

**PPC Paris**

---

**J P C R o m**

---

**Manuel d'utilisation**

---

Octobre 1989  
Révision E



# Table des matières

<b>Table des matières</b>	<b>3</b>
<b>1 Introduction</b>	<b>7</b>
<b>2 Différences principales entre la version A et la version B</b>	<b>8</b>
<b>3 Différences principales entre la version B et la version C</b>	<b>9</b>
<b>4 Différences principales entre la version C et la version D</b>	<b>11</b>
<b>5 Différences principales entre la version D et la version E</b>	<b>12</b>
<b>6 Référence</b>	<b>13</b>
ADBUF\$ . . . . .	14
ADCREATE . . . . .	16
ADDELETE . . . . .	18
ADFIND . . . . .	19
ADGET . . . . .	20
ADPUT . . . . .	22
ADSIZE . . . . .	23
ARR . . . . .	24
ASC\$ . . . . .	25
ATH\$ . . . . .	26
ATTN . . . . .	27
BELL . . . . .	28
BOLD . . . . .	29
BOX . . . . .	30
CASE . . . . .	31
CENTER\$ . . . . .	32
CESURE . . . . .	33
COMB . . . . .	34
CONTRAST . . . . .	35
CSIZE . . . . .	36
DATEADD . . . . .	37
DATESTR\$ . . . . .	38
DBLIST . . . . .	40
DDAYS . . . . .	42
DDIR . . . . .	43
DMY . . . . .	45
DOW . . . . .	46
DOW\$ . . . . .	48

DRAW	50
EDIT	51
ENDUP	53
ENDUP\$	54
ENTRY\$	55
ESC\$	57
EXECUTE	59
EXIT	60
FILEPOS	61
FILESIZE	63
FIND	64
FINPUT	66
FKEY	70
FORMAT\$	71
FPRIM	72
FRAC\$	73
FRAME	74
GDUMP	75
GEND	76
GENLEN	77
GENPOS	79
GENRPLC\$	81
GINIT	83
GLINE	84
GPSET	86
HMS	87
HMSADD	88
HMSSUB	89
HR	90
HTA\$	91
IDRAW	92
IF ... THEN ... ELSE ... END IF	93
IMOVE	95
INVERSE	96
KA	97
KEYWAIT\$	101
LABEL	102
LDIR	103
LEAVE	104
LEX	105
LINETYPE	107
LOOP ... END LOOP	108
LORG	109
MAP	110
MAP\$	111
MARGIN	112
MAXD	113
MAXM	115
MDY	116

MEMD . . . . .	117
MEMM . . . . .	118
MENU . . . . .	119
MERGE . . . . .	120
MODE . . . . .	122
MOVE . . . . .	123
NEXTOP\$ . . . . .	124
NLOOP . . . . .	125
NPRIM . . . . .	127
OPCODE\$ . . . . .	128
PAGELEN . . . . .	130
PAINT . . . . .	132
PARPOLL . . . . .	134
PBLIST . . . . .	136
PCR . . . . .	138
PDIR . . . . .	139
PEEK\$ . . . . .	140
PEN . . . . .	141
PENDOWN . . . . .	142
PENUP . . . . .	143
PERF . . . . .	144
PFF . . . . .	145
PGCD . . . . .	146
PHI . . . . .	147
PLF . . . . .	148
PLOTTER IS . . . . .	149
POKE . . . . .	150
POSI . . . . .	151
PPCM . . . . .	153
PRIM . . . . .	154
RED\$ . . . . .	155
REDUCE\$ . . . . .	156
RENUMREM . . . . .	157
REPEAT ... UNTIL . . . . .	159
REPLACES\$ . . . . .	161
ROMAN . . . . .	163
RREC\$ . . . . .	165
SELECT ... CASE ... END SELECT . . . . .	166
SHRINK . . . . .	168
SLEEP . . . . .	169
SPACE\$ . . . . .	170
SRQ . . . . .	171
STACK . . . . .	173
STARTUP\$ . . . . .	174
SYSEDIT . . . . .	175
TEDIT . . . . .	177
TICLEN . . . . .	180
TOKEN . . . . .	181
UNDERLINE . . . . .	182

