordre `PRINT` envoie les informations à cette imprimante. On appelle *sorties* les informations *envoyées* vers l'affichage ou l'imprimante.

— syntaxe simplifiée —

```
DISP liste d'affichage
```

— syntaxe simplifiée —

```
PRINT liste d'impression
```

La *liste d'affichage* et la *liste d'impression* suivent le même format et contiennent les éléments à envoyer. Les éléments des listes peuvent être:

- Des variable et des éléments de tableaux.
- Des expressions numériques.
- Des expressions alphanumériques.

Le HP-71 évalue et affiche les éléments sur la même ligne (si possible) dans l'ordre de la liste.

Exemples:

```
DISP A; B; C1#; C2#
```

```
DISP S#T/V; L# & N#
```

Le HP-71 formate les nombres selon le format d'affichage courant (les formats d'affichage numériques sont décrits sous le titre «*Formats des nombres*», page 54).

Ordre DISP implicite

Dans la plupart des cas, il n'est pas nécessaire de placer `DISP` devant la *liste d'affichage*. La frappe de `A# [ENDLINE]`, par exemple est équivalente à celle de `DISP A# [ENDLINE]`. Cette caractéristique peut réduire notablement la frappe lors de l'entrée d'un programme. La ligne de programme:

```
10 "VALEUR ="&C1
```

est interprétée et affichée de la même façon que

```
10 DISP "VALEUR ="&C1
```

L'ordre `DISP` implicite ne peut pas être utilisé après `THEN` ou `ELSE` dans une instruction `IF...THEN` ou `IF...THEN...ELSE` car vous pouvez alors utiliser un ordre `GOTO` implicite (consultez la section «*Branchements conditionnels*», page 187). Le HP-71 interprète toute expression alphanumérique suivant `THEN` ou `ELSE` comme un label permettant de transférer l'exécution. Vous devez donc frapper `DISP` après `THEN` ou `ELSE` lorsque vous désirez utiliser l'instruction d'exécution conditionnelle pour afficher des informations.

Lors de l'utilisation de l'ordre `DISP` implicite, veillez à ce que l'expression entrée ne puisse pas être interprétée comme une ligne de programme. Si vous entrez une ligne commençant par un nombre suivi d'un espace, le HP-71 interprète l'expression comme une ligne de programme. Exemple, le HP-71 évalue `1E1` comme la valeur `10` mais interprète `1 E1` comme la ligne de programme `1 DISP E1`.

Espacement des sorties (TAB)

Vous pouvez contrôler l'espacement entre les éléments des listes d'affichage et d'impression en utilisant différents symboles de ponctuation et l'ordre `TAB`.

```
TAB<numéro de colonne>
```

`TAB` opère de façon similaire à la touche de tabulation d'une machine à écrire. Il déplace le curseur ou la tête d'impression à la colonne spécifiée avant d'afficher ou d'imprimer la suite des informations. `TAB(15)`, par exemple, place le curseur en colonne 15 avant d'écrire le bloc d'informations suivant.

Si vous utilisez comme paramètre de `TAB` un nombre supérieur à la largeur courante spécifiée par `WIDTH` ou `PWIDTH`, le HP-71 réduit ce nombre modulo la largeur courante. C'est-à-dire soustrait répétitivement du nombre la largeur courante jusqu'à ce que le nombre soit inférieur à la largeur. Par exemple, si la largeur courante `WIDTH` est 96 (défaut),

```
TAB(116)
```

est interprété égal à

```
TAB(20)
```

Exemple:**Entrée/Résultat**

```
DISP "ABC";TAB(8);"DEF"
```

ABC	DEF
-----	-----

Affiche le premier élément, place le curseur en colonne 8 et affiche le deuxième élément.

Vous pouvez aussi contrôler l'espacement des éléments des listes de sortie en utilisant différents symboles de ponctuation:

- Point-virgule (;) Ne place aucun espace entre les éléments.
- Virgule (,) Remplit le reste de la zone d'affichage (décrite ci-dessous) par des espaces.

Une *zone d'affichage* est une portion de l'affichage de 21 caractères de long. L'affichage accepte une zone d'affichage partielle lorsque sa largeur n'autorise pas une zone complète. L'affichage par défaut (96 caractères) contient donc cinq zones: quatre zones de 21 caractères et une de 12. Si vous séparez deux éléments d'une liste par une virgule, le HP-71 affiche le premier et complète la zone d'affichage par des espaces, si nécessaire, puis il affiche l'élément suivant dans la zone suivante.

Exemples:**Entrée/Résultat**

```
DISP "ABC";"DEF" END LINE
```

ABCDEF

Affiche les deux éléments sans espaces.

```
DISP "ABC", "DEF" END LINE
```

ABC	DEF
-----	-----

Affiche le premier élément en commençant en colonne 1 et le second en commençant en colonne 22.

C	DEF
---	-----

Déroule l'affichage vers la gauche pour afficher le deuxième élément.

Formatage par chaîne de format

L'utilisation des virgules et points-virgules décrite ci-dessus permet de répondre à de nombreuses applications. Néanmoins, dans certaines applications, ces techniques ne vous permettront pas de contrôler efficacement les longueurs des chaînes ou les grandeurs des nombres.

Le HP-71 vous permet de contrôler précisément le format des informations affichées et imprimées grâce à des *chaînes de format* (décrites ci-après).

Exemple: Les deux programmes suivants illustrent l'utilisation des sorties formatées pour arrondir des valeurs à un nombre de décimales spécifié et pour positionner les nombres et les textes de façon à améliorer la lisibilité de vos sorties.

Formatage avec virgules, points-virgules et TAB:

```

10 OPTION BASE 1
20 DESTROY A1,B1,I,A,B
30 FOR I=1 TO 4
40 READ A(I),B(I)
50 DISP A(I);TAB(25);B(I)
60 A1=A1+A(I) @ B1=B1+B(I)
70 NEXT I
80 DISP "-----";TAB(25);"-----"
90 DISP "TOTAL=";A1;TAB(25);"TOTAL=";B1
100 DATA 5.8052,7,.3737,8.6,4.322,9,679.4646,.8

```

Sortie:

5.8052	7
.3737	8.6
4.322	9
679.4646	.8
-----	-----
TOTAL= 689.9655	TOTAL= 25.4

Formatage avec DISP USING et IMAGE (décrits ci-après):

```

10 OPTION BASE 1
20 DESTROY A1,B1,I,A,B
30 FOR I=1 TO 4
40 READ A(I),B(I)
50 DISP USING 100; A(I),B(I)
60 A1=A1+A(I) @ B1=B1+B(I)
70 NEXT I
80 DISP USING 110;"TOTAL=",A1,"TOTAL=",B1
90 DATA 5.8052,7,.3737,8.6,4.322,9,679.4646,.8
100 IMAGE 10X,4D.DD,10X,4D.DD
110 IMAGE 3X,14"-",3X, 14"-"/2(3X,7A,4D.DD)

```

Sortie:

```

          5.81                7.00
           .37                8.60
          4.32                9.00
        679.46                .80
-----
TOTAL= 689.97    TOTAL= 25.40

```

Qu'est-ce qu'une chaîne de format

Une *chaîne de format* est une chaîne de caractères représentant un format de sortie. Une chaîne de format se compose d'un ou plusieurs *spécificateurs de champ* séparés par des *délimiteurs* qui sont généralement des virgules. Chaque spécificateur de champ contrôle le format d'un élément de données. Les spécificateurs se composent de caractères, appelés *symboles d'image* qui définissent le type d'information pour chaque position de caractère, et de *multiplicateurs* qui spécifient le nombre de répétition d'un spécificateur de champ (ou d'un groupe de spécificateurs).

Spécificateur de champ

4D.DD,7X,5D.DD

Chaîne de format

Une chaîne de format peut être une expression alphanumérique:

- Incluse dans un ordre DISP USING ou PRINT USING.

```
DISP USING "4D.DD,7X,5D.DD"; A1,B1
```

- Affectée à une variable alphanumérique et référencée par le nom de cette variable.

```
100 A#="40.00,7%,50.00"
110 DISP USING A#; A1, B1
```

Une chaîne de format peut aussi être une *chaîne sans guillemets* dans un ordre IMAGE référencé par un ordre DISP USING ou PRINT USING.

```
100 DISP USING 110; A1, B1
110 IMAGE 40.00,7%,50.00
```

Les symboles d'image sont listés et décrits dans le dictionnaire des mots-clés du *Manuel de référence* sous le titre IMAGE.

Utilisation d'une chaîne de format (IMAGE, DISP USING, PRINT USING)

Les ordres DISP USING et PRINT USING utilisent les chaînes de format pour formater les sorties.

— syntaxe simplifiée —

```
DISP USING chaîne de format ; liste d'affichage
DISP USING numéro de ligne ; liste d'affichage
```

— syntaxe simplifiée —

```
PRINT USING chaîne de format ; liste d'impression
PRINT USING numéro de ligne ; liste d'impression
```

Chaque élément de la *liste d'affichage* ou *d'impression* utilise un spécificateur de champ de la *chaîne de format*. S'il y a plus d'éléments de données que de spécificateurs, le HP-71 réutilise la chaîne à partir du début jusqu'à ce que tous les éléments de la liste soient formatés. Cette caractéristique vous permet par exemple de créer rapidement un court format pour afficher une douzaine de valeurs numériques utilisant toutes le même format; la chaîne de format peut alors contenir un seul spécificateur. Chaque spécificateur de champ doit correspondre en type à l'élément correspondant de la liste d'affichage ou d'impression. Un spécificateur de champ numérique par exemple doit correspondre uniquement à un élément de donnée numérique sinon le HP-71 génère une condition d'erreur.

Comme indiqué ci-dessus, les chaînes de format peuvent se trouver dans les ordres DISP USING et PRINT USING ou être affectées à une variable alphanumérique. Vous pouvez aussi placer les chaînes de format sur une ligne séparée en utilisant l'ordre IMAGE.

— syntaxe simplifiée —

IMAGE chaîne de format sans guillemets

L'ordre IMAGE vous permet d'utiliser la même chaîne de format pour plusieurs ordres DISP USING et PRINT USING et de ne l'entrer qu'une seule fois.

Exemple: En utilisant les informations sur les chaînes de format données dans le *Manuel de référence*, construisez une chaîne de format pour une liste d'éléments à imprimer sous la forme d'un tableau de quatre lignes et deux colonnes. Utilisez Valeur 1 et Valeur 2 comme titre des colonnes et placez six espaces entre les colonnes.

10 OPTION BASE 1	Définit la limite basse des tableaux égale à 1.
20 PRINT USING 100; "Valeur 1","Valeur 2"	Imprime les titres des colonnes.
30 FOR I=1 TO 4	Commence la boucle.
40 READ A(I),B(I)	Affecte des valeurs aux éléments du tableau en utilisant l'ordre DATA.
50 PRINT USING 110; A(I),B(I)	Imprime les valeurs.
60 NEXT I	Fin de la boucle.
100 IMAGE 2(6X,8A)/	Chaîne de format pour les titres.
110 IMAGE 2(6X,8D)	Chaîne de format pour les valeurs.
120 DATA 14857,233649,122990,333125	
130 DATA 759982,1243,233219,416627	

De façon à déterminer les spécificateurs à utiliser, vous devez d'abord savoir comment vous voulez présenter les informations. Si vous voulez obtenir le résultat suivant, par exemple:

6 espaces		6 espaces
{	Valeur 1	{
	14857	233649
	122990	333125
	759982	1243
	233219	416627
	}	}
	8 chiffres	8 chiffres

La chaîne de format des titres utilise six espaces, huit positions de caractères, six espaces à nouveau puis huit caractères et enfin un symbole de fin-de-ligne pour donner une ligne blanche. Le symbole d'espace est X, le symbole de position de caractère est A et le symbole de fin-de-ligne est /. En utilisant uniquement ces symboles, l'instruction 100 est:

```
100 IMAGE XXXXXX,AAAAAAAA,XXXXXX,AAAAAAAA/
```

Chaque jeu de symboles d'image est appelé *spécificateur de champ*. Un spécificateur de champ définit la façon dont un élément de la liste sera formaté ou définit l'espacement entre les lignes. Les six X spécifient un champ de six espaces. Les huit A spécifient un champ de huit caractères alphanumériques. Les spécificateurs de champ sont généralement séparés par des virgules.

Vous pouvez raccourcir une chaîne de format en utilisant des *multiplicateurs*. Si vous placez un multiplicateur devant un spécificateur de champ, le HP-71 réutilisera ce spécificateur le nombre de fois indiqué avant d'utiliser le spécificateur suivant. L'ordre IMAGE de la ligne 100 devient alors:

```
100 IMAGE 6X,8A,6X,8A/
```

ou plus simplement,

```
100 IMAGE 2(6X,8A)/
```

La chaîne de format pour les données numériques demande six espaces, six chiffres, six espaces à nouveau et six chiffres. L'instruction de la ligne 110 devient:

```
110 IMAGE 2(6X,8D)
```

Contrôle de l'affichage et de l'imprimante

Longueur de ligne (WIDTH, PWIDTH)

Les ordres WIDTH et PWIDTH définissent la longueur de ligne maximum pour les sorties vers l'affichage et vers l'imprimante respectivement.

WIDTH *longueur de ligne*

PWIDTH *longueur de ligne*

Le paramètre *longueur de ligne* peut être une expression numérique quelconque dont la valeur se trouve entre 0 et 96. Ces ordres ne modifient pas le nombre de caractères que vous pouvez frapper à l'affichage. Ils déterminent uniquement la longueur de ligne pour les informations envoyées par les ordres suivants:

- DISP
- DISP USING
- PRINT
- PRINT USING
- LIST
- PLIST

Exemple:

Entrée/Résultat

```
WIDTH 10 [ENDLINE]
"VINGT-TROIS CARACTERES" [ENDLINE]
```

VINGT-TROI

le HP-71 affiche les dix premiers caractères.

S CARACTER

Le HP-71 affiche les dix caractères suivants.

ES

Le HP-71 affiche les derniers caractères.

L'affichage du HP-71 est le périphérique d'impression par défaut s'il n'y a pas d'imprimante connectée à l'ordinateur. Vous pouvez alors afficher des informations en utilisant `DISP` et `PRINT`. De ce cas, si vous exécutez `PWIDTH`, cet ordre n'affecte que la longueur des lignes affichées par `PRINT`. De la même façon, `WIDTH` n'affecte que la longueur des lignes affichées par `DISP`.

Modification de la séquence de fin-de-ligne (ENDLINE)

Lorsque l'ordinateur envoie une ligne d'informations à une imprimante (ou à l'affichage) en utilisant PRINT, il la fait suivre d'une *séquence de fin-de-ligne*. Une séquence de fin-de-ligne est une chaîne de trois caractères maximum qui indique à l'imprimante ce qu'elle doit faire après avoir reçu une ligne. La séquence de fin-de-ligne par défaut contient deux caractères, retour chariot (CR) et avance ligne (LF) et indique à l'imprimante qu'elle doit avancer à la ligne suivante et placer la tête d'impression sur la première colonne. Sur le HP-71, cette séquence par défaut donne l'affichage d'une nouvelle ligne.

Vous pouvez modifier la séquence de fin-de-ligne pour lui affecter tout caractère de votre choix en utilisant l'ordre ENDLINE.

```
ENDLINE chaîne
```

Le paramètre *chaîne* est une expression alphanumérique dont le résultat doit contenir un maximum de trois caractères.

Exemple:

```
ENDLINE CHR#(13)&CHR#(10)
&CHR#(10)
```

Définit la séquence CR, LF, LF comme séquence de fin-de-ligne, de façon à obtenir un double interligne.

Ces deux caractères, CHR#(10) et CHR#(13) sont deux des 32 *caractères de contrôle* ASCII (codes décimaux compris entre 0 et 31) que le HP-71 utilise pour contrôler les périphériques *. De nombreux périphériques répondent différemment aux caractères de contrôle ou n'y répondent pas du tout. Avant d'utiliser des caractères de contrôle, consultez le *Manuel d'utilisation* du périphérique auquel vous voulez envoyer ces codes de façon à déterminer ses réponses.

Contrôle du curseur et de l'affichage

Le HP-71 vous permet de contrôler l'affichage en utilisant certains caractères. Les informations qui suivent ne s'adressent qu'aux utilisateurs ayant déjà une bonne connaissance de la programmation.

Caractères de contrôle et séquences d'échappement. Vous pouvez contrôler plus précisément la façon dont le HP-71 affiche les informations en utilisant des *caractères de contrôle* et des *séquences d'échappement*. Bien que les effets des caractères de contrôle et des séquences d'échappement puissent être obtenus par l'utilisation de fonctions et d'ordres de l'ordinateur (tels que WIDTH et ENDLINE), vous pouvez développer des techniques plus performantes en utilisant des caractères spéciaux.

* Les codes de caractères sont listés sous le titre «Jeu et codes de caractères du HP-71» dans le *Manuel de référence*.

Les caractères de contrôle (générés par des séquences de touches particulières indiquées ci-après) contrôlent la façon dont le HP-71 affiche les informations. Les ordinateurs utilisent couramment ces caractères pour contrôler les communications entre-eux. Les ordinateurs n'affichent pas les caractères de contrôle mais les reconnaissent comme des instructions. Pour accéder à ces caractères, vous devez d'abord appuyer sur **[9] [CTRL]** puis sur une autre touche. Le tableau suivant liste les caractères de contrôle que le HP-71 reconnaît et les séquences de touches nécessaires pour les utiliser.

Caractères de contrôle d'affichage

Caractère ASCII	Séquences de touches	Description
CHR#(8)	[9] [CTRL] [H]	Espace arrière (BS). Déplace le curseur d'un espace vers la gauche.
CHR#(10)	aucune	Avance ligne (LF). Affiche une nouvelle ligne.
CHR#(13)	[9] [CTRL] [M]	Retour chariot (CR). Place le curseur sur la première colonne. Le HP-71 essaie d'évaluer la ligne d'affichage si vous appuyez sur [END LINE] .
CHR#(27)	[9] [CTRL] [9] [I]	Echappement (ESC). Indique le début d'une séquence d'échappement.

Une *séquence d'échappement* est une chaîne, commençant par le caractère d'échappement (code décimal ASCII 27), qui représente une ou plusieurs instructions pour un périphérique tel qu'un affichage ou une imprimante (tout au long de ce manuel, nous représenterons le caractère d'échappement par ESC). L'affichage du HP-71 reconnaît plusieurs séquences d'échappement et ignore celles qu'il ne connaît pas.

Vous pouvez générer les séquences d'échappement de deux façons:

- Comme séquences de touches exécutées directement au clavier.
- Comme chaînes alphanumériques exécutées lors de leur «affichage».

Le tableau suivant liste les séquences d'échappement qui contrôlent l'affichage. Pour exécuter une séquence d'échappement directement au clavier, appuyez sur **[9] [CTRL] [9] [I]** puis sur les touches indiquées dans le tableau. Pour stocker une séquence d'échappement comme chaîne de caractères, utilisez **CHR#(27)** pour générer le caractère d'échappement et concaténez un des caractères représentés par les touches dans le tableau.

Séquences d'échappement du HP-71

Séquence ou caractère d'échappement suivi de:	Description
<code>C</code>	Déplace le curseur d'un espace à droite (identique à <code>></code>).
<code>D</code>	Déplace le curseur d'un espace à gauche (identique à <code><</code>).
<code>E</code>	Efface l'affichage, y compris le symbole BASIC.
<code>H</code>	Place le curseur en première colonne, sur le symbole BASIC.
<code>J</code> ou <code>K</code>	Efface l'affichage de la position du curseur à la fin de la ligne (identique à <code>f</code> <code>[-LINE]</code>).
<code>O</code>	Supprime le caractère à la position du curseur et décale tous les caractères suivants d'un espace vers la gauche (identique à <code>f</code> <code>[-CHAR]</code>).
<code>N</code>	Active le curseur d'insertion (identique à <code>f</code> <code>[I/R]</code>).
<code>R</code>	Active le curseur de remplacement (identique à <code>f</code> <code>[I/R]</code>).
<code>g</code> <code><</code>	Eteint le curseur.
<code>g</code> <code>></code>	Allume le curseur.
<code>%</code> <i>mn</i>	Place le curseur à la colonne représentée par le code de caractère ASCII de <i>m</i> . Le HP-71 utilise le code de caractère ASCII de <i>n</i> comme position de rang pour les affichages multi-lignes. Bien que ce paramètre n'ait aucun effet sur l'affichage du HP-71, il doit être présent pour compléter la séquence d'échappement.

Exemple: Activez le curseur d'insertion puis revenez au curseur de remplacement.