VARSWAP . . . . .	183
WHILE ... END WHILE . . . . .	184
WRAP . . . . .	185
WREC . . . . .	186
XAXIS . . . . .	187
XEDIT . . . . .	188
YAXIS . . . . .	198
<b>7 Les autres caractéristiques de JPC Rom</b>	<b>199</b>
La fonction VER\$ . . . . .	200
Tabulations assembleur . . . . .	201
Correction et Mode CALC . . . . .	202
Position du curseur . . . . .	203
Accélération du curseur . . . . .	204
Initialisation au Memory-Lost . . . . .	205
Reconnaissance des types de fichiers . . . . .	206
Les messages du module HP-IL . . . . .	207
<b>8 Les auteurs</b>	<b>211</b>
<b>9 Messages de JPC Rom</b>	<b>215</b>
<b>10 Vous avez trouvé une erreur</b>	<b>217</b>

# Chapitre 1

## Introduction

Ce document est le manuel de JPC Rom. Il s'agit d'un outil de référence pour les utilisateurs de JPC Rom. L'utilisateur est supposé être déjà familiarisé avec le fonctionnement du HP-71 et la programmation en langage Basic.

JPC Rom combine plus de 130 nouveaux mots-clefs dans un seul fichier de 44 Ko. Ceci apporte les extensions les plus utiles, et augmente la performance globale du HP-71.

Ce fichier Lex est constitué par les contributions des membres de PPC-Paris. Tout ce travail a été accompli par des membres bénévoles.

Ce manuel représente, à lui seul, une somme de travail colossale. Toutes les remarques, suggestions ou critiques visant à améliorer son contenu seront bienvenues. Il a été réalisé par Pierre David, Jean-Jacques Dhénin, Michael Markov et Janick Taillandier.

Précisons enfin que toute l'impression a été faite par un HP-71 et une imprimante LaserJet.

Nous espérons que vous apprécierez le résultat. N'hésitez à nous faire part de vos réflexions :

PPC Paris

B.P. 604

75028 Paris Cedex 01

France

PPC Paris n'offre aucune garantie en ce qui concerne les procédures décrites, le contenu de cette bibliothèque ou l'application de ces 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 PPC Paris ni toute autre partie) est responsable de l'ensemble des coûts de correction et de tous les dommages en résultant. PPC Paris n'est pas responsable des dommages résultants ou connectés à l'utilisation de ces procédures ou programmes.

## Chapitre 2

# Différences principales entre la version A et la version B

Le passage de la version A00 à la fonction B00 (voir la fonction VER\$ dans *les caractéristiques additionnelles de JPC Rom*) résulte d'un certain nombre de corrections, modifications et améliorations. Les principales sont résumées ci-dessous :

### Corrections

MARGIN voit maintenant son paramètre limité à 96.

le mot-clef FIND trouvait parfois des chaînes qui n'existaient pas (voir *JPC 45*). Janick Taillandier a arrangé cela.

COMB et ARR ont été réécrits par Guy Toublanc pour ne plus utiliser la factorielle, mais un algorithme multiplicatif.

COMB a été modifié pour ne plus renvoyer de résultat faux lorsqu'on utilise  $S=S+COMB(n, \theta)$ .

La version de FINPUT publiée dans *JPC* sortait lorsqu'on appuyait sur la touche [ f ] [CONT]. Cela n'était guère normal.

### Nouveaux mots-clefs et nouvelles caractéristiques

La fonction DATESTR\$ a été ajoutée pour convertir le nouveau format de dates de JPC Rom.

Les mots-clefs de programmation structurée ont été ajoutés à cette version.

### Modifications et améliorations

Les fonctions de calculs de dates ont été réécrites pour utiliser le nouveau format de dates de JPC Rom aussi bien que le format de dates standard.

Le mot-clef KSPEED a été enlevé. L'accélération du curseur subsiste, mais la vitesse de répétition est maximale et le délai entre la première pression de touche et la première répétition n'est plus modifiable.

La fonction POSI a été modifiée pour accepter des paramètres aussi bien numériques qu'alphanumériques.

La séquence envoyée à l'imprimante par BOLD a été modifiée pour être compatible avec toutes les imprimantes *PCL*, notamment les ThinkJet et LaserJet.

FF a été renommé en PFF, LF en PLF, PL en PAGELEN, CR en PCR. FPRM a été renommé en FPRIM et NPRM en NPRIM. HMS+ a été renommé en HMSADD et HMS- en HMSSUB.