Entrée/Résultat

`g` `CTRL` `g` `[` `N`

Active le curseur d'insertion (flèche clignotante vers la gauche).

`g` `CTRL` `g` `[` `R`

Active le curseur de remplacement (rectangle plein clignotant).

Positionnement du curseur. Vous pouvez placer le curseur sur une colonne quelconque en appuyant sur

`g` `CTRL` `g` `[` `g` `%`

suivi du caractère dont le code représente le numéro de la colonne (dans la suite de ce manuel, nous ferons référence à cette séquence par l'expression *instruction ESC%*).

Exemple: Déplacez le curseur en colonne 65.

Entrée/Résultat

`g CTRL g I g % A A`

Positionne le curseur en colonne 65.

Le code de caractère pour un A majuscule est 65*. Le HP-71 déplace par conséquent le curseur en colonne 65 (cette séquence demande un second caractère après `%`. Ce caractère sert pour d'autres périphériques à spécifier le rang où le curseur doit aller. Souvenez-vous que bien que l'affichage du HP-71 n'utilise pas ce second caractère, vous devez le spécifier). Si vous utilisez un affichage multi-ligne, cette instruction positionne le curseur à la position ligne-colonne spécifiée.

Plutôt que d'utiliser les séquences de touches pour exécuter les séquences d'échappement, vous pouvez stocker ces séquences dans des chaînes de caractères et les utiliser dans des programmes (ou affecter les séquences à des touches du clavier Utilisateur).

Exemple: Affectez une instruction `ESC%` à la variable `A#` de façon à pouvoir déplacer le curseur en colonne 11 avant d'afficher des informations.

Entrée/Résultat

`A# = CHR$(27) & "% " & CHR$(11) & "A"`

Affecte une instruction `ESC%` à `A#`.

`END LINE`

`A# & "HP-71" END LINE`

Evalue la ligne d'affichage.

HP-71

Affiche `HP-71` en commençant en colonne 11.

Création de champs protégés. Les instructions `ESC<` (curseur éteint) et `ESC>` (curseur allumé) sont particulièrement utiles dans des programmes pour écrire des informations dans des champs protégés. On appelle *champ protégé* une partie de l'affichage dont le contenu ne peut pas être modifié au clavier. Le HP-71 définit comme champ protégé toute information écrite alors que le curseur est éteint. Les informations écrites lorsque le curseur est allumé ne sont pas protégées et peuvent être modifiées ou remplacées.

L'exécution de `ESC<` éteint le curseur pour vous permettre d'écrire des informations protégées (l'instruction `ESC>` rallume le curseur). Vous ne pouvez pas placer le curseur dans un champ protégé sauf si vous utilisez l'instruction `ESC%` mais vous pouvez effacer le contenu des champs protégés en exécutant `ON` ou `ENDLINE`. Les champs protégés créés par `WINDOW`, décrits au chapitre 7, ne peuvent être modifiés en aucune façon. Le HP-71 ne peut en aucun cas lire d'informations dans un champ protégé.

* Les caractères du HP-71 et les codes correspondants sont listés dans le *Manuel de référence* du HP-71.

Exemple: Le programme suivant demande à un utilisateur d'entrer un numéro de téléphone. En utilisant les instructions ESC< (curseur éteint) et ESC> (curseur allumé), le message apparaît de la façon suivante:

```
Tel?■ - - .
```

Lorsque l'utilisateur entre le numéro, le curseur saute automatiquement les tirets (le HP-71 reconnaît le nombre affiché comme un nombre sans ponctuation). Le programme suivant affiche le message avec un champ protégé:

10 DIM P\$[40],F\$[2],N\$[2],S1	Dimensionne quatre variables.
20 F\$=CHR\$(27)&"<"	Affecte l'instruction ESC< à F#.
30 N\$=CHR\$(27)&">"	Affecte l'instruction ESC> à N#.
40 H\$=F\$&"-"&N\$	Affecte un tiret protégé à H#.
50 P\$=" "&H\$&" "&H\$&" "&F\$&"."	Construit message F#.
60 INPUT "Tel?",P\$;S1	Demande à l'utilisateur d'entrer un numéro de téléphone (l'ordre INPUT est décrit dans la section « <i>Entrée d'un ou plusieurs éléments</i> » en page 241).

Dans cette routine,

- F# contient la séquence d'échappement qui éteint le curseur.
- N# contient la séquence d'échappement qui allume le curseur.
- H# contient un tiret protégé.
- P# contient le message par défaut pour l'ordre INPUT.

Les informations à afficher dans un champ protégé sont précédées de F# pour éteindre le curseur. Les espaces qui doivent être remplacés par les chiffres du numéro sont précédés de N# pour allumer le curseur. L'ordre INPUT de la ligne 60 affiche le message et attend que l'utilisateur entre un nombre.

Pour utiliser la routine, entrez-la dans un nouveau fichier de programme (consultez la page 143 si vous ne vous souvenez plus comment créer un fichier de programme) et exécutez-la.

Entrée/Résultat

STD

Active le format d'affichage standard.

RUN

```
Te1?■ - -
```

Le HP-71 vous demande un nombre. Remarquez la position du curseur, juste à droite du point d'interrogation.

0778

```
Te1?077-8■-
```

Remarquez comment le curseur saute les tirets.

383

```
Te1?077-83-83■
```

END LINE

Le programme accepte l'entrée.

31 END LINE

Vérifiez la valeur entrée.

```
9077825
```

Le HP-71 n'a pas lu les tirets dans l'entrée.

Vous pouvez créer un champ protégé aux deux extrémités de l'affichage en utilisant l'ordre WINDOW.

Exemple: Créez une fenêtre d'affichage encadrée par HF à gauche et 71 à droite.

Entrée/Résultat

```
"HF";TAB(21);"71"@WINDOW3,20
```

END LINE

```
HF                                71
```

Affiche HF et 71.

END LINE

```
HP>■                                71
```

Place le symbole BASIC en colonne 3. les caractères HF et 71 se trouvent maintenant dans des champs protégés.

Stockage et rappel de données

Table des matières

Présentation	240
Entrée de données au clavier	241
Entrée d'un ou plusieurs éléments (INPUT)	241
Entrée d'une ligne d'affichage (LINPUT)	244
Données de programme	245
Stockage de données dans un programme (DATA)	245
Rappel de données de programme (READ)	246
Positionnement du pointeur (RESTORE)	246
Fichiers de données	247
Types de fichiers de données	247
Création d'un fichier de données (CREATE)	248
Ouverture d'un fichier de données (ASSIGN#)	248
Fermeture d'un fichier de données	249
Accès aux fichiers de données	249
Accès séquentiel	250
Stockage séquentiel (PRINT #)	250
Pointeur et accès séquentiel	252
Rappel séquentiel (RESTORE #, READ#)	254
Accès sélectif	256
Enregistrements de fichier	256
Déplacement du pointeur (RESTORE #)	258
Stockage sélectif (PRINT #)	258
Rappel sélectif (READ #)	260
Stockage et rappel de tableaux	261
Transfert de canaux à un sous-programme	263

Présentation

Les programmes opèrent sur des données qu'ils peuvent obtenir de plusieurs sources:

- Le clavier. Un utilisateur peut entrer des données au clavier lorsqu'un programme le lui demande.

- Un programme. Un programme peut contenir des données stockées dans les lignes.
- Un fichier. Vous pouvez stocker des données dans des fichiers pour utilisation par un programme.

Le présent chapitre décrit comment le HP-71 utilise ces différentes sources pour obtenir des données et plus précisément:

- Comment un programme accepte des données au clavier.
- Comment stocker et rappeler des données sur des lignes de programme.
- Comment créer des fichiers de données, comment y stocker des données et comment rappeler ces données.

Entrée de données au clavier

Entrée d'un ou plusieurs éléments (INPUT)

L'ordre INPUT permet à un programme d'affecter à des variables des valeurs entrées au clavier.

```
INPUT [message [, chaîne par défaut] ;] liste de variables
```

L'ordre INPUT se compose de quatre parties: le mot-clé INPUT, un *message*, une *chaîne par défaut* et une *liste de variables*. Le message et la chaîne par défaut sont optionnels mais vous devez spécifier une liste de variables. Le diagramme suivant illustre les différents composants d'un ordre INPUT.

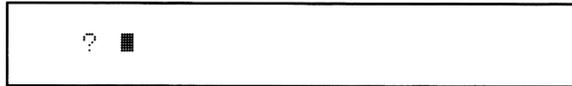
	Message	Chaîne par défaut	Liste de variables
INPUT	"NOM, AGE: "	,"XAVIER, 28";	N#, A

Entrée de valeurs. Les valeurs frappées au clavier en réponse à un ordre INPUT doivent être entrées dans l'ordre des variables correspondantes dans la liste. Les valeurs entrées doivent correspondre en type avec les variables et doivent être séparées par des virgules. L'instruction suivante, par exemple, demande l'entrée d'une valeur numérique, d'une virgule, d'une autre valeur numérique, d'une autre virgule et enfin d'une expression alphanumérique.

Entrée/Résultat

INPUT A,B,C# [END LINE]

Le HP-71 acceptera deux valeurs numériques et une chaîne.



Le HP-71 vous demande les données.

5,3,ABC [END LINE]

Entrez les informations.

Si vous aviez frappé

ABC,5,3

par exemple, le HP-71 aurait affiché un message d'erreur car il refuse d'entrer l'expression alphanumérique ABC dans la variable numérique A.

Si vous aviez frappé

5,3,456

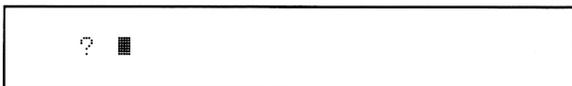
le HP-71 n'aurait pas affiché de message d'erreur car 456 peut être affecté comme nombre à une variable numérique ou comme chaîne à une variable alphanumérique. Dans cet exemple, 456 serait affecté à une variable alphanumérique et considéré comme une chaîne.

Entrée d'expressions. Vous pouvez aussi entrer des expressions numériques ou alphanumériques en réponse à un message INPUT. Chaque expression doit être du type de la variable correspondante dans la liste de INPUT.

Exemple:

Entrée/Résultat

INPUT A,B,C# [END LINE]



4*5+3, D1+SIN(D2), S#&T# [END LINE]

Entrez des expressions numériques et alphanumérique.

Messages d'entrée. Si vous utilisez un ordre `INPUT` avec uniquement une liste de variables, le HP-71 affiche le message `?` lors de l'exécution. Vous pouvez remplacer ce point d'interrogation par le texte de votre choix en plaçant un message entre guillemets (vous ne pouvez utiliser qu'une chaîne entre guillemets) après le mot-clé `INPUT` suivi d'un point-virgule et de la liste de variables. Si vous voulez afficher le message `ENTRE #`, par exemple, l'instruction est:

```
INPUT "ENTREZ #";N
```

Lors de l'exécution de cette instruction, le HP-71 affiche le message et place le curseur à sa droite. Vous ne pouvez pas modifier ce message à partir du clavier lorsqu'il est affiché (sauf en utilisant des séquences d'échappement spéciales).

Exemple:

Entrée/Résultat

```
INPUT "ENTREZ #";N
```

```
ENTREZ #■
```

Affiche le message choisi.

```
S
```

Entrez une valeur.

Chaînes par défaut. Dans certaines applications, il peut être nécessaire qu'une variable prenne une valeur par défaut lorsque l'utilisateur n'en spécifie pas. Vous pouvez pour cela placer une *chaîne par défaut* après le message. Les chaînes par défaut sont couramment utilisées lorsqu'un message appelle souvent la même réponse.

La *chaîne par défaut* doit apparaître sous forme d'une expression alphanumérique suivant le message, séparée de ce dernier par une virgule et de la liste de variables par un point-virgule.

Exemple:

Entrée/Résultat

```
INPUT "ENTREZ #", "45";N END LINE
```

Demande une valeur.

```
ENTREZ #45
```

Le HP-71 place le curseur sur le premier caractère de la chaîne par défaut.

```
END LINE
```

Le HP-71 affecte à la variable `N` la valeur de la chaîne par défaut.

Vous pouvez, bien entendu, changer la valeur affichée avant d'appuyer sur END LINE pour entrer une autre réponse et vous pouvez même utiliser la pile d'instructions pour rechercher et entrer une expression précédente.

Conditions d'entrée. Lorsque le HP-71 affiche un message d'entrée, les conditions suivantes sont actives:

- La pile d'instructions est active et vous permet de rechercher et d'utiliser des expressions précédentes.
- Les touches `VIEW` et `gERRM` sont actives et vous permettent de consulter les affectations de touche et le dernier message d'erreur.
- La pression de `ATTN` efface la ligne d'entrée. Une deuxième pression de `ATTN` interrompt l'exécution du programme.
- La pression de `fCONT` relance l'exécution du programme sans changer les valeurs des variables de la liste.
- La pression de `RUN`, `fSST` ou `fCALC` est identique à celle de `ENDLINE`.
- Vous pouvez frapper des informations avant l'affichage des messages correspondants si vous pouvez les anticiper. Le HP-71 stocke les séquences de touches dans le tampon du clavier et les accepte lorsque le message apparaît. Le tampon de clavier peut contenir un maximum de 15 touches précédées ou non de `f` ou `g`.
- Une condition `ON TIMER #` arrivant à expiration ne provoque pas de branchement tant que l'entrée n'est pas terminée.
- Si une condition `ON ERROR` est active, le HP-71 effectue le branchement si l'entrée est erronée.
- Si vous entrez un élément de type incorrect et si il n'y a pas de condition `ON ERROR` active, le HP-71 ré-affiche le message et attend le type de donnée correcte. Si le HP-71 rencontre une condition d'erreur lors de l'évaluation des éléments entrés, il interrompt le programme et affiche un message d'erreur. S'il rencontre une condition d'avertissement, il substitue une valeur par défaut et continue l'exécution.

Entrée d'une ligne d'affichage (LINPUT)

L'ordre `LINPUT` est similaire à l'ordre `INPUT` mais `LINPUT` affecte le contenu de l'affichage à une seule variable alphanumérique.

```
LINPUT [message [, chaîne par défaut] :] variable
```

L'ordre `LINPUT` peut utiliser un *message* et une *chaîne par défaut* spécifiés par l'utilisateur de la même façon que `INPUT` mais la liste de variables est remplacée par une variable alphanumérique unique. `LINPUT` affecte le contenu entier de la ligne d'affichage à la variable alphanumérique spécifiée. De ce fait vous pouvez utiliser dans l'entrée des virgules qui normalement séparent différents éléments.

Exemple:

Entrée/Résultat

```
LINPUT "TEXTE:"; A# END LINE
```

```
TEXTE: █
```

Demande une entrée.

```
TEXTE:ABC,DEF,GHI END LINE
```

Frappe ABC,DEF,GHI comme ligne d'entrée.

```
A# END LINE
```

Affiche la ligne entrée.

```
ABC,DEF,GHI
```

Montre que le HP-71 a accepté les virgules comme caractères de la chaîne entrée.

Données de programme

Un programme nécessite souvent des données constantes lors de chaque exécution. Vous pouvez stocker ces données directement dans les lignes de programme avec l'ordre DATA et les utiliser avec l'ordre READ.

Stockage de données dans un programme (DATA)

Vous pouvez stocker des données dans un programme en utilisant l'ordre DATA.

— syntaxe simplifiée —

```
DATA éléments de donnée
```

Les *éléments de donnée* sont des expressions numériques et alphanumériques qui peuvent apparaître dans un ordre quelconque mais doivent être séparées par des virgules. Un ordre DATA n'est pas «exécuté» par un programme, il contient simplement des données.

Exemples:

```
DATA 55.4,79,REVENU,INTERET,45E18
```

```
DATA TOTAL,PRIX,(A1+B1+C1)/5,S$(18)&"BAS"
```

Un programme ou un sous-programme ne peut pas utiliser un ordre DATA d'un autre programme ou sous-programme.

Rappel de données de programme (READ)

Le programme accède aux données stockées dans les ordres DATA en les affectant à des variables avec l'ordre READ.

— syntaxe simplifiée —

```
READ liste de variables
```

L'ordinateur conserve un *pointeur de données* qui indique en permanence l'élément suivant à lire dans un ordre DATA. Lorsqu'un ordre READ affecte une donnée à une variable, le HP-71 avance le pointeur d'un élément. Lorsque le HP-71 lance l'exécution d'un programme, il place le pointeur sur le premier élément du premier ordre DATA. Les ordres READ successifs avancent le pointeur d'élément en élément et d'un ordre DATA au suivant.

Lorsque le HP-71 a lu le dernier élément d'un ordre DATA, il place le pointeur sur le premier élément de l'ordre DATA suivant. Lorsque le pointeur atteint le dernier élément du dernier ordre DATA, l'ordre READ donne une erreur. Les différents ordres DATA d'un environnement local constituent en fait un grand tableau de données (les environnements locaux sont décrits au chapitre 12, «*Sous-programmes et fonctions définies par l'utilisateur*»).

Positionnement du pointeur (RESTORE)

Vous pouvez changer la position du pointeur et le placer sur le premier élément d'un ordre DATA quelconque en utilisant l'ordre RESTORE.

```
RESTORE [ identificateur d'instruction ]
```

Exemples:

```
RESTORE
```

Place le pointeur sur le premier élément du premier ordre DATA du programme ou sous-programme.

```
RESTORE 1000
```

Place le pointeur sur le premier élément de l'ordre DATA de la ligne 1000.

```
RESTORE ELECDATA
```

Place le pointeur sur le premier élément de l'ordres DATA de la ligne portant le label ELECDATA.

RESTORE ne peut déplacer le pointeur que dans l'environnement local. S'il n'y a pas d'ordre DATA sur la ligne spécifiée dans un ordre RESTORE, le HP-71 recherche le premier ordre DATA suivant cette ligne et y place le pointeur. Si le HP-71 ne trouve pas d'ordre DATA dans le fichier, il ignore simplement l'ordre RESTORE .

Fichiers de données

De nombreux programmes génèrent ou utilisent des quantités importantes de données qu'il faut stocker sous une forme logique de façon à pouvoir les utiliser facilement. Pour cela, le HP-71 vous permet de définir et d'utiliser des fichiers de données.

Types de fichiers de données

Le HP-71 vous permet de créer trois types de fichiers de données:

- Les fichiers DATA qui peuvent contenir des données numériques et alphanumériques.
- Les fichiers TEXT qui permettent de transférer des données à d'autres ordinateurs Hewlett-Packard, tels que le HP-75.
- Les fichiers SDATA qui ont le même format que les fichiers de données créés par le HP-41.

Bien que les formats de ces fichiers soient différents, la plupart des opérations sont similaires. Nous décriront donc ces opérations de façon générale et nous vous indiqueront les différences.

Vous pouvez effectuer différents types d'opérations sur les fichiers de données:

- Création d'un fichier de données.
- Ouverture d'un fichier de données.
- Fermeture d'un fichier de données.
- Stockage d'informations dans un fichier de données.
- Rappel d'informations d'un fichier de données.

Création d'un fichier de données (CREATE)

Vous pouvez créer un fichier de données en utilisant l'ordre CREATE.

— syntaxe simplifiée —

```
CREATE type de fichier nom de fichier [: unité]
```

Le *type de fichier* doit être DATA, SDATA ou TEXT. Le *nom de fichier* peut être un nom quelconque valide et l'*unité* peut être la mémoire vive principale (par défaut) ou une mémoire vive indépendante dans un logement.

Vous pouvez en option spécifier la taille des fichiers et pour les fichiers DATA uniquement la taille des enregistrements (les tailles de fichier et d'enregistrement sont décrites sous le titre «*Enregistrements de fichiers*», page 256). Lorsque vous stockez des informations séquentiellement dans un fichier de données situé dans la mémoire de l'ordinateur, la taille du fichier s'accroît automatiquement pour accepter les informations. Il n'est donc pas nécessaire de spécifier de taille lorsque vous créez des fichiers de données en mémoire pour des opérations séquentielles.

Lors de la création d'un fichier de données pour accès sélectif, vous devez spécifier la taille du fichier. Le HP-71 n'augmente pas automatiquement la taille du fichier lorsque vous utilisez l'accès sélectif; les enregistrements spécifiés dans des accès sélectifs doivent déjà exister (lors de la création d'un fichier de données sur une unité de stockage de masse, vous devez spécifier la taille du fichier et celle des enregistrements. L'utilisation des fichiers sur des unités de stockage de masse est décrite dans le *Manuel d'utilisation* de l'interface HP-IL, HP 82401A.

Exemples:

```
CREATE DATA STAT
```

Crée un fichier DATA nommé STAT.

```
CREATE TEXT LABNOTES
```

Crée un fichier TEXT nommé LABNOTES.

```
CREATE SDATA AJUST
```

Crée un fichier SDATA nommé AJUST.

Ouverture d'un fichier de données (ASSIGN #)

Pour accéder à un fichier de données, vous devez d'abord l'ouvrir avec un ordre ASSIGN #.

— syntaxe simplifiée —

```
ASSIGN # numéro de canal TO fichier de données
```

Cet ordre ouvre un fichier de données en lui affectant un canal. Un canal est une zone de mémoire créée par ASSIGN # qui contient les informations de contrôle du fichier pour faciliter le transfert des données entre le clavier ou un programme et le fichier associé au canal. Un numéro de canal doit être un entier entre 01 et 255 et ne peut être associé qu'à un seul fichier. Vous pouvez définir un maximum de 64 associations canal-fichier à un moment donné.

Si le HP-71 ne trouve pas en mémoire le fichier que vous voulez associer à un canal et si vous n'avez pas spécifié d'unité, le HP-71 crée un fichier DATA en mémoire principale et l'associe au canal spécifié.

Exemples:

```
ASSIGN # 1 TO STOCK:PORT(4)
```

Ouvre le fichier STOCK dans le logement 4 et lui associe le canal 1.

```
ASSIGN # 221 TO A#
```

Ouvre le fichier indiqué par A# et lui associe le canal 221.

```
ASSIGN # B*7 TO VOLTAGES
```

Ouvre le fichier VOLTAGES et lui associe le canal spécifié par B*7.

Fermeture d'un fichier de données

Nous vous recommandons de toujours fermer un fichier dès que vous avez fini de l'utiliser. La fermeture d'un fichier ré-incorpore à la mémoire disponible les 34 octets utilisés par le canal. Le HP-71 ferme automatiquement tous les fichiers de données ouverts par un programme ou sous-programme lorsque ce dernier exécute END ou END SUB. Lorsque vous fermez un fichier, vous terminez simplement son association à un canal.

L'ordre ASSIGN # permet de fermer le fichier associé au canal spécifié si vous l'utilisez avec l'une des chaînes suivantes comme nom de fichier:

- ""
- "*"
- *

Exemples:

```
ASSIGN # 1 TO ""
```

```
ASSIGN # 5 TO "*"
```

```
ASSIGN # 12 TO *
```

Accès aux fichiers de données

Vous pouvez accéder aux données d'un fichier de deux façons: *séquentiellement* et *sélectivement*. Lors de l'accès séquentiel, le HP-71 stocke ou lit les éléments les uns après les autres. Lors de l'accès sélectif, le HP-71 stocke ou lit les éléments que vous désignez. L'ordre des éléments dans le fichier n'est pas nécessairement l'ordre chronologique du stockage.

Le HP-71 utilise un *pointeur* pour marquer l'emplacement du prochain élément à stocker ou à lire. Lors de l'accès séquentiel, le HP-71 déplace automatiquement le pointeur à l'élément suivant. Lors de l'accès sélectif, vous devez spécifier la position du pointeur dans le fichier.

Si vous n'avez jamais utilisé de fichiers, nous vous recommandons d'étudier l'accès séquentiel avant l'accès sélectif. La première méthode est beaucoup plus simple car vous n'avez pas à vous soucier de la position du pointeur. Il vous suffit de vous souvenir de remettre le pointeur en début de fichier une fois toutes les données stockées.

Accès séquentiel

Lorsque vous voulez toujours accéder à un ensemble d'éléments dans un ordre donné, vous pouvez utiliser l'accès séquentiel.

L'accès séquentiel présente les avantages suivants:

- Il est plus simple que l'accès sélectif.
- Le HP-71 déplace automatiquement le pointeur.
- Vous pouvez stocker une liste d'éléments sans vous préoccuper des tailles d'enregistrement et de fichier.

Stockage séquentiel (PRINT #)

Une fois un fichier créé et ouvert, vous pouvez y stocker des informations en utilisant l'ordre PRINT #.

— syntaxe simplifiée —

```
PRINT # numéro de canal ; liste d'éléments
```

Les éléments de la liste peuvent être des expressions ou des tableaux numériques ou alphanumériques et doivent être séparés par des virgules (bien que vous puissiez lire des valeurs numériques *et* alphanumériques dans un fichier DATA, vous ne pouvez y stocker que des valeurs numériques). Le HP-71 stocke les éléments dans l'ordre de la liste.

Nota: Lorsque vous stockez des informations numériques dans un fichier DATA, la taille des enregistrements doit être supérieure ou égale à huit octets. Si vous essayez de stocker des données numériques dans des enregistrements de taille inférieure, le HP-71 affiche un message d'erreur.

Exemple: Le programme suivant vous permet de tenir votre compte bancaire à jour. Le programme vous demande des informations concernant chaque chèque écrit et stocke séquentiellement le montant et l'ordre des chèques dans un fichier DATA (nous utiliserons ce fichier dans un autre exemple; nous vous recommandons par conséquent de le conserver). Le programme vous permet d'entrer les informations pour un nombre illimité de chèques. Lorsque vous avez terminé l'introduction, frappez FIN, Ø en réponse au message pour arrêter le programme. Entrez le programme suivant (frappez tout d'abord EDIT CHEQUE END LINE) et exécutez-le.

```

10 CREATE DATA CHEQUES
20 ASSIGN #1 TO CHEQUES
30 DESTROY N$,A
40 INPUT "Ordre, Montant"; N$, A
50 IF N$ = "FIN" THEN FIN
60 PRINT #1; N$, A
70 GOTO 30
80 FIN: PRINT #1; N$

90 ASSIGN #1 TO *

```

Crée le fichier de données CHEQUES.
Ouvre le fichier et l'associe au canal 1.
Libère les variables N\$ et A.
Demande les informations.
Termine le programme lorsque vous entrez FIN.
Stocke les informations.
Retourne à l'instruction d'entrée.
Stocke le mot FIN pour marquer la fin des entrées.
Ferme le fichier en rompant son association avec le canal 1.

Entrée/Résultat

```
RUN CHEQUE END LINE
```

```
Ordre, Montant: █
```

```
Xavier, 1050.75 END LINE
```

```
Ordre, Montant: █
```

```
Constance, 18.95 END LINE
```

```
Ordre, Montant: █
```

Exécutez le programme CHEQUE.

Le programme vous demande l'ordre et le montant du chèque.

Entrez l'ordre et le montant.

Le programme vous demande les informations pour un autre chèque.

GRAPHICART,127.65 [END LINE]

Ordre, Montant : ■

PHOTO CENTRE,34.5 [END LINE]

Ordre, Montant : ■

FIN, 0 [END LINE]

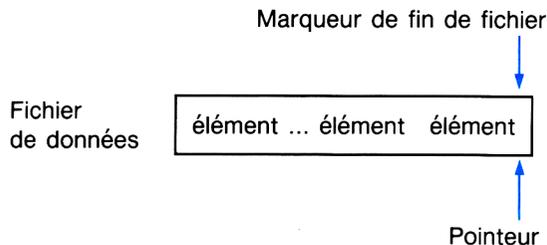
Indiquez au HP-71 que vous avez fini l'entrée des informations.

Dans l'exemple ci-dessus, vous avez stocké différents types de données dans un même fichier. Le programme stocke toutes les informations les unes après les autres, dans l'ordre de leur introduction.

Pointeur et accès séquentiel

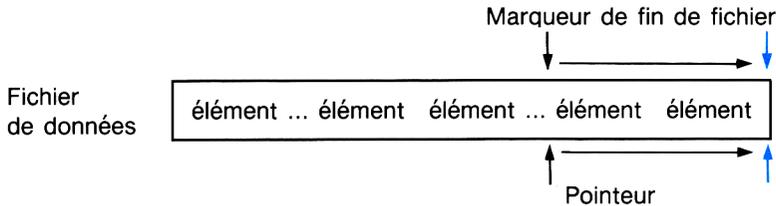
Le HP-71 utilise le *pointeur* pour marquer la position du prochain élément à stocker ou lire. Lorsque le HP-71 accède à un élément du fichier il déplace automatiquement le pointeur à l'élément suivant.

Lors de l'ouverture d'un fichier, le HP-71 place le pointeur au début de ce dernier. Si vous utilisez le stockage séquentiel des informations, le HP-71 stocke les éléments dans l'ordre de la liste d'éléments des ordres PRINT#. Une fois toutes les données de la liste stockées, le pointeur reste à sa position (juste après le dernier élément stocké) et le HP-71 y place un *marqueur de fin de fichier*. Le HP-71 place toujours un marqueur de fin de fichier après le dernier élément d'une liste stockée séquentiellement.



L'exécution d'un autre ordre PRINT # sur le même fichier stocke les nouveaux éléments à la suite des précédents et déplace le marqueur de fin de fichier à la fin de la nouvelle liste. Le HP-71 augmente automatiquement la taille du fichier si nécessaire.

```
PRINT # 1; ELEMENT,ELEMENT,...,ELEMENT
```

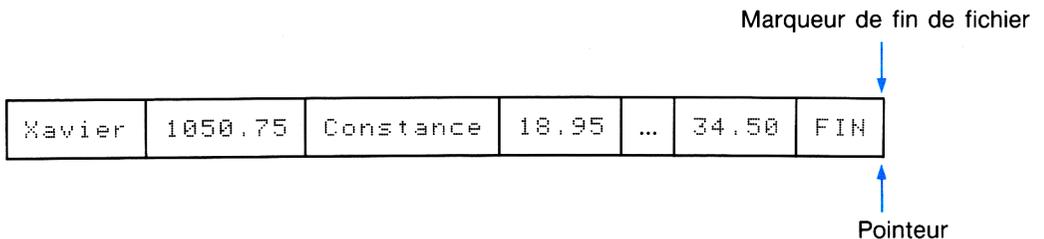


Le pointeur continue à se déplacer séquentiellement dans le fichier pendant l'ajout des éléments et jusqu'à la fermeture du fichier ou au repositionnement du pointeur par RESTORE # (décrit sous le titre «Rappel séquentiel»).

Nota: Lors du stockage séquentiel, vous devez toujours stocker *toutes* les informations dans le fichier avant de repositionner le pointeur ou de fermer le fichier. Les mouvements du pointeur et du marqueur de fin de fichier influencent la façon dont s'effectue la mise à jour des fichiers séquentiels. Si vous fermez un fichier puis le rouvrez ou si vous positionnez le pointeur au début du fichier en utilisant RESTORE #, puis stockez des informations en utilisant PRINT #, les nouvelles données remplacent les anciennes. Même les anciennes données non remplacées sont perdues car le HP-71 ne peut pas accéder aux éléments situés après le marqueur de fin de fichier.

Le diagramme suivant illustre le stockage des informations de l'exemple précédent et la position du pointeur avant la fermeture du fichier.

Fichier DATA pour le programme CHEQUE



Rappel séquentiel (RESTORE #, READ #)

Le HP-71 rappelle les données séquentielles dans l'ordre dans lequel elles sont stockées. Si vous stockez des informations dans un fichier et ne fermez pas ce dernier, vous devrez repositionner le pointeur à l'aide de RESTORE # au début du fichier avant de pouvoir y lire des données.

— syntaxe simplifiée —

```
RESTORE # numéro de canal
```

Exemples:

```
RESTORE # 14
```

Place le pointeur au début du fichier associé au canal 14.

```
RESTORE # A+B
```

Place le pointeur au début du fichier associé au canal indiqué par l'expression A+B.

Si vous venez d'ouvrir un fichier, le pointeur se trouve positionné au début de ce fichier. Il n'est alors pas nécessaire d'utiliser RESTORE # avant de pouvoir lire des données.

L'ordre READ # vous permet de lire des données dans un fichier et de les affecter à des variables.

— syntaxe simplifiée —

```
READ # numéro de canal ; liste de variables
```

Dans cet ordre, le *numéro de canal* doit être un numéro valide et la *liste de variables* doit contenir une ou plusieurs variables (simples ou en tableaux) séparées par des virgules.

L'ordre READ # rappelle des données dans le fichier et les affecte aux variables de la liste. Chaque variable de la liste doit correspondre en type à la donnée qui lui est affectée. Les exécutions successives de READ # lisent des éléments successifs du fichier et les affectent aux variables (cet ordre ne modifie pas le contenu du fichier).

Si vous essayez de rappeler avec READ # des informations situées au delà du marqueur de fin de fichier, le HP-71 affiche un message d'erreur.

Exemple: Le programme suivant rappelle les informations sur les chèques stockées dans le fichier CHEQUES créé dans l'exemple de la page 251 et les affiche ligne par ligne. Pour afficher ces informations, entrez le programme après avoir exécuté EDIT RAPCHEQ [ENDLINE].

```

10 DESTROY N$,A,S
20 ASSIGN #1 TO CHEQUES
30 READ # 1;N$
40 IF N$="FIN" THEN FIN
50 READ # 1;A
60 DISP USING 100;N$,A
70 S=S+A
80 GOTO 30

90 FIN: DISP USING 110; S
100 IMAGE 12A,X,"F",5D.DD

110 IMAGE "TOTAL=",7X,"$",5D.DD

120 ASSIGN # 1 TO *
```

Libère les variables N\$, A et S.
Ouvre le fichier de données.
Lit un ordre.
Teste pour la fin de fichier.
Lit le montant du chèque.
Affiche l'ordre et le montant.
Calcule le total des montants.
Retourne à l'ordre de lecture pour le chèque suivant.
Affiche le montant total des chèques.
Spécifie la chaîne de format pour l'affichage des informations sur les chèques.
Spécifie la chaîne de format pour l'affichage du total.
Ferme le fichier de données.

Entrée/Résultat

```
RUN RAPCHEQ END LINE
```

Exécutez RAPCHEQ.

Xavier	F	1050.75
--------	---	---------

Lit les chèques du fichier de données et les affiche.

Constance	F	18.95
GRAPHICART	F	137.65
PHOTO CENTRE	F	34.50
TOTAL=	F	1241.85

Calcule et affiche le total.

Il n'est pas nécessaire que les précisions des données numériques d'un fichier (REAL, INTEGER, SHORT) correspondent à celles des variables auxquelles elles sont affectées. Le HP-71 convertit les nombres rappelés à la précision des variables auxquelles ils sont affectés. Si une variable de READ # a une précision inférieure à celle de l'élément rappelé, ce dernier est arrondi à la précision de la variable. Si une variable de READ # a une précision supérieure, le HP-71 ne la change pas mais considère la valeur avec la précision de la variable.

Accès sélectif

L'accès sélectif vous permet d'accéder individuellement aux différents *enregistrements* d'un fichier de données. L'accès sélectif est un peu plus complexe que l'accès séquentiel. Si vos applications n'utilisent que l'accès séquentiel, ces informations ne vous sont pas nécessaires.

L'accès sélectif présente les avantages suivants:

- Vous pouvez placer le pointeur sur un enregistrement quelconque. Ceci vous permet de stocker et de rappeler les données dans un ordre quelconque.
- Le HP-71 ne place pas de marqueur de fin de fichier après le dernier élément stocké.

Il y a néanmoins deux restrictions pour l'accès sélectif:

- Vous ne pouvez accéder qu'au contenu d'un seul enregistrement à la fois (mais vous pouvez stocker plusieurs éléments dans un enregistrement).
- Vous ne pouvez pas utiliser le stockage sélectif avec un fichier `TEXT`.

Enregistrements de fichier

Chaque fichier de données se compose d'unités plus petites appelées *enregistrements*. L'accès sélectif vous permet d'adresser individuellement ces enregistrements (dans le cas de l'accès séquentiel, le HP-71 stocke les données sans s'occuper des limites entre enregistrements).

Vous ne pouvez stocker qu'un seul élément dans un enregistrement `SDATA` et cette donnée doit être de type numérique. Par contre, contrairement aux fichiers `DATA`, vous pouvez accéder à plusieurs enregistrements `SDATA` avec un seul ordre `PRINT #` ou `READ #`.

Vous pouvez utiliser le rappel sélectif ou séquentiel avec les fichiers `TEXT` mais vous ne pouvez utiliser que le stockage séquentiel. La taille des enregistrements de fichiers `TEXT` est déterminée par l'ordinateur au moment du stockage des données. Le HP-71 dimensionne chaque enregistrement selon la quantité d'informations à stocker.

Le tableau suivant indique les opérations que vous pouvez effectuer sur chaque type de fichier de données (indiqué par un x):

Opérations permises sur les fichiers de données

Opération	DATA	SDATA	TEXT
PRINT # sélectif	x	x	
READ # sélectif	x	x	x
Accès à un seul enregistrement	x		
Plusieurs éléments par enregistrement	x		

Le tableau ci-dessous donne la taille d'enregistrement pour les différents type de fichiers de données.

Taille des enregistrements des fichiers de données

Type de fichier	Taille d'enregistrement
DATA	256 octets (défaut). Peut être défini par l'utilisateur.
SDATA	8 octets (fixe).
TEXT	Défini par l'ordinateur pour chaque enregistrement selon la quantité d'informations à stocker.

Lorsque vous créez un fichier DATA, vous pouvez spécifier la taille de ses enregistrements et leur nombre.

— syntaxe simplifiée —

```
CREATE type de fichier nom de fichier [:unité] [, taille de fichier [, taille d'enregistrement]]
```

Dans cet ordre, la *taille de fichier* est le nombre d'enregistrements des fichiers DATA et SDATA et le nombre d'octets des fichiers TEXT. La *taille d'enregistrements* est le nombre d'octets par enregistrement pour les fichiers DATA uniquement. Lorsque vous créez un fichier SDATA, vous pouvez spécifier le nombre d'enregistrements mais le HP-71 ignore la taille d'enregistrement si vous l'indiquez (les enregistrements de fichiers SDATA contiennent tous huit octets). Le HP-71 ignore aussi les tailles d'enregistrement pour les fichiers TEXT.

Les fichiers DATA sont les plus souples d'utilisation pour les opérations d'accès sélectif. Vous pouvez spécifier la taille du fichier et celle des enregistrements et vous pouvez y stocker des données de type quelconque.

Déplacement du pointeur (RESTORE #)

Vous pouvez utiliser `RESTORE #` pour déplacer le pointeur à un enregistrement particulier d'un fichier.

```
RESTORE # numéro de canal [, numéro d'enregistrement]
```

Exemple:

```
RESTORE # 1,25
```

Déplace le pointeur qu début de l'enregistrement 25.

La possibilité de déplacer le pointeur s'avère particulièrement utile lorsque vous voulez accéder à des données commençant à un enregistrement spécifique. Pour cela, il suffit de positionner le pointeur sur cet enregistrement à l'aide de `RESTORE #` puis de lire ou de stocker les informations. Vous devez utiliser `RESTORE #` pour positionner le pointeur si vous voulez accéder *séquentiellement* à un groupe de données commençant à un enregistrement spécifié*. Par contre, pour les accès sélectifs, il n'est pas nécessaire d'utiliser `RESTORE #`.

Si vous ne spécifiez pas de *numéro d'enregistrement*, le HP-71 place le pointeur sur le premier enregistrement (numéro 0).

Stockage sélectif (PRINT #)

Vous pouvez stocker des informations dans un enregistrement en spécifiant son numéro dans un ordre `PRINT #` (valable pour les fichiers `DATA` et `SDATA` uniquement).

— syntaxe simplifiée —

```
PRINT # numéro de canal, numéro d'enregistrement; liste d'éléments
```

Nota: Lorsque vous stockez des données numériques dans un fichier `DATA`, la taille des enregistrements du fichier doit être supérieure ou égale à huit octets, sinon une erreur apparaît.

Exemples:

```
PRINT # 1,0;A#
```

Stocke A# dans l'enregistrement 0.

```
PRINT # 1,A+B;T#&G#
```

Stocke une chaîne dans l'enregistrement indiqué par A+B.

L'exemple suivant illustre le mouvement du pointeur pendant des opérations de stockage sélectif. Cet exemple utilise un fichier `DATA` avec des enregistrements de huit octets.

* Les éléments stockés séquentiellement se trouvent souvent dans des enregistrements consécutifs s'il n'y a pas assez d'espace dans un seul enregistrement pour les contenir tous. Le début d'une chaîne, par exemple, peut se trouver dans un enregistrement et la fin dans le suivant. Veillez donc à ne pas altérer par inadvertance des contenus d'enregistrements.

Exemple: Créez un fichier DATA et stockez-y des données numériques.

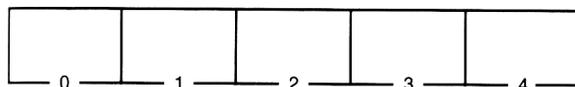
Entrée/Résultat

```
CREATE DATA DATA,5,8
```

Crée un fichier DATA avec des enregistrements de huit octets;

```
ASSIGN # 1 TO DATA
```

Ouvre le fichier DATA.