### Note

Toutes ces améliorations ou corrections ont été faites en conservant la compatibilité des programmes écrits avec l'ancienne version de JPC Rom. Ainsi, vos programmes écrits avec l'ancien JPC Rom sont-ils totalement compatibles avec le nouveau JPC Rom.

## Chapitre 3

# Différences principales entre la version B et la version C

Le passage de la version B00 à la fonction C (voir la fonction VER\$ dans *les caractéristiques additionnelles de JPC Rom*) résulte d'un certain nombre de corrections, modifications et améliorations. Les principales sont résumées ci-dessous :

### Corrections

Les mots-clefs de programmation structurée ne fonctionnaient pas lorsqu'une remarque suivait directement le début de la boucle, comme par exemple :

```
10 WHILE 1 !
20 END WHILE
```

Idem pour SELECT, CASE, etc. Signalé par Henri Kudelski en Suisse et Gérard Kossman en France.

Les fonctions STARTUP\$ et ENDUP\$ ne retournaient pas convenablement leur résultat, ce qui avait pour effet de provoquer une erreur Data type lorsque le programme suivant était exécuté :

```
10 DESTROY ALL
20 DIM S$
30 DIM S$[LEN(STARTUP$)]
```

Idem pour ENDUP\$. Signalé par Tapani Tarvainen en Finlande.

La touche [f] [BACK] retrouve sa signification normale dans la pile de commande en mode CALC, ce qui permet d'éditer librement les expressions dans les lignes de commandes. Signalé par Michael Markov aux Etats-Unis et Tapani Tarvainen en Finlande.

Après une configuration (par exemple, COPY d'un fichier Lex en mémoire, LEX ON/OFF), les tabulations assembleur étaient activées lorsqu'on utilisait EDTEXT.

Le mot-clef STACK a été remplacé par celui de Henri Kudelski en Suisse qui fait moins de misères au lancement du programme ML.

### Nouveaux mots-clefs et nouvelles caractéristiques

Le mot-clef SYSEEDIT a été ajouté, ainsi que OPCODE\$ et NEXTOP\$. OPCODE\$ et NEXTOP\$ ont été écrits par Jean-Jacques Dhénin. SYSEEDIT a été écrit par Pierre David et Janick Taillandier.

La fonction FILESIZE de Henri Kudelski a été ajoutée.

Le gestionnaire de carnet d'adresses KA et ses fonctions d'accès programmables (ADCREATE, ADDELETE, ADFIND, ADGET, ADPUT et ADSIZE) ont été ajoutées. Ces mots-clefs ont été écrits par Pierre David.

La fonction KEYWAIT\$ (de Hewlett-Packard) a été ajoutée. Ses numéros d'Id et de token n'ont pas été modifiés par rapport à ceux de la User's Library.

Le mot-clef ROMAN a été ajouté pour permettre l'accès au jeu de caractères *Roman*. Ce mot-clef a été écrit par Pierre David et Janick Taillandier.

JPC Rom reconnaît maintenant les types de fichiers non standard lors d'un CAT, tels ceux du HP-41 ou du HP-75, mais en plus les fichiers non standard produits par le HP-71. Ce Lex a été écrit par Jan Buitenhuis des Pays Bas et Janick Taillandier de France.

# Différences principales entre la version B et la version C

## Modifications et améliorations

*JPC Rom* s'appelait avant *JPCLEX*.

BLIST a été renommé en DBLIST, à cause de l'homonymie avec le mot-clef du Lex BREAK de la User's Library.

SWAP a été renommé en VARSWAP.

La fonction INV\$ a été supprimée, et ses fonctionnalités reportées sur INVERSE.

La syntaxe de INVERSE et de PAINT ont été étendues pour offrir plus de souplesse.

La syntaxe de SPACE\$ a été étendue pour la généraliser et l'étendre à la répétition de chaînes quelconques.

ENABLE et DISABLE ont été renommés en LEX ON/OFF à cause du conflit avec ENABLE dans le module HP-IL.

Les fonctions REPLACE\$ et RPLC\$ sont désormais une seule fonction REPLACE\$. Si il n'y a que trois paramètres, ou si le quatrième est numérique, les fonctionnalités sont celles de l'ancienne RPLC\$. Si le quatrième paramètre est alphanumérique, c'est le *joker* utilisé dans l'ancienne fonction REPLACE\$.