Le fichier est vide car il vient d'être créé.

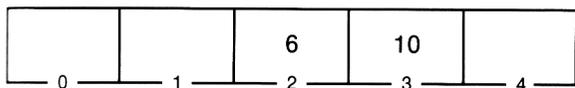
Pointeur

```
PRINT # 1,2;6
```

Stocke 6 dans l'enregistrement 2.

```
PRINT # 1,3;10
```

Stocke 10 dans l'enregistrement 3.



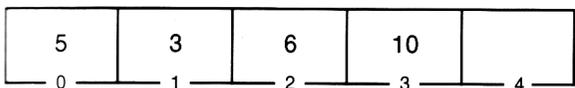
Le pointeur de fichier se trouve au début de l'enregistrement 4.

Pointeur

```
PRINT # 1,0;5
```

Stocke deux éléments supplémentaires. Le HP-71 ne place pas de marqueur de fin de fichier après les stockages sélectifs.

```
PRINT # 1,1;3
```

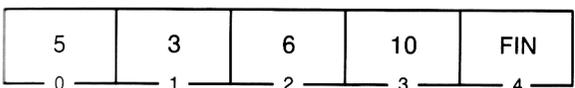


Le pointeur se trouve après le dernier élément stocké.

Pointeur

```
PRINT # 1,4;"FIN"
```

Stocke la chaîne FIN dans l'enregistrement 4.



Le pointeur se place sur l'enregistrement 4 et le HP-71 écrit la chaîne FIN.

Pointeur

Après la dernière opération, le pointeur se trouve positionné dans l'enregistrement 4. Si vous essayez maintenant d'effectuer une opération de stockage sélectif, les nouvelles informations remplacent le mot `FIN`. Par contre, vous pouvez ajouter des informations dans l'enregistrement 4 après la chaîne `FIN` en utilisant l'accès séquentiel. Nous vous recommandons néanmoins d'éviter d'utiliser concouramment les accès sélectif et séquentiel.

Rappel sélectif (`READ #`)

De même que pour le stockage, vous pouvez spécifier le numéro de l'enregistrement dans lequel vous voulez lire des informations. Une opération de lecture sélective ne peut rappeler des informations que d'un seul enregistrement.

— syntaxe simplifiée —

```
READ # numéro de canal, numéro d'enregistrement; liste de variables
```

Vous pouvez stocker et rappeler plusieurs éléments dans un enregistrement de fichier `DATA` mais vous devez placer dans un seul ordre `PRINT #` tous les éléments qui doivent se trouver dans le même enregistrement. Le stockage de plusieurs éléments dans un même enregistrement s'avère très utile dans certaines applications. Si vous écrivez un programme pour stocker une liste de numéros de téléphone, par exemple, vous pouvez placer le nom, le prénom et le numéro de téléphone dans un même enregistrement et rappeler les données concernant une personne à l'aide d'une seule instruction telle que `READ # 1,4;F#,L#,N#`.

Exemples:

```
READ # 2,5;A,B#
```

Rappelle deux éléments de l'enregistrement 5.

```
READ # 5,2*B-1;N#
```

Rappelle une chaîne de l'enregistrement indiqué par l'expression `2*B-1`.

Toute tentative pour lire des informations au delà de la fin du fichier ou d'une limite d'enregistrement donne une erreur.

Les enregistrements des fichiers `TEXT` et `SDATA` ne peuvent contenir qu'un seul élément mais vous pouvez lire les contenus de plusieurs enregistrements en même temps.

Stockage et rappel de tableaux

Vous pouvez stocker dans un fichier de données un tableau numérique ou alphanumérique entier avec un seul ordre PRINT #. Vous pouvez utiliser des parenthèses, comme indiqué ci-dessous, pour spécifier un tableau dans une *liste d'éléments*. Il n'est pas nécessaire de spécifier les dimensions du tableau lorsque vous le stockez dans un fichier de données. Le HP-71 adapte automatiquement la taille du fichier à condition qu'il y ait suffisamment de mémoire vive disponible.

Exemples:

```
PRINT # 4; A()
```

Stocke un tableau unidimensionnel, A dans le fichier de données associé au canal 4.

```
PRINT # 2; T4(,)
```

Stocke le tableau bidimensionnel, T4 dans le fichier de données associé au canal 2 (la virgule entre les parenthèses indique qu'il s'agit d'un tableau bidimensionnel).

```
PRINT # 1; B
```

Stocke le tableau B dans le fichier de données associé au canal 1. Il n'est pas nécessaire d'indiquer les dimensions du tableau.

Le HP-71 stocke les tableaux comme une séquence de données. Il n'y a aucune indication dans le fichier montrant qu'il s'agit d'un tableau. Pour éviter toute confusion, nous vous recommandons de ne placer qu'un seul tableau dans un fichier de données et d'y accéder séquentiellement avec un ordre PRINT # ou READ # unique. Dans certaines applications il est néanmoins nécessaire de rappeler des éléments de tableau dans un ordre différent.

Lorsque vous stockez un tableau, le premier élément du premier rang devient le premier élément du fichier suivi du second élément du premier rang etc. Le dernier élément du premier rang est suivi du premier élément du deuxième rang et ainsi de suite.

La matrice suivante, par exemple, est stockée dans un fichier de données (en supposant que vous l'avez associé au canal 1) en suivant la séquence A(1,1), A(1,2), A(1,3)...A(3,4), comme illustré:

$$A(,) = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

```
PRINT # 1; A
```

Fichier=

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

Les éléments étant stockés linéairement, vous pouvez les rappeler dans un format quelconque. En considérant le tableau de l'exemple ci-dessus, les instructions suivantes permettent toutes de rappeler les éléments du tableau (on suppose le fichier associé au canal 1 et la limite inférieure égale à 1):

```
RESTORE # 1
```

```
DIM B(3,4) @ READ # 1; B(,)
```

Affecte le fichier de données au tableau B de trois rangs et quatre colonnes. Ce tableau est identique au tableau A de l'exemple précédent.

```
RESTORE # 1
```

```
DIM C(4,3) @ READ # 1; C(,)
```

Affecte le fichier de données au tableau C de quatre rangs et trois colonnes.

$$C(,) = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$$

```
RESTORE # 1
```

```
DIM T1(6,2) @ READ # 1; T1(,)
```

Affecte le fichier de données au tableau T1 de six rangs et deux colonnes.

$$T1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

```
RESTORE # 1
```

```
DIM X(12) @ READ # 1; X()
```

Affecte le fichier de données au tableau unidimensionnel X de douze éléments.

$$X() = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

Vous pouvez aussi lire les valeurs du fichier de données dans une liste de variables simples, ou en lire un sous-ensemble dans un tableau plus petit.

```
RESTORE # 1
```

```
READ # 1; D,E,F,G,H,I
```

Affecte les six premiers éléments du fichier aux variables D, E, F, G, H et I.

$$D = 1, E = 2, F = 3, G = 4, H = 5, I = 6$$

Transfert de canaux à un sous-programme

Lorsque vous effectuez la même séquence d'instructions sur plusieurs fichiers de données, il est souvent pratique d'écrire cette séquence sous forme d'un sous-programme. Il suffit alors au programme d'ouvrir chaque fichier de données avec `ASSIGN #` et de transférer le numéro de canal au sous-programme qui doit effectuer les opérations.

Les transferts de numéros de canal au sous-programme sont légèrement différents des transferts de variables ou constantes. Un numéro de canal doit apparaître dans la liste de paramètres réels d'un sous-programme sous la forme d'une constante entière entre 1 et 255 et précédée du symbole `#`. Exemple:

```
SUB NOM1(#5)
```

déclare le canal 5 comme canal local. Ce numéro de canal identifie le même fichier de données que le numéro correspondant dans la liste de paramètres réels de l'ordre `CALL` (les sous-programmes, les transferts de paramètres, les environnements locaux et les environnements d'appel sont décrits au chapitre 12, «*Sous-programmes et fonctions définies par l'utilisateur*»).

Le numéro de canal dans la liste de paramètres réels peut être une expression numérique mais doit toujours être précédé du symbole `#`.

Exemples: Les ordres `CALL` suivants transfèrent un canal à l'ordre `SUB` précédent:

```
CALL NOM1(#2*N)
```

```
CALL NOM1(#8)
```

Le deuxième ordre `CALL` transfère le canal 8 à `NOM1`. le fichier de données associé au canal 8 dans le programme d'appel devient associé au canal 5 dans le sous-programme. Toutes les opérations effectuées sur le canal 5 du sous-programme sont effectuées sur le fichier affecté au canal 8 du programme d'appel. Si le programme d'appel possède aussi un canal 5, ce dernier n'est pas affecté par les numéros de canal du sous-programme. Un sous-programme peut utiliser les mêmes numéros de canal que le programme appelant en les maintenant indépendants.

Si vous ne placez pas de numéro de canal dans la liste de paramètres locaux du sous-programme, les canaux déclarés par le sous-programme sont locaux à ce sous-programme. Si un sous-programme ne possède pas de liste de paramètres locaux, tous les numéros des canaux qu'il utilise sont ceux du programme ou sous-programme appelant.

Le tableau suivant résume les relations des canaux entre l'environnement de sous-programme et l'environnement d'appel.

Relations entre canaux

Forme de l'ordre SUB	Relation
Ordre SUB sans liste de paramètres locaux.	Les numéros de canaux sont ceux du programme appelant.
Ordre SUB avec une liste de paramètres locaux ne contenant pas de numéro de canal.	Les canaux sont locaux au sous-programme.
Ordre SUB avec une liste de paramètres locaux contenant des numéros de canaux.	Les canaux sont locaux au sous-programme. Les canaux de la liste de paramètres locaux sont associés aux mêmes fichiers que les canaux correspondants dans la liste de paramètres de l'ordre CALL.

Annexes et index

Informations générales

Table des matières

Numéro de série et version du système d'exploitation (VER#)	267
Conditions d'environnement	267
Précautions d'utilisation	267
Précision de l'horloge	268
Conformité de l'interpréteur BASIC avec la norme ANSI	268
Extensions au BASIC minimal	268
Déviations du HP-71 par rapport au BASIC minimal	269
Alimentation	271
Consommation	271
Indicateur de baisse de charge	271
Remplacement de la batterie de piles	272
Informations d'entretien	272
Modules enfichables	273
Vérification du fonctionnement	273
Garantie	274
Modifications	274
Informations	275
Maintenance	275
Maintenance en Europe	276
Dans les autres pays	276
Coût de la maintenance	277
Garantie sur les réparations	277
Instructions d'expédition	277
Détérioration de la batterie de piles	277
Assistance technique	278

Numéro de série et version du système d'exploitation (VER\$)

Chaque HP-71 possède un numéro de série gravé sur sa face inférieure. Nous vous recommandons de noter ce numéro et de le conserver au cas où votre ordinateur serait perdu. Hewlett-Packard ne garde pas la liste des numéros de série et des clients.

La fonction VER\$ donne une chaîne de neuf caractères* indiquant la version du système d'exploitation de votre ordinateur. Frappez VER\$ [END LINE]. Veuillez nous communiquer ce numéro lorsque vous nous contactez pour des conseils techniques.

Conditions d'environnement

Respectez les limites suivantes pour l'environnement de votre ordinateur.

- Température de fonctionnement: 0° à 45°C.
- Température de stockage: -40° à 55°C.
- Humidité: 0 à 95% humidité relative.

Vous ne devez jamais utiliser ou stocker votre ordinateur en dehors de ces limites. Les hautes températures peuvent endommager gravement l'ordinateur et la batterie.

Précautions d'utilisation

Certains circuits du HP-71 opèrent en permanence et sont par conséquent toujours susceptibles aux modifications d'environnement. Vous pouvez endommager l'ordinateur par:

- l'installation ou le retrait de la batterie lorsque le chargeur est connecté (peut provoquer la perte de la mémoire).
- l'installation ou le retrait d'un module lorsque le chargeur est connecté (peut provoquer la perte de la mémoire).
- des décharges électrostatiques.
- de forts champs magnétiques.
- la connexion d'équipements non approuvé par Hewlett-Packard.

* Si vous avez enfilé une ou plusieurs ROM, la chaîne peut comporter plus de neuf caractères.

Veillez à observer les précautions indiquées ci-après.

ATTENTION

- Tenez ou touchez en permanence l'ordinateur lors de l'installation de la batterie ou d'un module de façon à neutraliser les charges électrostatiques. Ceci est particulièrement important pour le module HP-IL et les logements de lecteur de carte.
- Ne placez pas vos doigts, des outils ou tout autre objet dans les logements des modules.
- Eteignez l'ordinateur avant de connecter ou de retirer la batterie ou un module.

Précision de l'horloge

L'horloge du système est régulée par un quartz ayant une précision de trois minutes par mois dans les pires conditions de températures. La précision typique est de 1,5 minutes par mois. La procédure d'ajustement (présentée au chapitre 5) permet d'obtenir une précision de 15 secondes par mois. Le quartz de l'ordinateur est affecté par la température, les chocs, l'humidité et l'âge. Vous obtiendrez la précision optimale à $25^{\circ}\text{C} \pm 5^{\circ}$. Il peut être nécessaire d'ajuster l'horloge après un brusque changement d'environnement (voir chapitre 5, page 94).

Conformité de l'interpréteur BASIC avec la norme ANSI

L'interpréteur BASIC du HP-71 est conforme à la définition du BASIC minimal de l'ANSI (American National Standards Institute) sauf dans les cas indiqués ci-dessous. La conformité à la norme a été vérifiée par les tests du National Bureau of Standards (NBS). Ces tests sont disponibles dans NBS Special Publications 500-70/1 et 500-70/2 au National Bureau of Standards, U.S. Department of Commerce, Washington, D.C., 20234, U.S.A.

Extensions au BASIC minimal

Le HP-71 dépasse le BASIC minimal dans les cas suivants (les numéros entre parenthèses font référence aux numéros de programme des tests NBS):

- Les variables numériques et alphanumériques sont initialisées à 0 et à la chaîne vide respectivement. Leur référence avant affectation donne la valeur par défaut 0 pour les variables numériques et la chaîne vide pour les variables alphanumériques. (23)
- Le jeu de caractères comprend tous les caractères ASCII de code décimal compris entre 0 et 255. (93.1, 102, 112)

- Le HP-71 accepte les guillemets ou les apostrophes dans les chaînes entrées si le symbole n'est pas déjà le premier caractère de la réponse. (109)
- Dans les entrées de programme, le HP-71 accepte des blancs en début de ligne et l'absence de blancs entre les mots-clés. Une fois la ligne entrée, l'interpréteur supprime les blancs inutiles et insère des blancs lorsque nécessaire pour la lisibilité. (187, 190, 191)
- Dans une affectation, le mot-clé LET est optionnel. (185)
- L'interpréteur place un ordre END invisible à la fin physique d'un fichier BASIC, l'utilisateur peut donc omettre l'ordre END si l'exécution se termine à la fin du fichier. De plus, l'interpréteur permet de placer plusieurs ordres END dans un programme BASIC. (3, 4)
- Les fonctions définies par l'utilisateur ne sont pas restreintes en position aux lignes de numéros inférieurs à celles où elles sont référencées. (157, 159, 162)
- Le HP-71 accepte des éléments nuls dans les instructions DATA. (105)
- Les ordres READ et INPUT reconnaissent les expressions et non seulement les constantes. (112)
- Le système ne donne pas une erreur si l'exécution se termine avant un ordre NEXT correspondant à un ordre FOR exécuté (50)

Déviations du HP-71 par rapport au BASIC minimal

Le HP-71 n'est pas conforme au BASIC minimal dans les cas suivants:

- Le HP-71 affecte les variables FOR de gauche à droite lors de l'entrée dans une boucle.

Exemple: La boucle suivante est exécutée une fois par le HP-71 (ANSI demande six exécutions):

```

:
150 J= -2
160 FOR J=9 TO J STEP J
170 NEXT J
:

```

ANSI demande l'évaluation de la limite et de l'incrément une fois lors de l'entrée dans la boucle. Le HP-71 s'y conforme *après* avoir défini la valeur initiale. (48)

- Les réponses du HP-71 aux erreurs d'entrée et les normes ANSI sont listées ci-dessous (pour les réponses DEFAULT EXTEND, consultez le *Manuel de référence*).

Réponses aux erreurs d'entrée: HP-71 — ANSI

Entrée	HP-71		ANSI
	DEFAULT ON	DEFAULT OFF	
Dépassement inférieur numérique.	Avertissement: le système donne zéro.	Erreur: l'utilisateur doit entrer une nouvelle valeur.	Le système donne zéro.
Dépassement supérieur numérique.	Avertissement: le système donne sa plus grande valeur avec signe.	Erreur: l'utilisateur doit entrer une nouvelle valeur.	L'utilisateur doit entrer une nouvelle valeur.
Dépassement supérieur alphanumérique.	Pistable; sinon fatale.		Permet une nouvelle entrée.
Affectation de variable.	Après la vérification de chaque élément. tous les éléments.		Après la vérification de
Erreur d'exécution.	Pistable; sinon fatale. ré-entrer la ligne.		L'utilisateur doit

- Si un programme BASIC minimal contient l'instruction suivante:

```
INPUT I, A(I), X
```

et si la réponse de l'utilisateur est:

```
1,2,"abc"
```

alors "abc" est une entrée non-valide et ANSI n'affecte pas non plus I et A(I). Le HP-71 demande à l'utilisateur de ré-entrer les valeurs mais I et A(I) ont les valeurs courantes (1 et 2 respectivement). Si l'utilisateur entre maintenant 2, 3, 4, A(1) a la valeur 2 et A(2) a la valeur 3. Avec ANSI, A(1) n'a pas de valeur et A(2)=3. (108.3)
- Dans le BASIC minimal, \emptyset/\emptyset donne un avertissement et une valeur en dépassement. Le HP-71 donne une erreur pour \emptyset/\emptyset . (28)
- Le BASIC minimal suppose OPTION BASE \emptyset actif sauf si vous exécutez OPTION BASE 1 avant toute déclaration de tableau. Le HP-71 utilise la limite active en mémoire permanente lorsque le programme commence. Pour éviter des limites incertaines, tout programme du HP-71 utilisant des tableaux doit contenir un ordre OPTION BASE (page 68). (56)

- Le BASIC minimal suppose le radian comme unité d'angle sauf si vous exécutez DEGREES dans le programme. Un programme du HP-71 utilise l'unité d'angle active en mémoire permanente lorsque le programme commence. Pour éviter des unités d'angle incertaines, nous vous recommandons de placer un ordre RADIANS ou DEGREES dans les programmes utilisant des fonctions d'angles (page 50). (120)
- Un programme BASIC minimal utilise la même séquence de nombres aléatoires lors de chaque exécution, sauf si le programme exécute un ordre RANDOMIZE. Un programme du HP-71 utilise une séquence de nombres aléatoires différente lors de chaque exécution. Un programme du HP-71 peut néanmoins utiliser la même séquence s'il contient un ordre RANDOMIZE *expression numérique* (page 52). Le résultat de l'expression numérique détermine la séquence utilisée. (130)
- Un programme BASIC minimal déclare les variables comme spécifiées par les ordres DIM, que l'exécution passe ou non sur ces ordres DIM. Le HP-71 demande que l'exécution passe sur ces ordres DIM, sinon les déclarations n'ont pas lieu. (62)

Alimentation

Consommation

Le HP-71 consomme un minimum de courant lorsque l'affichage est éteint (après OFF, BYE,  OFF ou 10 minutes d'inactivité). Il consomme plus de courant lorsque l'ordinateur est «allumé» et plus encore pendant l'exécution d'un programme ou l'utilisation de l'indicateur sonore.

Lorsque l'ordinateur est éteint, il consomme environ 0,03 mA. Lorsqu'il est allumé, il consomme 0,75 mA. Pendant l'exécution d'un programme ou l'utilisation de l'indicateur sonore, il consomme 10 mA. Une batterie de piles neuves permet d'utiliser un HP-71 équipé de quatre modules mémoire pendant au moins 60 heures de fonctionnement continu (exécution de programme) à une température d'environ 25°C avant que le témoin **BAT** s'allume. L'installation d'un lecteur de carte ou d'une interface HP-IL réduit la durée de vie des piles selon l'utilisation.

Eteignez toujours le HP-71 avant de le connecter au secteur. Ceci évite les pics de tension pouvant affecter le contenu de la mémoire. Lorsque le HP-71 est connecté au secteur, il n'utilise les piles qu'en cas de panne du secteur. Vous n'endommagerez pas le HP-71 en l'utilisant sans piles mais vous pourrez perdre le contenu de la mémoire s'il y a une interruption, même courte, de l'alimentation secteur.

Indicateur de baisse de charge

Le HP-71 possède des protections pour la mémoire en cas de baisse de charge des piles. Après la première indication de baisse de charge, remplacez ou rechargez les piles dès que possible.

- Lorsque la tension devient inférieure à un seuil prédéfini, le HP-71 allume le témoin **BAT** pour vous indiquer qu'il vous reste entre 5 minutes et 2 heures d'utilisation continue (selon l'état des piles).

- Vous pouvez continuer à utiliser l'ordinateur après l'apparition du témoin **BAT**. Ceci peut néanmoins provoquer une ré-initialisation de la mémoire si la charge devient trop faible.
- Le lecteur de carte peut ne pas fonctionner correctement lorsque la charge est faible. Le HP-71 ne refuse pas toujours les opérations de lecteur de carte mais affiche le message:

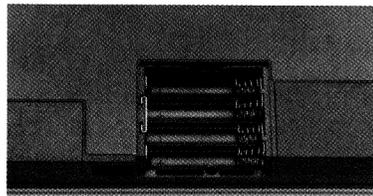
WRN: Low Battery

Remplacement de la batterie de piles

Le HP-71 utilise quatre piles alcaline de type AAA. Lorsque vous retirez les piles, vous disposez de 30 secondes pour en placer de nouvelles avant de perdre le contenu de la mémoire. Si vous disposez d'un adaptateur secteur, connectez-le de façon à ne pas risquer de perdre le contenu de la mémoire. Si vous ne disposez pas d'adaptateur secteur, nous vous recommandons de copier vos fichiers sur cartes magnétiques ou sur cassette au cas où la mémoire serait ré-initialisée.

Pour installer les piles dans le HP-71:

1. Eteignez le HP-71. Appuyez sur **f** **OFF** ou frappez **OFF** **END LINE**.
2. Retournez l'ordinateur et placez-le sur une surface plane et molle.
3. Appuyez sur la porte du compartiment des piles avec votre pouce (porte centrale) et faites-la glisser vers l'arrière.
4. Retirez les quatre piles et placez les nouvelles en veillant à les positionner conformément aux indications données dans le compartiment.
5. Placez la porte en position et faites-la glisser vers l'avant jusqu'au déclic.



Informations d'entretien

Vous pouvez nettoyer le HP-71 avec un chiffon humide (un détergent doux est acceptable). N'utilisez pas un chiffon trop humide ou un détergent abrasif, particulièrement sur la fenêtre d'affichage.

Modules enfichables

Le HP-71 dispose de quatre logements externes pour des modules de mémoire vive (RAM) ou de mémoire morte (ROM), d'un logement pour le lecteur de carte et d'un logement pour l'interface HP-IL. Chaque logement est protégé par un capuchon lorsque vous recevez le HP-71. Laissez ces capuchons en place si vous n'utilisez pas les logements.

Les instructions d'installation et d'utilisation des modules mémoires, des modules préprogrammés et du module HP-IL se trouvent dans la documentation accompagnant chacun d'entre-eux. Les instructions pour le lecteur de cartes sont données dans l'annexe C de ce manuel, page 284.

Vérification du fonctionnement

Si vous suspectez un mauvais fonctionnement de votre ordinateur, suivez la procédure ci-dessous:

1. Eteignez le HP-71 (appuyez sur ou exécutez OFF).
2. Connectez l'adaptateur au secteur puis à l'ordinateur.
3. Allumez l'ordinateur (appuyez sur).
4. Exécutez la fonction PI. Frappez PI . L'ordinateur doit afficher 3.14159265359, ce qui indique qu'environ 60% des circuits de l'ordinateur fonctionnent correctement.
5. Si l'ordinateur refuse plusieurs fois une certaine opération, telle que la copie d'un fichier sur carte, ou affiche répétitivement le même message d'erreur, tel que `Excess Chars`, relisez attentivement les instructions du manuel concernant cette opération; il est possible que vous spécifiez incorrectement l'opération.
6. Si l'ordinateur ne fonctionne toujours pas correctement, appuyez simultanément sur et puis sur pour effectuer une ré-initialisation de premier niveau. L'affichage doit alors contenir le curseur de remplacement (mode BASIC) ou le curseur d'insertion (mode CALC).
7. Appuyez simultanément sur et puis sur pour exécuter une ré-initialisation de niveau 2 (INIT: 2). L'ordinateur effectue alors un auto-test de ses circuits. Le HP-71 affiche ROM TEST 1 au début du test. Une fois la première partie du test terminée, l'ordinateur affiche:

```
ROM TEST 1G 2,
```

indiquant qu'il effectue la deuxième partie du test. Lorsque le test complet est terminé, le HP-71 affiche:

```
ROM TEST 1G 2G 3G 4G
```

si le test n'a pas décelé de circuits défectueux. Si le test décèle des circuits défectueux, un des nombres sera suivi d'un B au lieu du G. Par exemple:

```
ROM TEST 1G 2B 3G 4G.
```

indique que le test a trouvé une ROM défectueuse. L'ordinateur doit alors être réparé.

8. Si l'affichage reste vierge lorsque vous appuyez sur ou si les caractères sont «gelés» à l'affichage, ré-initialisez la mémoire:
 - a. Déconnectez l'adaptateur.
 - b. Déconnectez tous les modules.
 - c. Retirez les piles.
 - d. Appuyez sur pendant environ 30 secondes pour décharger les circuits.
 - e. Remettez les piles ou connectez l'adaptateur puis appuyez sur . L'ordinateur doit alors afficher `Memory Lost` et la pression d'un touche quelconque doit afficher le symbole BASIC et le curseur de remplacement.

Si vous ne pouvez pas déterminer la cause du problème avec cette procédure, contactez nos bureaux à l'une des adresses listées page 276.

Garantie

Le HP-71 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 modifications 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

L'ordinateur vous est livré selon les spécifications en vigueur au moment de la fabrication. Hewlett-Packard n'est pas tenu de modifier des ordinateurs déjà vendus.

* Pour la France seulement:

Lorsque l'acheteur est non-professionnel ou consommateur au sens de la loi 78-23 du 10 janvier 1978, les obligations de Hewlett-Packard définies ci-dessus ne sont pas exclusives de la garantie légale en matière de vices cachés (Articles 1641 et suivants du Code Civil).

Informations

Pour toute question concernant les termes de cette garantie, veuillez contacter:

- En Europe

Hewlett-Packard S.A.
150, route du Nant-d'Avril
P.O. Box CH-1217 Meyrin 2
Geneva
Switzerland
Téléphone: (022) 83 81 11

- Aux Etats-Unis:

Hewlett-Packard Company
Portable Computer Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330
Téléphone: (503) 758-1010
Toll-Free Number: (800) 547-3400 (sauf Orégon, Hawaii et Alaska)

- Dans les autres pays:

Hewlett-Packard Intercontinental
3495 Deer Creek Rd.
Palo Alto, California 94304
U.S.A.
Téléphone: (415) 857-1501

Nota: N'envoyez pas votre ordinateur à ces adresses. Consultez la liste des centres de maintenances en page suivante.

Maintenance

Les ordinateurs 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 ordinateur 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:

Allemagne

HEWLETT-PACKARD GmbH
Kleinrechner-Service
Vertriebszentrale
Bernner Strasse 117
Postfach 560 140
D-6000 Frankfurt 56
Téléphone: (611) 50041

Autriche et pays de l'Est

HEWLETT-PACKARD Ges.m.b.H.
Kleinrechner-Service
Wagramerstrasse-Lieblgasse 1
A-1220 Wien (Vienna)
Téléphone: (0222) 23 65 11

Belgique

HEWLETT-PACKARD BELGIUM SA/NV
Woluwedal 100
B-1200 Brussels
Téléphone: (02) 762 32 00

Danemark

HEWLETT-PACKARD A/S
Datavej 52
DK-3460 Birkerød (Copenhagen)
Téléphone: (02) 81 66 40

Espagne

HEWLETT-PACKARD ESPANOLA S.A.
Calle Jerez 3
E-Madrid 16
Téléphone: (1) 458 2600

Finlande

HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 Espoo 10 (Helsinki)
Téléphone: (90) 455 02 11

France

HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de Poche
F-91947 Les Ulis Cedex
Téléphone: (6) 907 78 25

Hollande

HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
NL-1181 KK Amstelveen (Amsterdam)
P.O. Box 667
Téléphone: (020) 472021

Italie

HEWLETT-PACKARD ITALIANA S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 Cernusco Sul Naviglio (Milan)
Téléphone: (2) 90 36 91

Norvège

HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 Oesteraas (Oslo)
Téléphone: (2) 17 11 80

Royaume-Uni

HEWLETT-PACKARD Ltd
King Street Lane
GB-Winnersh, Wokingham
Berkshire RG11 5AR
Téléphone: (0734) 784 774

Suède

HEWLETT-PACKARD SVERIGE AB
Skalholtsgatan 9, Kista
Box 19
S-163 93 Spanga (Stockholm)
Téléphone: (08) 750 2000

Suisse

HEWLETT-PACKARD (SCHWEIZ) AG
Kleinrechner-Service
Allmend 2
CH-8967 Widen
Téléphone: (057) 31 21 11

Dans les autres pays

Tous les centres de maintenance Hewlett-Packard ne sont pas équipés pour assurer la maintenance des ordinateurs. Cependant, si vous avez acheté votre ordinateur 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'ordinateur ne peut pas y être réparé, veuillez l'envoyer au centre de maintenance le plus proche.

Les frais d'expédition et 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 ordinateurs 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 ordinateur réparé par Hewlett-Packard est garanti, pièces et main-d'œuvre, pendant 90 jours à compter de la date de réparation.

Instructions d'expédition

N'envoyez pas les piles dans ou avec l'ordinateur.

Si vous devez nous renvoyer votre ordinateur pour réparation, conformez-vous aux indications suivantes:

- Joignez à l'ordinateur la carte de maintenance portant la description de la panne.
- Si l'ordinateur est sous garantie, joignez une copie de la facture ou une preuve de la date d'achat.

L'ordinateur, la carte de maintenance, une courte description du problème et, si nécessaire, une preuve de la date d'achat doivent être envoyés dans un emballage protecteur. Les détériorations dues au transport ne sont pas couvertes par la garantie. Nous vous recommandons d'assurer votre envoi. Envoyez votre ordinateur au centre de maintenance le plus proche (si vous n'êtes pas dans le pays où vous avez acheté l'ordinateur, consultez la section sur les réparations dans les autres pays).

Que l'ordinateur soit sous garantie ou non, les frais d'expédition à Hewlett-Packard sont à votre charge. Les ordinateurs sont renvoyés port payé.

Détérioration de la batterie de piles

Ne renvoyez pas les piles dans ou avec l'ordinateur. La garantie ne couvre pas les piles et les détériorations qu'elles peuvent provoquer.

Si votre HP-71 est endommagé par la coulure d'une pile, contactez d'abord le fabricant de la pile. Certains fabricants de piles acceptent de réparer l'ordinateur s'il a été endommagé par une coulure de pile. Si le fabricant des piles que vous utilisez ne garantit pas ces dommages, envoyez-nous l'ordinateur pour réparation. Que l'ordinateur soit sous garantie ou non, la réparation sera à vos frais.

Assistance technique

Les informations et programmes de ce manuel supposent que le lecteur possède une connaissance de base de la terminologie utilisée. Le support technique offert par Hewlett-Packard se limite aux explications des procédures décrites dans le manuel et aux vérifications des résultats obtenus dans les exemples. Pour tout problème technique concernant ce manuel, consultez d'abord le *Manuel de référence* du HP-71. Pour plus d'informations consultez le bureau commercial HP ou écrivez à:

Hewlett-Packard Company
Portable Computer Division
Customer Support
1000 N.E. Circle Blvd.
Corvallis, OR 97330

Accessoires

Votre HP-71 est accompagné des accessoires suivants:

- Le *Manuel d'utilisation*.
- Le *Manuel de référence*.
- L'*Aide mémoire*.
- Un étui pour l'ordinateur.
- Un grille de clavier pour personnalisation.
- Quatre piles AAA.
- Une brochure d'accessoires optionnels.
- Une carte de maintenance.

La brochure d'accessoires décrit les accessoires optionnels disponibles pour votre HP-71. Pour plus d'informations consultez un revendeur agréé ou le bureau commercial le plus proche.

La disponibilité des accessoires, standard ou optionnels, est sujette à modification.

Utilisation du lecteur de cartes magnétiques HP 82400A

Présentation	284
Installation du lecteur	285
Retrait du lecteur	285
Entretien du lecteur et des cartes	286
Entretien des cartes	286
Nettoyage de la tête de lecture	286
Marquage des cartes magnétiques	286
Les cartes magnétiques	287
Utilisation du lecteur	287
Opérations du lecteur	289
Copie de carte en mémoire	291
Copie de mémoire sur carte	291
Protection des fichiers sur carte (PROTECT, UNPROTECT)	292
Utilisation de carte de fichier privé (:PCRD)	292
Etiquette d'un fichier sur carte (CAT CARD)	293

Présentation

Cette annexe s'adresse uniquement à ceux qui disposent du lecteur de cartes magnétiques HP 82400A et décrit comment installer et utiliser ce lecteur.

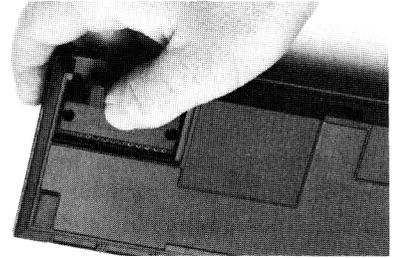
Nous vous recommandons de vous familiariser avec les opérations de fichiers décrites au chapitre 6. Les informations de ce chapitre vous aideront à comprendre les procédures de copie de fichiers.

Cette annexe couvre les sujets suivants:

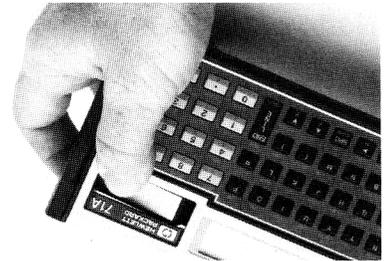
- Installation et test du lecteur.
- Entretien du lecteur et des cartes.
- Copie de fichiers de et sur des cartes magnétiques.
- Obtention d'informations du lecteur.

Installation du lecteur

1. Eteignez le HP-71 (appuyez sur **f** **OFF**).
2. Le lecteur se place dans le logement situé à droite de l'affichage, au dessus des touches numériques. Retournez l'ordinateur et retirez la porte du logement en appuyant dessus et en la faisant glisser vers l'arrière.
3. Remettez l'ordinateur à l'endroit et appuyez sur le couvercle de plastique situé dans le logement jusqu'à ce qu'il tombe. Nous vous recommandons de conserver ce couvercle au cas où vous vouliez retirer le lecteur.



4. Retournez l'ordinateur et orientez le lecteur de telle façon que le label soit vers le logement et le connecteur en face des broches de l'ordinateur. Appuyez sur le lecteur près du connecteur jusqu'au déclic.



Nota: Le connecteur peut être ajusté serré sur les broches et il sera peut être nécessaire d'appuyer fortement. Veillez cependant à ne pas le forcer s'il n'est pas correctement aligné sur les broches.

5. Remettez la porte sur le logement.

Retrait du lecteur

Pour retirer le lecteur:

1. Eteignez le HP-71 (appuyez sur **f** **OFF**).
2. Retirez la porte du logement.
3. Appuyez avec votre pouce sur le lecteur par le dessus de l'ordinateur jusqu'à ce qu'il se détache.
4. Remettez le couvercle de plastic.
5. Remettez la porte du logement.

Entretien du lecteur et des cartes

Entretien des cartes

La propreté des cartes magnétiques est un élément essentiel au bon fonctionnement du lecteur. Une surface de carte poussiéreuse ou grasse affectera la qualité des transferts d'informations. Manipulez donc les cartes par la tranche. Vous pouvez nettoyer les cartes avec un chiffon doux imbibé d'alcool isopropyle.

Placez la carte sur une surface plane et propre et frottez le côté ne portant pas d'inscription avec votre chiffon.

Une pliure ou une rainure peut endommager une carte définitivement. Nous vous conseillons de garder des doubles de vos fichiers importants et de conserver ces cartes dans un étui.



Nettoyage de la tête de lecture

La tête de lecture est similaire à celle d'un magnétophone de haute fidélité; toute accumulation de poussière affecte la qualité du contact avec les cartes et donc celle des transferts. Vous pouvez nettoyer la tête de lecture en faisant passer dans le lecteur la carte spéciale fournie à cet effet. Il n'est pas nécessaire d'exécuter d'instruction avant de faire passer la carte dans le lecteur.

ATTENTION

L'utilisation de cette carte ne devrait être nécessaire que très rarement. L'utilisation fréquente de cette carte peut user prématurément la tête de lecture.

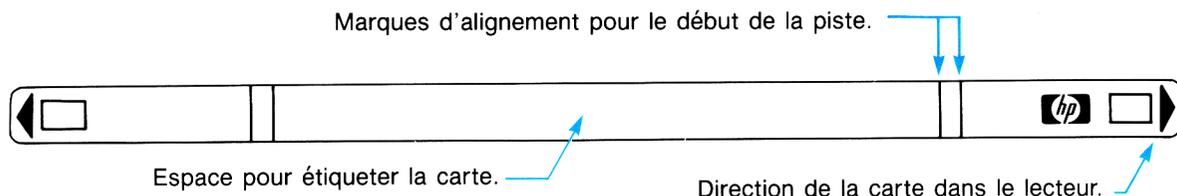
Marquage des cartes magnétiques

Vous pouvez écrire sur le côté blanc des cartes avec tout stylo n'ayant pas une pointe dure. De nombreux feutres non effaçables sont disponibles. Le crayon peut être effacé mais risque de «baver».

Les cartes magnétiques

Chaque carte comporte deux *pistes* qui contiennent toutes deux:

- L'étiquette du fichier enregistré sur cette carte.
- Le nombre total de pistes du fichier.
- Le numéro d'identification de *cette* piste, nombre entre 1 et le nombre total de pistes.
- L'état de protection contre l'écriture.
- 650 octets d'information du fichier. Une piste ne peut contenir des informations que pour un seul fichier. Une carte peut donc contenir au plus deux fichiers.



Lorsque vous passez des cartes dans le lecteur, veillez à ce que la face imprimée soit vers le haut. L'ordre des pistes est sans importance.

Nota: Veillez à garder les cartes magnétiques propres et ne les manipulez que par la tranche. Des poussières ou même des empreintes altèrent les performances du lecteur, provoquent l'affichage de messages d'erreur et réduisent la durée de vie des cartes. Vous pouvez nettoyer les cartes avec un chiffon doux imbibé d'alcool isopropyle. Conservez vos cartes à l'abri de forts champs magnétiques, tels que les aimants, les câbles conduisant de forts courants, les transformateurs etc. Un fort champ magnétique peut endommager les cartes de façon permanente.

Utilisation du lecteur

Le HP-71 affiche de nombreux messages pendant les opérations de lecteur. Toutes les opérations du lecteur passent par les étapes suivantes:

1. Frappez un ordre de lecture en mode BASIC (tel que COPY TO CARD).
2. Appuyez sur `ENDLINE` pour lancer l'opération. Le HP-71 répond par le message approprié et attend:

```
Wrt: Align then ENDLN
```

Le HP-71 attend votre réponse. Vous pouvez appuyer sur `ATTN` pour annuler l'opération.

3. Insérez la piste voulue dans le lecteur de telle façon que seul un des deux traits d'alignement disparaissent dans la fente; la carte doit dépasser de l'autre côté du lecteur. Appuyez une deuxième fois sur `[END LINE]`. Le HP-71 affiche:

```
Pull 1 of n
```

4. Tirez la carte à travers le lecteur. Le HP-71 vous donne environ cinq secondes pour commencer à tirer sur la carte. Si vous ne touchez pas à la carte, le HP-71 émet un bip, affiche l'avertissement `R/W Error` et vous demande de recommencer. Si vous ne voulez *pas* tirer sur la carte, attendez que le HP-71 affiche à nouveau `Align then ENDLN` (environ 7 secondes) puis appuyez sur `[ATTN]`.
5. Une fois la carte passée dans le lecteur, le HP-71 peut répondre de plusieurs façons:
 - Si vous copiez sur carte un fichier de mémoire, le HP-71 vous demande de passer une deuxième fois la même piste:

```
Vfy: Align then ENDLN
```

Cette fois-ci, le HP-71 vérifie que les informations ont été correctement copiées sur la carte. Si la vérification est négative, le HP-71 affiche l'avertissement `Verify Fail` et vous demande de recommencer l'opération: un passage de la carte pour copier et un second pour vérification (si la copie est à nouveau défectueuse, nettoyez la carte ou utilisez-en une autre).

- Le HP-71 vous indique lorsque vous avez terminé la lecture d'une piste.

```
Trk n Done
```

- Si le fichier utilise plus d'un piste, le HP-71 vous demande la piste suivante:

```
Wrt: Align then ENDLN
```

```
Pull 2 of n
```

L'ordinateur est prêt pour la piste suivante de la même carte ou d'une autre carte.

- Si vous tirez trop vite ou trop lentement, le HP-71 affiche l'avertissement `Too Fast` ou `Too Slow` et vous demande de recommencer. Si vous tirez *très* lentement sur une carte, le HP-71 peut répondre comme si vous n'aviez pas passé de carte.
- Le HP-71 vous demande autant de passage que nécessaire. Lorsque l'opération est finie, il réaffiche le symbole de mode BASIC et le curseur.

Opérations du lecteur

Le HP-71 dispose des ordres suivants pour contrôler le lecteur:

- | | |
|-------------------------------------|--|
| • <code>CAT CARD</code> | Affiche l'étiquette d'un fichier sur carte. |
| • <code>COPY fichier TO CARD</code> | Copie sur carte un fichier de mémoire. Le nom de <i>fichier</i> peut contenir un nom d'unité. |
| • <code>COPY CARD TO fichier</code> | Copie en mémoire principale un fichier sur carte. Vous ne pouvez pas copier en mémoire vive indépendante un fichier sur carte. |
| • <code>PROTECT</code> | Protège une piste contre l'écriture. |
| • <code>UNPROTECT</code> | Enlève la protection contre l'écriture d'une piste. |
| • <code>RUN :CARD</code> | Charge en mémoire un programme sur carte, le déclare comme fichier courant et l'exécute. |
| • <code>CHAIN :CARD</code> | Supprime le fichier courant, charge un programme sur carte, le déclare comme fichier courant et l'exécute. |

Tous les ordres de lecture sont programmables. Seul `RUN :CARD`, `RUN :PCRD`, `CHAIN :CARD` et `CHAIN :PCRD` changent la déclaration de fichier courant.

Vous pouvez spécifier une carte magnétique dans une opération en utilisant les noms d'unité `:CARD` et `:PCRD` (`:PCRD`, qui spécifie un fichier privé sur carte, est décrit ci-dessous. `CARD` est aussi un mot réservé qui ne peut pas être utilisé comme nom de fichier). Les deux mots spécifient une unité de la même façon que `PORT` et `MAIN` lorsque vous les utilisez avec l'ordre `COPY`. La copie d'informations de et vers des cartes est similaire à la copie de et vers les mémoires vives indépendantes.

Pour spécifier un fichier sur carte, vous pouvez utiliser l'une des formes suivantes:

- `CARD`
- `:CARD`
- `:PCRD`
- *nom de fichier* `:CARD`
- *nom de fichier* `:PCRD`

L'exemple suivant illustre une opération typique utilisant le lecteur de carte. Cet exemple suppose que le fichier réside sur trois pistes.

COPY CARD TO ACRES ENDLINE

Copie en mémoire sous le nom ACRES un fichier sur carte.

Read: Align then ENDLN

Le HP-71 vous demande d'aligner la piste.

(alignez la piste)

ENDLINE

Indique que vous êtes prêt à passer la carte.

Pull Card

Le HP-71 vous demande de tirer sur la carte.

(tirez sur la carte)

Trk 1 done

Indique que le HP-71 a bien lu la piste.

Read: Align then ENDLN

Le HP-71 vous demande d'aligner une autre piste.

(alignez la piste)

ENDLINE

Indique que vous êtes prêt à tirer sur la carte.

Pull 2 of 3

Le HP-71 vous demande de tirer sur la carte.

(tirez sur la carte)

Trk 2 Done

Indique que le HP-71 a bien lu la deuxième piste.

Read: Align then ENDLN

Le HP-71 vous demande d'aligner la piste suivante.

(alignez la piste)

ENDLINE

Indique que vous êtes prêt à tirer sur la carte.

Pull 3 of 3

Le HP-71 vous demande de tirer sur la carte.

(tirez sur la carte)

Après la lecture de la dernière piste, le fichier est stocké et le symbole BASIC apparaît.

Après la lecture de la dernière piste, le HP-71 ne demande plus de piste.

L'exemple ci-dessus illustre un opération typique du lecteur. Le HP-71 affiche d'abord un message tel que `Read: Align then ENDLN` puis vous demande les pistes suivantes si nécessaire.

Copie de carte en mémoire

Lorsque vous voulez copier un fichier sur carte, vous devez spécifier le lecteur comme unité du fichier source dans l'ordre `COPY`. Si vous exécutez l'ordre `COPY nom de fichier` sans spécifier le lecteur comme unité source, le HP-71 cherche le fichier en mémoire RAM et ROM. Les fichiers sur carte ne peuvent être copiés directement que dans la mémoire vive principale. Une fois en mémoire principale, vous pouvez bien entendu les copier dans une autre unité (vous ne pouvez pas copier un fichier directement de carte sur carte).

Exemples:

```
COPY CARD TO VOILIER
```

Copie un fichier sur carte dans un fichier nommé VOILIER.

```
COPY :CARD TO TRIANGLE
```

Vous pouvez utiliser `CARD` ou `:CARD`

```
COPY CARD
```

Copie un fichier sur carte en mémoire principale en utilisant le même nom.

```
COPY LATERAL: CARD
```

Copie le fichier sur carte LATERAL en mémoire principale.

```
COPY GRAPHICS: CARD TO IMAGES
```

Copie le fichier sur carte GRAPHICS en mémoire principale sous le nom IMAGES.

Copie de mémoire sur carte

Vous pouvez copier tout fichier en RAM ou ROM sur des cartes magnétiques.

Exemples:

```
COPY VOILIER:PORT(0) TO CARD
```

Copie VOILIER du logement 0 sur carte.

```
COPY TRIANGLE:MAIN TO :CARD
```

Copie TRIANGLE de la mémoire principale sur carte.

```
COPY TO CARD
```

Copie le fichier courant sur carte.

```
COPY IMAGES:PORT(0)
  TO GRAPHICS: CARD
```

Copie IMAGES du logement 0 sur carte en utilisant le nom GRAPHICS.

Protection de fichiers sur carte (PROTECT, UNPROTECT)

L'ordre PROTECT vous permet de protéger le contenu de cartes déjà enregistrées contre un effacement accidentel.

```
PROTECT
```

Lorsque vous exécutez cet ordre, le HP-71 vous demande de passer une carte dans le lecteur. Lorsque vous tirez la carte à travers le lecteur, le HP-71 enregistre une marque de protection sur la carte. Si vous voulez protéger l'autre piste, vous devez exécuter PROTECT à nouveau et passer l'autre piste dans le lecteur.

Vous pouvez retirer une protection en utilisant l'ordre UNPROTECT.

```
UNPROTECT
```

Lorsque vous exécutez UNPROTECT, le HP-71 vous demande de passer une carte dans le lecteur. Le HP-71 efface alors la marque de protection de la piste passée.

Utilisation de carte de fichier privé (:PCRD)

Vous pouvez privatiser le contenu d'une carte en utilisant :PCRD au lieu de :CARD comme nom d'unité pour le nouveau fichier. Un programme sur carte privé ne peut être que copié et exécuté; il ne peut être ni lu ni modifié. De même que pour les fichiers en mémoire vive, les fichiers privés sur carte ne peuvent pas être «déprotégés» (les fichiers privés en mémoire vive sont décrits au chapitre 6 sous le titre «Contrôle de l'accès aux fichiers», page 116).

Copie d'un fichier sur une carte privée. Vous ne pouvez rendre privé un fichier sur carte qu'au moment de la copie. Pour cela, utilisez la syntaxe suivante de l'ordre COPY:

— syntaxe simplifiée —

```
COPY [nom de fichier] [:unité] TO [nom de fichier] :PCRD
```

Exemples:

```
COPY TYPESET:MAIN TO :PCRD
```

Copie le fichier TYPESET de mémoire vive sur une carte privée.

```
COPY TO ELECTRON:PCRD
```

Copie le fichier courant courant sur une carte privé et le nomme ELECTRON.

```
COPY TO :PCRD
```

Copie le fichier courant sur une carte privée.

Copie d'une carte privée dans un fichier. Vous pouvez copier une carte privée dans un fichier en spécifiant `CARD` ou `:PCRD` comme unité source dans l'ordre `COPY`. Il n'est pas nécessaire de spécifier une *carte privée* comme unité source lors de la copie en fichier d'une carte privée. Lorsque le HP-71 copie en mémoire une carte privée, le fichier résultant en mémoire est privé.

Exemples:

```
COPY :PCRD TO TYPESET
```

Copie la carte privée dans le fichier `TYPESET`.

```
COPY ELECTRON:PCRD
```

Copie `ELECTRON` d'une carte en mémoire principale.

```
COPY CARD
```

Copie un fichier sur carte en mémoire principale.

Etiquette d'un fichier sur carte (`CAT CARD`)

L'ordre `CAT CARD` vous permet d'afficher l'étiquette d'un fichier sur carte.

```
CAT CARD
```

Exemple: Affiche l'étiquette pour un fichier sur carte nommé `PROG1`.

```
CAT CARD ENDLINE
```

```
Cat: Align then ENDLN
```

Vous demandez d'aligner une piste et d'appuyer sur **ENDLINE**.

(Alignez une piste)

```
ENDLINE
```

Vous disposez d'environ 7 secondes pour passer la carte.

(Tirez sur la carte)

```
PROG1 (trk nnn of nnn)
```

Affiche le numéro de la piste lue.

```
PROG1 P BASIC 37 01  
/11/83 22:36
```

Affiche l'étiquette.

Lorsque vous essayez d'obtenir l'étiquette d'une carte écrite par un autre ordinateur ou d'une carte vierge, le HP-71 donne zéro comme type de fichier.

Index des sujets

Les numéros de page en caractères gras donnent les références de pages principales et les numéros en caractères réguliers indiquent les références secondaires. En plus de cet index des sujets, vous trouverez un index des mots-clés, groupés par catégories, à l'intérieur du dos de couverture.

A

- ABS, fonction, **48**
- AC, témoin d'état, **30**
- Accès aux fichiers de données, **249-263**. Voir aussi *rappel de données de programme*
- Accessoires, **282**
- ACOS, fonction, **52**
- ACS, fonction, **52**
- ADD, ordre, **80-82**, **87**
- Addition (+), opérateur, **47**
- ADJABS, ordre, **92-93**
- ADJUST, ordre, **94-95**
- AF, ordre, **94-95**
- Affectation (LET), **67**, **269**
 - En mode CALCUL, **38-39**
 - Multiple, **67**, **146**
 - Sous-chaînes, **74**
- Affichage
 - Catalogue de carte, **289**, **293**
 - Catalogue de fichiers, **102**, **117-118**
 - Champs protégés (WINDOW), **135-136**
 - Champs protégés par des séquences d'échappement, **237-239**
 - Contrôle, **26**, **29-30**, **135-140**, **232-239**
 - Définitions de touches (FETCH KEY, KEYDEF#, **VIEW**), **125-126**
 - Entrée de ligne (LINPUT), **244-245**
 - Étiquette de fichier. Voir *Manuel de référence* du HP-71
 - Fenêtre, **14-16**
 - Graphiques (GDISP), **137-139**
 - Indicateurs de format (-13, -14), **198-199**
 - Informations, **24**, **67**, **224-239**
 - Instructions de format, **55-56**
 - Lecture de caractères (DISP#), **136-139**
 - Lignes de programme (FETCH, LIST, GOTO, **▲**, **▼**), **21**, **156-158**
 - Liste, **230**
 - Vitesse (DELAY), **26**
 - Zone, **227**
- Ajout de lignes à un programme, **158**
- Ajout de points à un tableau statistique (ADD), **80-82**, **87**
- Ajustement de courbe, **86-89**
- Ajustement de la vitesse de l'horloge (SETTIME, ADJUST, AF, EXACT, RESET CLOCK), **94-96**
- Aléatoires, nombres (RND), **52-53**
- Alimentation, **271-273**
- Ampersand (opérateur &), **73**
- AND, opérateur logique, **62-63**
- ANGLE, fonction, **52**
- ANSI, conformité de l'interpréteur, **268-271**
- Annulations
 - Définitions de touches (DEF KEY), **128**
 - Pistage (TRACE OFF), **168**
 - Protection de carte (UNPROTECT), **289**, **292**
- Apostrophes, **67**, **72**, **269**
- Arc cosinus (ACOS ou ACS), **52**
- Arc sinus (ASIN ou ASN), **52**
- Arc tangente (ATAN ou ATN), **52**
- Arc tangente dans le quadrant correct (ANGLE), **52**
- Arithmétique
 - Hiérarchie, **64**
 - Opérateurs, **47**, **64**
- Arrêt de l'exécution, **153**
- Arrondi des nombres affichés, **56**
- ASIN, fonction, **52**
- ASN, fonction, **52**
- ASSIGN #, ordre, **248-249**
- Assistance technique, **280**
- ATAN, fonction, **52**
- ATN, fonction, **52**
- ATTN**, touche, **16**, **31**, **153**
- AUTO, ordre, **149**

Automatique

- Correspondance de parenthèses en mode CALCUL, **39-40**
- Exécution d'ordres (STARTUP), **139-140**
- Avertissement, messages, 13, 68, **175-176**
- En mode CALCUL, **46**
- Exceptions mathématiques, 68, **177, 269-270**
- Indicateur (- 1), **196**

B

BACK, touche, 12, **16, 21, 44**

BASIC

- Calculs, **19, 37**
- Conformité à la norme ANSI, **268-271**
- Extensions, **268-269**
- Fichiers, **99, 143-144**
- Indicateur (-26), **200**
- Mode, **13-14, 19, 37**
- Symbole (>), **13-14**
- Version minimale, **268-271**
- BAT**, témoin, **30, 271**
- Batterie, 13, 30, **271-272**
 - Consommation, **271**
 - Endomagement, **279**
 - Remplacement, **272**
 - Témoin (**BAT**), **30, 271**
- BEEP OFF, ordre, **32-33**
- BEEP ON, ordre, **32-33**
- BEEP, ordre, **32-33**
- BIN, fichiers binaires, **99, 160**
- Boucles de programme, **185-187, 269**
 - Emboîtées, **186-187**
- Branchements, **179-184**. Voir aussi *programme*
 - Conditionnel (ON...GOSUB, ON...GOTO, ON TIMER#...GOSUB, ON TIMER...GOTO), **181-184**
 - Non conditionnel (GOSUB, GOTO, RETURN, POP, CALL, CHAIN), **179-181**

C

- Calculs, 18-20, **36-64, 78-89**
- CALCUL, mode, 18-20, **37-46, 48**
 - Affectations, **38-39**
 - Caractéristiques, **38-44**
 - Clavier Utilisateur, **38**
 - Exécution à reculons, **44-45**
 - Exécution pas à pas (**f** **SST**), **41, 42**
 - Hierarchie des opérateurs, **41-42**
 - Messages d'avertissement, **46**
 - Opérations non autorisées, **46, 48**
 - Pile d'instructions, **42-44**
 - Rappel d'une expression complète, **42-44**
 - Résultats implicites, **38, 40**
 - Symbole de **END LINE** (␣), **42-44**
 - Traitement d'erreur, **42-46**
 - Virgule (,) pour les listes d'arguments, **40-41**
- CALC**, témoin, **19, 30**
- Calendrier, **17, 90-91**
- CALL, ordre
 - Appel de sous-programme dans un autre fichier, **210**
 - Exécution au clavier, **150-151**
 - Exécution dans un programme, **180, 205-210**
 - Paramètres, **206-209**
- Canal (ASSIGN #), **248-249**
- Canal (CALL, SUB), **206, 263-264**
- Caractère
 - Code, **74-75, 77, 132-135, 268**
 - Contrôle, **234-239**
 - Jeux, **132-135, 268**
 - Vitesse de déroulement, **26**
- Caractères secondaires, définition, (CHARSET, CHARSET#), **132-135**
- CARD, mot-clé, **289**
- :CARD, mot-clé, **289**
- Cartes magnétiques, **287-293**
 - Marquage, **286**
 - Organisation, **287**
 - Utilisation. Voir *lecteur de cartes*
- CAT, ordre. Voir *Manuel de référence* du HP-71
- CAT ALL, ordre, 102, **117-118**
- CAT CARD, ordre, **289, 293**
- Catalogue de fichiers (CAT ALL), 102, **117-118**
- Catalogue de fichiers sur carte (CAT CARD), **289, 293**
- Catégories de nombres (CLASS), **60**
- CEIL, fonction, **48**
- CFLAG, ordre, **192-193**
- CHAIN, ordre
 - Exécution au clavier, **151-152**
 - Exécution dans un programme, **180-181**
- CHAIN :CARD, ordre, **289**
- Chaîne alphanumérique, **71-77**
 - Code de caractère (CHR#, NUM), **74-75, 77**
 - Concaténation (&), **73**
 - Conversion en majuscules (UPRC#), **75**
 - Entre guillemets, **71-72**
 - Fonctions, **74-77**
 - Longueur (LEN), **74**
 - Nulle, **72, 268**
 - Par défaut (INPUT), **243-244**
 - Position de sous-chaîne (POS), **74-75**
 - Sous-chaînes, **73-75**
 - Valeur numérique équivalente (STR#, VAL), **74-77**

Variables, 71-73
 Chaîne de format (IMAGE, DISP USING, PRINT USING), 228-232
 Champs protégés (WINDOW), 135-136
 Champs protégés par des séquences de fin-de-ligne, 237-239
 [-CHAR], touche, 21
 Chargement et exécution de programme sur carte (RUN :CARD, CHAIN :CARD), 289, 291
 CHARSET, ordre, 132-135
 CHARSET#, fonction, 132-135
 CHR#, fonction, 75, 77
 Cinéma, programme, 138
 CLAIM PORT, ordre, 107
 CLASS, fonction, 60
 Clavier, 11-12
 Calculs, 18-20, 36-46
 Entrée d'informations (INPUT), 241-244, 269-270
 Normal, 28-29, 121
 Touches préfixes, 11
 Utilisateur (USER, USER ON, USER OFF, [USER], [1 USER]), 28-29, 122-124, 126
 Utilisation, 11-12
 CLSTAT, ordre, 78-80, 87
 [CMDS], touche, 31-32
 Code de caractère, 74-75, 77, 132-135, 268
 Conformité de l'interpréteur à la norme ANSI, 268-271
 Concaténation d'instructions (@), 22, 67, 146
 Concaténation de chaînes (*), 73
 Consommation, 271
 [CONT], touche, 153, 155
 CONT, ordre, 155
 CONTRAST, ordre, 29-30
 Contrôle
 Accès aux fichiers (SECURE, UNSECURE, PRIVATE) 116-117
 Affichage, 135-140, 232-239
 Curseur, 135-136
 Imprimante, 232-234
 Longueur de ligne (WIDTH, PWIDTH), 232-233
 Contrôle, caractères, 234-239
 Conventions de syntaxe, 14, 16, 21, 34
 Conversion décimal/hexadécimal (DTH#), 48
 Conversion degrés/radians (RAD), 52
 Conversion hexadécimal/décimal (HTD), 48
 Conversion radians/degrés (DEG), 52
 COPY, ordre, 102, 112-114, 127
 COPY CARD TO, ordre, 289, 291
 COPY TO CARD, ordre, 289, 291
 CORR, ordre, 84
 Correction d'erreurs. Voir aussi *traitement d'erreur*

Avec TRACE FLOW et TRACE VARS, 166-171
 Effacement de l'affichage, 16
 En mode CALCUL, 42-46
 Avec les touches de mise au point, 23-26
 Corrélation (CORR), 84
 COS, fonction, 52
 Cosinus (COS), 52
 Courant
 Fichier, 27-28, 100
 Ligne, 154, 158
 CREATE, ordre, 248, 257, 259
 Création de fichier de données (CREATE), 248, 257, 259
 Curseur
 Contrôle, 135-136, 234-239
 Déplacement, 15-16
 Insertion, 19-20, 23
 Remplacement, 14, 16

D

DATA, fichiers, 99, 247-264
 DATA, ordre, 245, 269
 Date
 Affichage (DATE), 90-91; (DATE#), 17, 90-91
 Mise à jour (SETDATE), 17, 90-91
 DATE, ordre, 90-91
 DATE#, fonction, 17, 90-91
 Déclaration de tableaux (DIM, REAL, SHORT, INTEGER), 57, 69-71
 Déclaration de mémoire vive indépendante (FREE PORT), 105-106
 DEF FN, ordre, 218-219
 DEF KEY, ordre, 28, 121-124
 Défaut
 Chaîne (INPUT), 243-244
 Dimensions de tableau, 70
 Fichier, 110
 Unité, 111
 DEFAULT ON/OFF/EXTEND, ordre, 58, 269-270
 Définition de touche (DEF KEY), é!, 121-124
 Annulation, 128
 Types, 124
 Modifications (FETCH KEY, KEYDEF#, [f] [VIEW]), 125-126
 DEG, fonction, 52
 DEGREES, ordre, 50, 270
 DELETE, ordre, 158-159
 Dépassement inférieur de capacité (UNF), 57-61
 Dépassement supérieur de capacité (OVF), 57-61
 Déplacement du pointeur (RESTORE #), 258
 Déroulement de l'affichage, 26

DESTROY, ordre, 67-68
 Détérioration de la batterie, 279
 Déviations par rapport au BASIC minimal, 269-271
 DIM, ordre, 69, 72-73, 271
 Dimensions de tableau, 69-73
 par défaut, 70
 DISP, ordre, 225-227
 Implicite, 24, 67, 226
 Utilisation de guillemets, 67
 DISP USING, ordre, 230-232
 DISP\$, fonction, 136
 DIV, opérateur, 47
 Division entière (DIV), 47
 Division, opérateur /, 47
 Division par zéro, 57-60, 270
 Données. Voir aussi *fichiers de données*
 Accès aux fichier de données, 249-263
 Création de fichier de données (CREATE), 248
 Enregistrements, 256-260
 Entrée (INPUT), 241-244, 269
 Entrée dans un tableau statistique (ADD), 80-82, 87
 Fermeture de fichier de données (ASSIGN #), 248-249
 Ouverture de fichier de données (ASSIGN #), 248-249
 Pointeur (RESTORE), 246-247, 252-253, 258
 Programme (DATA, READ, RESTORE), 245-247, 269
 Rappel dans un fichier de données (RESTORE #, READ #), 254-263
 Rappel dans un programme (READ), 246, 269
 Saisie (INPUT), 241-244, 269
 Stockage dans un programme (DATA), 245
 Stockage dans un fichier de données (RESTORE #, PRINT #), 250-253, 256-263
 DROP, ordre, 81-82
 DTH\$, fonction, 48
 DVZ, indicateur, 57-60, 270

E

E, symbole d'exponentiation, 54-56
 $e^x - 1$ (EXPM1), 50
 Ecart-type (SDEV), 83-84
 EDIT, ordre, 21, 22, 27, 143-144, 156
 Effacement
 Affichage ([ATTN]), 16
 Mémoire, 13-14
 Egal à, opérateur =, 62
 ELSE, mot-clé, 188-189
 Enchaînement de programmes (CHAIN), 151-152

END DEF, ordre, 219
 [ENDLINE], touche, 13, 31, 42, 158
 END, ordre, 155, 269
 END SUB, ordre, 204-205
 ENDLIN, ordre, 234
 ENG, ordre, 56
 Enregistrements de fichier, 256-260
 Environnement
 Actif, 211
 Global, 211
 Local, 211
 Entrée
 Données et expressions (INPUT), 241-244
 Données dans un tableau statistique (ADD), 80-82, 87
 Ligne d'affichage (LINPUT), 244-245
 Lignes de programme, 22-25, 145-149, 269
 Entretien, 272
 Environnement
 Actif, 211
 Fonctions définies par l'utilisateur, 220-221
 Global, 211
 Local, 211
 Physique, 267
 Principal, 211
 Programme et sous-programme, 153, 210-215
 EPS, fonction, 61
 Erreurs
 Conditions, 162-177
 Contrôle (ON ERROR, ERRN, ERRL), 171-175
 De syntaxe, 163-164
 En exécution, 163-165
 Exceptions mathématiques, 57-60, 176, 269-270
 Messages, 13, 68, 163-165
 Rappel de message ([ERRM], ERRM#), 164-165, 175
 Test, 148
 Types, 163
 Vérification, 148
 ERRL, fonction, 175-176
 [ERRM], touche, 164-165
 ERRM\$, fonction, 164-165
 ERRN, fonction, 173-174, 176
 Espacement des sorties (TAB, , ,), 226-227
 Etiquette de fichier (CAT). Voir *Manuel de référence* du HP-71
 EXACT, indicateur -46, 200
 EXACT, ordre, 96
 Exceptions mathématiques
 Avertissements, 177, 269-270
 Erreurs, 176, 269-270
 Indicateurs (IWL, DVZ, DVF, UNF, INX), 57-60, 68, 176-177, 197

- Proposition IEEE, **59-60**
 - Traitement (DEFAULT ON, DEFAULT, EXTEND), **58, 269-270**
 - Valeurs (TRAP), **59-60**
 - Exécution ([END LINE]), **13, 31, 42, 158**
 - A reculons en mode CALCUL, **44-45**
 - A un ligne spécifiée (RUN, GOSUB), **149-150**
 - Automatiques d'instructions à la mise sous tension (STARTUP), **139-140**
 - Du fichier courant (RUN, GOSUB, [RUN]), **25, 149-150**
 - Du fichier spécifié (RUN, CALL, CHAIN), **150-152**
 - Pas-à-pas ([F][SST]), **168-171**
 - Programme sur carte magnétiques (RUN :CARD, CHAIN :CARD), **289**
 - EXOR, opérateur logique, **62-63**
 - EXP, fonction, **50**
 - EXP(x) - 1 (EXPM1), fonction, **50**
 - EXPM1, fonction, **50**
 - EXPONENT, fonction, **50**
 - Exponentiation, opérateur ^, **47**
 - Exponentielle (EXP), **50**
 - Extinction automatique, **13**
- F**
-
- [f], touche, **11**
 - f, témoin, **30**
 - FACT, fonction, **49**
 - Facteur d'ajustement de l'horloge, **92-96**
 - Factorielle (FACT), **49**
 - Fenêtre d'affichage, **14-16**
 - Angle et contraste (CONTRAST), **29-30**
 - Déplacement, **15-16**
 - Effacement, **16**
 - Longueur, **14-15**
 - Fermeture d'un fichier de données (ASSIGN #), **248-249**
 - FETCH, ordre, **26, 156-157**
 - FETCH KEY, ordre, **125**
 - Fibonacci, programme, **216-218**
 - Fichiers. Voir aussi *BIN, LEX, BASIC, DATA, TEXT, KEY, SDATA, données*
 - Accès aux données, **249-263**
 - Catalogue de fichier sur carte (CAT CARD), **289, 293**
 - Catalogue des fichiers (CAT ALL), **102, 117-118**
 - Changement de nom (RENAME), **101, 115, 127**
 - Contrôle de l'accès (PRIVATE), **116-117**
 - Copie (COPY), **102, 112-114, 127**
 - Copie de fichiers sur cartes (COPY CARD TO, COPY TO CARD), **289, 291**
 - Création de fichier de données (CREATE), **248, 257, 259**
 - DATA, **99, 247-264**
 - Défaut, **110**
 - Enregistrements, **256-260**
 - Etiquette de fichier (CAT). Voir *Manuel de référence* du HP-71
 - Fermeture de fichier de données (ASSIGN #), **248-249**
 - Noms, **22, 109-110**
 - Nom en BASIC (EDIR); **21-22, 143-144**
 - Nom du fichier de travail (NAME), **27, 101, 145**
 - Opérations, **98-118**
 - Ordre de recherche, **112**
 - Ouverture de fichier de données (ASSIGN #), **248-249**
 - Par défaut, **110**
 - Pointeur de données, **252-253, 258**
 - Programme, **27, 143-144**
 - Protection (SECURE, UNSECURE, PRIVATE), **116-117, 127**
 - Rappel direct (READ #), **256-257, 260**
 - Rappel séquentiel (READ #, RESTORE #), **254-263**
 - Stockage direct (PRINT #, RESORE #), **256-263**
 - Stockage séquentiel (PRINT #, RESTORE #), **250-253, 256-260**
 - Suppression (PURGE), **115à116, 127**
 - Travail, **27-28, 100, 101, 144-145**
 - Transformation (TRANSFORM), **160-161**
 - Types de données, **247**
 - Touches, **99, 127-128**
 - Fin d'exécution (STOP, END), **155**
 - FIX, ordre, **55**
 - FLAG, ordre, **191-193**
 - Flèches, touches
 - Gauche/droite ([←], [→]), **15-16**
 - Haut/bas ([↑], [↓]), **21, 31, 156, 158**
 - FLOOR, fonction, **48**
 - Fonctions
 - Alphanumériques, **74-77**
 - Définies par l'utilisateur, **218-222, 269**
 - Numériques, **47-53**
 - Récurrentes, **222**
 - Statistiques, **78-89**
 - FOR...NEXT, ordre, **185-187, 269**
 - Format d'affichage
 - Fixe (FIX), **55**
 - Ingénieur (ENG), **56**
 - Scientifique (SCI), **55**
 - Standard (STD), **55**
 - Formatage, impression et affichage (IMAGE,

USING, TAB, ;, .), **224-239**
 Formatage des nombres, **54-56**
 Forme d'une fonction définie par l'utilisateur
 (DEF FN, END DEF), **218-219**
 Forme d'un sous-programme (SUB, END SUB),
204-205
 FP, fonction, **48**
 Frappe
 Aides à la frappe, **12, 21-22, 24, 26, 28**
 Correction d'erreurs, **16, 22-23, 25-26**
 FREE PORT, ordre, **105-106**

G

[G], touche, **11**
 g, témoin, **30**
 Garantie, **274-278**
 GDISP, ordre, **137-139**
 Génération de nombres aléatoires, **53**
 Global, environnement, **211**
 GOSUB, ordre
 Exécution au clavier, **150**
 Exécution dans un programme, **179**
 GOTO, ordre
 Exécution au clavier, **158**
 Exécution dans un programme, **179**
 Graphiques, affichage (GDISP), **137-139**
 Guillemets, **67, 72, 109, 269**

H

Heure, **94**
 Hiérarchie arithmétique, **64**
 En mode CALCUL, **41-42**
 Horloge
 Facteur de correction de la vitesse, **92-96**
 Mise à l'heure (SETTIME), **17-18, 92-95;**
 (ADJABS), **92-93**
 Précision, **268**
 HTD, fonction, **48**

I—J

IEEE, proposition pour le traitement des
 exceptions mathématiques, **59-60, 62**
 IF...THEN, ordre, **187-188**
 IF...THEN...ELSE, ordre, **188-189**
 IMAGE, ordre, **229-232**
 Impression d'informations, **224-234**
 Index des mots-clés, **Intérieur du dos de couverture**
 Indicateurs binaires d'état, **30, 57-60, 68, 190-201**
 Armement (SFLAG), **192**
 Effacement (CFLAG, RESET), **192-193**
 Exceptions mathématiques, **57-60, 68, 176-177,**
197
 Indicateurs système, **196-201**

Indicateurs utilisateur, **193-196**
 Test (FLAG), **191-193**
 Indicateurs système, **196-201**
 Arrondi (-11, -12), **198**
 Avertissement (-1), **196**
 Clavier Utilisateur (-9), **197**
 EXACT (-46), **200**
 Exceptions mathématiques (-4 à -8, IWL,
 OVZ, OVF, UNF, INX), **57-60, 68, 176-177,**
197
 Format d'affichage (-13, -14), **198-199**
 Minuscules (-15), **199**
 Nombre de chiffres (-17 à -20), **199-200**
 Option de base (-16), **199**
 Signal sonore (-2, -25), **197**
 Sous-tension continue (-3), **197**
 Témoin BASIC (-26), **200**
 Témoins (-57, -60 à -64), **201**
 Unité d'angle (-10), **197**
 Inexact, résultat (INX), **57-60**
 INF, fonction, **60**
 +INF, -INF, valeurs, **59-60**
 Inférieur à (<), **62**
 Information d'impression, **224-234**
 INIT, affichage, **13-14, 193, 273-274**
 Initialisation du HP-71 (INIT), **13-14, 153, 193,**
273-274
 INPUT, ordre, **241-244, 269-270**
 Insertion, curseur, **19-20, 23**
 Installation du lecteur de carte, **285**
 Instructions
 Concaténation, **22, 67, 146**
 Exécution automatique à la mise sous tension
 (STARTUP), **139-140**
 Interruption, **153**
 Labels, **146-147**
 Instructions d'expédition, **278-279**
 INT, fonction, **48**
 INTEGER, ordre, **57, 69-70**
 Intégration, programme, **76**
 Interactions programme/clavier, **129-132**
 Interruption de programme ([ATTN], PAUSE,
 WAIT), **152-155**
 Intervalle de valeurs des nombres (MINREAL,
 EPS, MINREAL), **61**
 Invalide, opération (IWL), **57-60, 68**
 INX, indicateur, **57-60**
 IP, fonction, **48**
 [I/R], touche, **21**
 IWL, indicateur, **57-60, 68**
 Jeux de caractères, **132-135, 268**

K

keys, fichier, 99, 127-128
 KEYDEF#, fonction, 125-126
 KEY#, fonction, 130-131
 KEYDOWN, fonction, 129

L

Label d'instruction, 146-147
 [LC], touche, 11, 140
 Lecteur de cartes, 284-293
 CAT CARD, ordre, 289, 293
 CHAIN :CARD, ordre, 289
 CHAIN :PCRD, ordre, 289, 292-293
 COPY CARD TO, ordre, 289, 291
 COPY TO CARD, ordre, 289, 291
 COPY TO :PCRD, ordre, 289, 292
 Fonctionnement, 289-293
 Installation, 286
 Nettoyage de la tête, 286
 Passage des cartes, 287-289
 PROTECT, ordre, 289, 292
 Retrait, 285
 RUN :CARD, ordre, 289
 RUN :PCRD, ordre, 289, 292
 UNPROTECT, ordre, 289, 292
 Lecture de la date (DATE, DATE#), 17, 90-91
 LEN, fonction, 74
 LET, ordre, 67, 269
 LEX, fichiers, 99, 160, 174
 LGT, fonction, 50
 Ligne
 Affichage, 21, 156-158
 Courante, 154, 158
 Largeur affichage et impression (WIDTH, PWIDTH), 232-233
 Longueur, 14
 Multi-instruction, 146
 Numéro, 145
 Programme, 145-149, 154-158
 Suppression (DELETE), 158-159
 Vitesse de déroulement, 26
 Limites basses (OPTION BASE), 68-69, 70, 72, 73, 270
 Limites basse, indicateur -16, 199
 [-LINE], touche, 21
 LINPUT, ordre, 244-245
 LIST, ordre, 127, 156-157
 Liste d'impression, 230
 Liste de paramètres, 204-210, 218-219, 263-264
 LN, fonction, 50
 Local, environnement, 211
 LOCK, ordre, 139
 LOG, fonction, 50

Logarithme (LGT, LOG10, LOG, LN, LOGP1), 50
 Logement, 104-108
 LOG10, fonction, 50
 LOGP1, fonction, 50

M

Magnétique, cartes et lecteur, 284-293
 Maintenance, 276-279
 Majuscules/minuscules, contrôle
 Indicateur (-15), 199
 Ordre (LC, LC ON, LC OFF), 140
 Touche ([LC]), 11, 140
 Marquage des cartes magnétiques, 286
 Mathématique, exceptions, 59-60. Voir *exceptions mathématiques*
 MAX, fonction maximum, 49
 MAXREAL, fonction, 61
 MEM, fonction, 107-108
 Mémoire
 Inutilisée (MEM), 107-108
 Logements (SHOW PORT), 108
 Modules enfichables, 103-108, 273
 Morte, 103-105, 273-274
 Ré-initialisation, 13-14, 153, 193
 Structure, 103-108
 Vive, 103-108, 273
 Mémoire vive indépendante, 104-108
 Déclaration (FREE PORT), 105-106
 Ré-incorporation à la mémoire principale (CLAIM PORT), 107
 Mémoire vive principale, 104-108
 Memory lost, message, 14
 MERGE, ordre, 128
 Messages d'entrée (INPUT), 243
 MIN, fonction minimum, 49
 MINREAL, fonction, 61
 Mise au point
 Date (SETDATE), 17, 90-91
 Définition de touche (FETCH KEY), 125
 Heure (SETTIME), 17-18, 92-95; (ADJABS), 92-93
 Touches, 15-16, 21
 Programme, 22-26, 156-159
 Vitesse de l'horloge, 94-96
 MOD, fonction, 49
 Modules enfichables, 103-108, 273
 Modulo (MOD), 49
 Mots réservés pour les noms de fichiers, 110
 Moyennes statistiques (MEAN), 83
 Multiplication, opérateur *, 47

N

NAME, ordre, 27, 101, 145
 NAN, fonction, 59-60
 NaN, valeur, 59-60, 68
 Nettoyage, 272
 NEXT, ordre, 185-187, 269
 Nom
 Fichier, 22, 109-110
 Programme (fichier BASIC), 21-22, 101, 109-110, 143-145
 Touche, 121-124
 Unité, 110-114
 Variable alphanumérique, 72
 Variable numérique, 68
 Nombre
 Aléatoires (RND), 52-53
 Arrondi, 56
 Formatage, 54-56
 Intervalle (MINREAL, EPS, MAXREAL), 61
 Norme ANSI, conformité de l'interpréteur, 268-271
 NOT, opérateur logique, 62-63
 Notation exponentielle (E), 54, 56
 NUM, fonction, 74, 77
 Numérique
 Fonctions, 47-53
 Opérateur %, 62
 Précision (OPTION ROUND), 56
 Précision des variables (REAL, SHORT, INTEGER), 57
 Numéro de série (VER#), 267

O

[OFF], [ON], touches, 13
 ON ERROR, ordre, 172
 ON...GOSUB, ordre, 181
 ON...GOTO, ordre, 181
 ON...TIMER#...GOSUB, ordre, 183-184
 ON...TIMER#...GOTO, ordre, 183-184
 Opérateurs
 Arithmétiques, 47, 64
 Hiérarchie, 64
 Logiques, 62-64
 Relationnels, 62
 Opérations non-autorisées en mode CALCUL, 46
 OPTION ANGLE DEGREES/RADIANS, ordre, 50
 OPTION BASE, ordre, 68-69, 70, 72-73, 270
 OPTION ROUND NEAR/NEG/POS/ZERO, ordre, 56
 OR, opérateur logique, 62-63
 Ordre d'évaluation, 64

Ordre de recherche des fichiers, 112
 Organisation des cartes magnétiques, 287
 Ouverture de fichier de données (ASSIGN #), 248-249
 OVF, indicateur, 57-60

P

Paramètres
 Dans un ordre CALL, 206-210
 Dans un ordre DEF FN, 218-219
 Référence, 206
 Transfert (CALL, SUB), 204-210, 263-264
 Valeur, 206
 Parenthèses
 Déclarations de tableaux, 69-72
 Expressions numériques, 64
 Fonctions alphanumériques, 74-75
 Fonctions numériques, 48-52
 Partie entière d'un nombre (IP), 48
 Partie fractionnaire d'un nombre (FP), 48
 Pas un nombre (NaN, NAN), 59-60
 PAUSE, ordre, 154
 %PCRD, mot-clé, 289, 292-293
 Personnalisation du HP-71, 120-140
 PI, fonction, 49
 Pile d'instructions, 31-32
 Pistage de l'exécution (TRACE FLOW, TRACE VARS), 166-167
 PLIST, ordre, 157
 Plus grand entier (INT ou FLOOR), 48
 Plus petit entier (CEIL), 48
 Pointeur, fichier de données, 252-253, 258
 Point d'interrogation, opérateur %, 62
 Point-virgule pour le formatage, 227
 POP, ordre, 180
 POS, fonction, 74-75
 Position du pointeur de données (RESTORE), 246-247
 Pourcentage, opérateur %, 47
 Précautions d'utilisation, 267-268
 Précision de l'horloge, 268
 PREDV, ordre, 85-86, 89
 Pression de touche, identification (KEY#), 130-131
 Simulation par programme (PUT), 131-132
 Test (KEYDOWN), 129
 PRGM, témoin, 25, 30
 PRINT, ordre, 225
 Utilisation des guillemets, 67
 PRINT USING, ordre, 230-232
 PRINT #, ordre, 250-252, 256-263
 PRIVATE, ordre, 116-117
 Privée, carte magnétique (%PCRD), 289, 292-293

Programme. Voir aussi *sous-programme*

Affichage de ligne (FETCH, LIST, PLIST, **[▲]**, **[▼]**), 21, 156-158

Ajout de lignes, 158

Arrêt, 153-155

Boucles (FOR...NEXT), 185-187, 269

Branchement conditionnel (ON...GOSUB, ON...GOTO, ON TIMER#...GOSUB, ON TIMER...GOTO), 181-184

Branchement non conditionnel (GOSUB, GOTO, RETURN, POP, CALL, CHAIN), 179-181

Chargement à partir d'une carte, 289

Cinéma, 138

Compteur, 130

Données (DATA, READ, RESTORE), 245-247, 269

Enchaînement, 151-152

Entrée, 22-25, 143-149, 269

Environnement, 153, 210-215

Exécution (RUN, CALL, CHAIN, GOSUB, **[RUN]**), 25, 149-152

Exécution conditionnelle de lignes (IF...THEN, IF...THEN...ELSE), 187-189

Exécution de programme sus carte magnétique (RUN :CARD, RUN :PCRD, CHAIN :CARD, CHAIN :PCRD), 289

Exécution pas-à-pas (**[F]** **[SST]**), 168-171

Fibonacci, 216-218

Fichier, 27, 143-144

Fin d'exécution (STOP, END), 155

Format des lignes, 145

Intégration, 76

Interruption (**[ATTN]**, PAUSE, WAIT), 152-155

Label d'instruction, 146-147

Ligne courante, 154-158

Ligne multi-instruction, 146

Mise au point, 22-26, 156-159

Nom, 21-22, 101, 109-110, 143-145

Numéros de lignes, 145

Numérotation automatique des lignes (AUTO), 149

Pistage de l'exécution (TRACE FLOW, TRACE VARS), 166-167, 171

Protection, 116-117, 127, 292

Rappel de données (READ), 246

Relance de l'exécution (**[F]** **[CONT]**, CONT), 155

Renumérotation de lignes (RENUMBER), 159

Sous-programmes, 179-184

Stockage (EDIT, NAME), 21-22, 27, 101, 143-145

Stockage de données (DATA), 245, 269

Suivi de l'exécution (TRACE FLOW, TRACE VARS), 166-167, 171

Suppression de ligne (DELETE), 158-159

Proposition IEEE pour le traitement des exceptions mathématiques, 59-60, 62

PROTECT, ordre, 289, 292

Protection de fichier (SECURE, UNSECURE, PRIVATE), 116-117, 127, 292

PURGE, ordre, 115-116, 127

PUT, fonction, 131-132

R

RAD, fonction, 52

RAD, témoin d'état, 30

RADIANS, ordre, 50, 270

RANDOMIZE, fonction, 52-53, 271

Rappel d'expression complète en mode CALCUL, 42-44

Rappel de données de programme (READ), 246, 269

Rappel direct de données (READ #), 260

Rappel séquentiel de données (READ #, RESTORE #), 254-263

READ, ordre, 246, 269

READ #, ordre, 254-257, 260

Ré-affectation de la mémoire des variables (DESTROY) 67-68

RED, fonction, 49

Redéfinition du clavier (DEF KEY), 28, 121-124

Réduction (RED), 49

Référence à une fonction définie par l'utilisateur, 220

Référence, paramètres (CALL), 206

Registre d'horloge, 182-184

Régression linéaire (LR), 84-85, 88

Ré-incorporation de mémoire vive (CLAIM PORT), 107

Ré-initialisation du HP-71 (INIT), 13-14, 193, 273-274

Relance de l'exécution (**[F]** **[CONT]**, CONT), 155

Relation, opérateurs, 62, 64

Remplacement, curseur, 14, 23

Remplacement de la batterie, 272

RENAME, ordre, 101, 115, 127

RENUMBER, ordre, 159

RES, fonction, 49

RESET, ordre, 193

RESET CLOCK, ordre, 96

Reste (RMD), 49

RESTORE, ordre, 246-247

RESTORE #, ordre, 254, 258, 262-263

Restriction de l'utilisation (LOCK), 139

Résultat (RES), 49

Résultat implicite en mode CALCUL, 38, 40

Résultat inexact (INX), 57-60

Retrait du lecteur de carte, 285

RETURN, ordre, 179
 RMD, fonction, 49
 RND, fonction, 52-53
 [RUN], touche, 25, 27, 41, 149
 RUN, ordre, 27, 149-151
 RUN :CARD, ordre, 289
 RUN :PCRD, ordre, 289

S

Saisie

- Données et expressions (INPUT), 241-244
- Données dans un tableau statistique (ADD), 80-82, 87
- Ligne d'affichage (LINPUT), 244-245
- Lignes de programme, 22-25, 145-149, 269

SCI, ordre, 55
 SDATA, fichiers, 99, 247-264
 SECURE, ordre, 116-117, 127
 Séquences d'échappement, 234-239
 Séquence de fin de ligne (ENDLINE), 234
 Séquence de touches, conventions, 13
 Service, 276-279
 SETDATE, ordre, 17, 90-91
 SETTIME, ordre, 17-18, 92-95
 SFLAG, ordre, 192
 SGN, fonction, 49
 SHORT, ordre, 57, 69-71
 SHOW PORT, ordre, 108
 Signaux sonores

- Force, 32-33
- Indicateurs (-2, -25), 33, 197

Signe (SGN), 49
 Signe moins, 64
 SIN, fonction, 52
 Sommation des données dans un tableau statistique (TOTAL), 82-83
 Sous-programmes, 151, 179-184, 203-218

- Canal de transfert (CALL, SUB), 263-264
- Environnements, 210-215
- Forme (SUB, END SUB), 204-205
- Récurrents, 214-218
- Transfert d'exécution (CALL), 205-210
- Transfert de paramètres (CALL, SUB), 204-205, 263-264

Soustraction, opérateur -, 47
 Spécification de carte magnétique (CARD, :CARD, :PCRD), 289-293
 SQB, fonction, 49
 [SST], touche, 41, 168-171
 STARTUP, ordre, 139-140
 STAT, ordre, 78-80, 87
 Statistiques, 78-89
 STD, ordre, 55

STEP, mot-clé, 185-187
 Stockage de programme (EDIT, NAME), 21-22, 27, 101, 143-145
 Stockage direct de données (CREATE, RESTORE #, PRINT #), 256-260
 Stockage séquentiel de données (CREATE, RESTORE #, PRINT #), 250-253, 256-263
 STOP, ordre, 155
 STR#, fonction, 71-77
 Structure de la mémoire, 103-108
 SUB, ordre, 204-205
 subscript, message, 68
 Supérieur à (>), 62
 Suppression de données d'un tableau statistique (DROP), 81-82
 Suppression de ligne (DELETE), 158-159
SUSP, témoin, 30, 153, 155
 Symbole de [END LINE] en mode CALCUL (ã), 42-44
 Syntaxe

- Conventions, 34
- Erreurs, 163-164
- Simplifiée, 34

Système d'exploitation, version (VER#), 267

T

TAB, ordre, 226-227
 Tableaux alphanumériques, 71-73

- Changement des dimensions sous contrôle du programme, 73
- Déclaration (DIM), 72-73, 271
- Dimensions par défauts, 73
- Limite basse (OPTION BASE), 72-73. Voir aussi *limite basse des tableaux numériques*

Tableaux numériques, 68-71

- Changement des dimensions sous contrôle du programme, 70-71
- Déclarations (DIM, REAL, SHORT, INTEGER), 69-70, 271
- Dimensions par défaut, 70
- Limite basse (OPTION BASE), 68-69, 70, 270
- Rappel (RESTORE #, READ #), 261-263
- Stockage (PRINT #), 261

Tableaux statistiques, 78-89

- Ajout de points (ADD), 80-82, 87
- Corrélations (CORR), 84
- Déclaration, (STAT), 78-80, 87
- Ecart-type (SDEV), 83-84
- Effacement d'éléments (ELSTAT), 78-80, 87
- Moyennes (MEAN), 83
- Rappel (RESTORE #, READ #), 261-263
- Régression linéaire (LR), 84-85, 88
- Sommation des points (TOTAL), 82-83

Stockage (PRINT #), **261**
 Suppression de points (DROP), **81-82**
 Valeurs probables (PREDW), **85-86, 89**
 TAN, fonction, **52**
 Témoins, **15-16, 20, 22, 30**
 Témoins de contenu (symboles ← et →), **15-16, 22, 30**
 Témoins d'indicateurs (-57, -60 à -64), **201**
 Témoins numériques d'indicateurs, **30**
 Tête de lecture, entretien, **286**
 TEXT, fichiers, **99, 247-264**
 THEN, mot-clé, **187-189**
 Touches, **11-13, 15-16, 21**
 Aide à la frappe, **12**
 Définies par l'utilisateur, **28-29, 121-124**
 Exécution immédiate, **12**
 Fonctions primaire et secondaires, **11**
 TRACE FLOW, ordre, **166-167**
 TRACE OFF, ordre, **168**
 TRACE VARS, ordre, **167**
 Traitement des erreurs, **16, 23, 25-26, 43-46, 165-171**
 Avec les touches de mise au point, **23, 25-26**
 Avec TRACE FLOW et TRACE VARS, **166-167, 171**
 Effacement de l'affichage, **16**
 En mode CALCUL, **42-46**
 Exceptions mathématiques, **58, 269-270**
 Transfert de canal (CALL, SUB), **263-264**
 TRANSFORM, ordre, **160-161**
 TRAP, fonction, **59-60, 68**

U

UNF, indicateur, **57-61**
 Unité
 Défaut, **111**
 Noms, **110-114**
 Unité d'angle, indicateur (-10), **197**
 UNPROTECT, ordre, **289-292**
 UNSECURE, ordre, **116-117, 127**
 UPRC#, fonction, **75**
USER, **1 USER**, touches, **28-29, 126**
 USER, USER OFF, USER ON, ordres, **126**
USER, indicateur, **30**
 Utilisation des cartes magnétiques, **287-293**

V

VAL, fonction, **74, 76**
 Valeur absolue (ABS), **48**
 Valeurs booléennes, **62**
 Valeurs infinies (INF, +INF, -INF), **59-60**
 Valeurs probables statistiques (PREDW), **85-86, 89**

Variables

Alphanumériques, **71-73, 268**
 Inexistantes, **68, 72**
 Noms, **68, 72**
 Partage entre clavie et programmes, **67**
 Précision numérique (OPTION ROUND), **56**
 Tableau, **68-73**
 Valeurs par défaut, **67-68, 72, 268**
 VER#, fonction, **126**
 Vérification du fonctionnement, **273-274**
 Version du système d'exploitation (VER#), **267**
VIEW, touche, **126**
 Vitesse de déroulement de l'affichage, **26**
 Virgule dans les listes d'arguments en mode CALCUL, **40-41**
 Virgule pour le formatage des sorties, **227**

W

WAIT, ordre, **154**
 WIDTH, ordre, **232-233**
 WINDOW, ordre, **135-136**
 Workfile, **27-28, 100, 101, 144-145**

Index des mots-clés

Cet index liste les mots-clés du HP-71 par catégories et donne le numéro de la page de ce manuel où ce mot-clé est introduit. Certains mots-clés apparaissent dans plusieurs catégories.

Entrée/modification de programme		Contrôle de programme (suite)		Opérateurs logiques et de relation		Mathématiques générales (suite)	
AUTO	149	ON TIMER #	182	AND	63	OPTION ROUND	56
DELÈTE	158	ON...GOSUB	191	EXOR	63	OVF	57
EDIT	143	ON...GOTO	181	NOT	63	RANDOMIZE	52
FETCH	157	ON...RESTORE	—	OR	63	RED	49
LIST	157	PAUSE	154	=	62	RES	49
NAME	145	POP	180	◇	62	RMD	49
PLIST	157	RETURN	179	<	62	RND	52
PRIVATE	116	STOP	155	<=	62	SGN	49
REM (!)	22	SUB	204	>	62	SQR	49
RENUMBER	159	WAIT	154	>=	62	SQRT	49
SECURE	116			?	62	UNF	57
TRANSFORM	160						
UNSECURE	116	Recherche et traitement d'erreurs		Opérateurs arithmétiques		Opérations logarithmiques	
@	146	CONT	155	+	47	EXP	50
		DEFAULT	58	-	47	EXPM1	50
Exécution de programme		ERRL	175	*	47	EXPONENT	50
CALL	151	ERRM#	175	/	47	LGT	50
CHAIN	151	ERRN	173	÷	47	LN	50
CONT	155	ON ERROR GOSUB	172	DIV (!)	47	LOG	50
RUN	149	ON ERROR GOTO	172	^	47	LOG	50
		PAUSE	154	%	47	LOG10	50
		TRACE	166			LOGP1	50
				Mathématiques générales		Opérations trigonométriques	
Contrôle de programme		Allocation d'espace mémoire		ABS	48	ACOS	52
BYE	184	CLAIM PORT	107	CEIL	48	ACS	52
CALL	205	DESTROY	67	CLASS	60	ANGLE	52
CHAIN	180	DIM	69	DVZ	57	ASIN	52
DEF FN	218	FREE PORT	105	EXPONENT	50	ASN	52
END	155	INTEGER	70	FACT	49	ATAN	52
END DEF	219	MEM	107	FLOOR	48	ATN	52
END SUB	204	OPTION BASE	68	FP	48	COS	52
FN	220	REAL	69	INT	48	DEG	52
FOR...NEXT	185	SHORT	70	INX	57	DEGREES	50
GOSUB	179	SHOW PORT	108	IP	48	OPTION ANGLE	50
GOTO	179	STAT	79	IWL	57	RAD	52
IF...THEN...ELSE	187			LET	67	RADIANS	50
OFF	13			MAX	49	SIN	52
OFF ERROR	172			MIN	49	TAN	52
OFF TIMER	184			MOD	49		
ON ERROR GOSUB	172						
ON ERROR GOTO	172						

Statistiques		Entrée/Sortie		Gestion de fichiers		Personnalisation	
		(suite)		(suite)		et contrôle	
ADD	80	ENG	56	NAME	145	du clavier	
CLSTAT	79	FIX	55	PRIVATE	116	ADDR#	—
CORR	54	GOISP	137	PROTECT	292	CHARSET	132
DROP	81	GOISP#	137	PURGE	115	CHARSET#	132
LR	54	IMAGE	231	RENAME	115	CONTRAST	29
MEAN	83	INPUT	241	SECURE	116	DEF KEY	121
PREDV	85	KEYDOWN	129	SHOW PORT	108	DELAY	26
SDEV	83	LC	140	TRANSFORM	160	DTH#	48
STAT	79	LINPUT	244	UNPROTECT	292	FETCH KEY	125
TOTAL	82	ON...RESTORE	—	UNSECURE	116	FIX	55
		PLIST	157			HTD	48
Constantes		PRINT	225	Heure et date		IMAGE	231
EPS	61	PRINT USING	230	ADJABS	93	KEY	121
INF	59	PRINT #	250	ADJUST	95	KEY#	130
MAXREAL	61	PUT	131	AF	95	KEYDEF#	125
MINREAL	61	PWIDTH	232	DATE	91	KEYDOWN	129
NAN	59	READ	246	DATE#	91	LC	140
PI	49	READ #	254	EXACT	96	LOCK	139
		RESTORE	246	RESET CLOCK	96	PEEK#	—
Chaînes		RESTORE #	254	SETDATE	90	POKE	—
alphanumériques		SCI	55	SETTIME	92	STARTUP	139
^	73	STD	55	TIME	94	USER	126
CHR#	75	TAB	226	TIME#	94	WINDOW	135
LEN	74	UPRC#	75				
NUM	74	USER	126	Etats du système			
POS	74	WIDTH	232	et indicateurs			
STR#	75	WINDOW	135	CFLAG	192		
VAL	74			DEFAULT	58		
VER#	267	Graphiques		DEGREES	50		
		GOISP	137	DELAY	26		
Entrée/Sortie		GOISP#	137	DVZ	57		
ASSIGN #	248	Gestion de fichiers		FLAG	191		
BEEP	32	ADDR#	—	INX	57		
BEEP OFF	32	CAT	118	IVL	57		
BEEP ON	32	CAT#	—	OPTION ANGLE	50		
CONTRAST	29	CLAIM PORT	107	OPTION BASE	68		
COPY	112	COPY	112	OPTION ROUND	56		
CREATE	248	CREATE	248	OVF	57		
DATA	245	EDIT	143	RADIANS	50		
DELAY	26	FREE PORT	105	RESET	193		
DISP	225	MEM	107	SFLAG	192		
DISP USING	230	MERGE	128	TRAP	59		
DISP#	136			UNF	57		
ENDLINE	234						

HP-71

Manuel d'utilisation

Addendum

Cet addendum contient des informations concernant l'utilisation de deux mots-clés du HP-71 tels que décrits dans le *Manuel d'utilisation* du HP-71, référence 00071-90002, daté de janvier 1984.

Utilisation de ON...GOTO et ON...GOSUB

Page 181 du *Manuel d'utilisation* du HP-71. Dans les deux situations décrites ci-dessous, l'exécution de ON...GOTO ou ON...GOSUB ne provoque pas de branchement de programme mais peut provoquer une action similaire à celle de ON...RESTORE. Dans aucun cas, la mémoire n'est altérée.

Situation 1. N'utilisez pas ON...GOTO/GOSUB lorsque OPTION ROUND POS ou OPTION ROUND NEG est actif.

Pour éviter ce problème, insérer une instruction OPTION ROUND NEAR ou OPTION ROUND ZERO dans tout programme contenant ON...GOTO/GOSUB. Positionnez cette instruction OPTION ROUND NEAR/ZERO de telle façon qu'elle soit active lors de l'exécution de ON...GOTO/GOSUB. De ce fait, aucune instruction OPTION ROUND POS/NEG active ne pourra interférer avec l'exécution de ON...GOTO/GOSUB, que OPTION ROUND POS/NEG existe ailleurs dans votre programme ou qu'il soit conservé en mémoire permanente à partir d'un environnement précédent.

Situation 2. L'expression de pointeur utilisée dans une instruction ON...GOTO/GOSUB (ON *expression de pointeur* GOTO/GOSUB) ne doit pas comporter d'autres opérateurs que +, -, *, / ou DIV et ne doit pas comporter d'autres fonctions (y compris les fonctions définies par l'utilisateur, les fonctions trigonométriques, SQRT etc.).

Pour éviter ce problème, utilisez uniquement des variables simples dans l'expression de pointeur avec uniquement les opérateurs +, -, *, / et DIV. Si vous désirez que le pointeur comporte une fonction ou un autre opérateur, affectez d'abord l'expression à une variable simple puis utilisez cette dernière comme expression de pointeur dans l'instruction ON...GOTO/GOSUB.

Exemples: Ces exemples illustrent les façons correctes et incorrectes d'utiliser une expression comme pointeur dans une instruction ON...GOTO/GOSUB, lorsque cette expression comporte une fonction et/ou un opérateur autre que +, -, *, / et DIV.

Incorrect:

```
ON FNJ(T) ^ SIN(M) GOSUB 500, 600,  
700
```

Cette instruction ON...GOSUB ne provoque pas de branchement de programme.

Correct:

```
A = FNJ(T) ^ SIN(M)  
ON A GOSUB 500, 600, 700
```

Cette instruction ON...GOSUB provoque un branchement de programme; elle est correctement exécutée.

Correction au manuel

Page 20 en bas de page, ajoutez: Placez à nouveau le HP-71 en mode BASIC en appuyant sur  **CALC**. (La séquence  **CALC** est une *bascule*. Elle permet de passer entre les modes CALCUL et BASIC.)



Hewlett-Packard France
Division Informatique Personnelle
91947 Les Ulis Cédex

European Headquarters
150, Route Du Nant-D'Avril
P.O. Box, CH-1217 Meyrin 2
Genève - Suisse

Portable Computer Division
1000 N.E. Circle Blvd.,
Corvallis, OR 97330
U.S.A.

Introduction à l'ordinateur portable HP-71 (page 6)

- 1: Généralités (page 10)**
- 2: Calculs avec le HP-71 (page 36)**
- 3: Variables: simples et en tableaux (page 66)**
- 4: Fonctions statistiques (page 78)**
- 5: Horloge et calendrier (page 90)**
- 6: Opérations sur les fichiers (page 98)**
- 7: Personnalisation du HP-71 (page 120)**
- 8: Rédaction et exécution de programmes (page 142)**
- 9: Conditions d'erreur (page 162)**
- 10: Branchement, boucle et exécution conditionnelle (page 178)**
- 11: Indicateurs d'état (page 190)**
- 12: Sous-programmes et fonctions définies par l'utilisateur (page 202)**
- 13: Formats d'impression et d'affichage (page 224)**
- 14: Stockage et rappel de données (page 240)**

- A: Informations générales (page 266)**
- B: Accessoires (page 282)**
- C: Utilisation du lecteur de cartes magnétiques HP 82400A (page 284)**

- Index des sujets (page 294)**
- Index des mots-clés (dos de couverture)**



Portable Computer Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.