DBLIST et PBLIST ont été réécrits pour autoriser l'indentation des structures et la redirection dans un fichier.

Les fonctions supprimées sont repérées par `obsolete` quand elles apparaissent dans un programme. Lorsque ces programmes sont exécutés, ils génèrent l'erreur JPC ERR:Removed Keyword (message numéro 16).

## Note

Toutes ces améliorations ou corrections ont été faites en conservant la compatibilité des programmes écrits avec l'ancienne version de JPC Rom. Ainsi, vos programmes écrits avec l'ancien JPC Rom sont-ils totalement compatibles avec le nouveau JPC Rom.

## Chapitre 4

# Différences principales entre la version C et la version D

Le passage de la version C à la version D (voir la fonction VER\$ dans *les caractéristiques additionnelles de JPC Rom*) résulte d'un certain nombre de corrections, modifications et améliorations. Les principales sont résumées ci-dessous :

### Corrections

Le jeu de caractères étendu Roman disparaissait à chaque mise sous tension.

Le mode *Tabulations assembleur* était réactivé régulièrement, comme par exemple à chaque mise sous tension.

La fonction POSI renvoyait une valeur incorrecte (1) lorsqu'on utilisait POSI ("", x) avec  $x < 6$ . Signalé par Joe Horn aux Etats-Unis.

La nouvelle version de DBLIST n'acceptait pas la syntaxe suivante : DBLIST 1000 INDENT 4.

Le mot-clef FIND ne trouvait parfois pas l'occurrence d'une lettre dans une ligne. Signalé par Henri Kudelski en Suisse et Claudio Benski en France.

Les fonctions de calcul de dates (telle que DOW par exemple) n'acceptaient pas le 29 février des années bissextiles lorsque le dernier chiffre de l'année n'était pas multiple de 4. Signalé avec beaucoup de précision par Laurent Istria en France.

En mode *consultation de fiche*, avec KA, la séquence de touche suivante [ f ] [EDIT], puis [ENDLINE] provoquait parfois l'entrée et la sortie immédiate du mode *édition de fiches*. Signalé par Henri Kudelski en Suisse.

### Nouveaux mots-clefs et nouvelles caractéristiques

Les mots-clefs DDIR et PDIR ont été ajoutés.

### Modifications et améliorations

Le désassemblage par OPCODE\$ ou SYSEDIT inversait les mnémoniques RSI et PC=(A), comme cela a été révélé par le désassemblage de la Rom du HP-28C.

### Note

Toutes ces améliorations ou corrections ont été faites en conservant la compatibilité des programmes écrits avec l'ancienne version de JPC Rom. Ainsi, vos programmes écrits avec l'ancien JPC Rom sont-ils totalement compatibles avec le nouveau JPC Rom.

## Chapitre 5

# Différences principales entre la version D et la version E

Le passage de la version D à la version E (voir la fonction VER\$ dans *les caractéristiques additionnelles de JPC Rom*) résulte d'un certain nombre de corrections, modifications et améliorations. La version E ajoute surtout deux énormes pièces, à savoir l'éditeur et le graphique. Les principales différences sont résumées ci-dessous :

### Corrections

L'adjonction du pseudo-mot-clef obsolete posait des problèmes lorsqu'on examinait les mots-clefs de JPC Rom à l'aide d'un programme Basic.

Les mots-clefs de programmation structurée souffraient d'un problème lorsqu'ils étaient suivis d'une remarque, comme dans :

```
10 LEAVE !
```

C'était vrai aussi pour WHILE, IF, et ELSE. Erreur signalée par Joseph A. Dickerson aux Etats-Unis.

La structure SELECT ne fonctionnait pas très bien non plus dans le cas suivant par exemple :

```
10 SELECT 0.4  
20 CASE >0.2 @ BEEP  
30 END SELECT
```

Le BEEP n'était jamais exécuté. Erreur signalée par Laurent Istria et Jean Horville.

### Nouveaux mots-clefs et nouvelles caractéristiques

L'éditeur en assembleur de Pierre David et Janick Taillandier a été ajouté à JPC Rom. Il contient les mots-clefs suivants : XEDIT, TEDIT, GENPOS, GENLEN, GENRPLC\$ et FILEPOS. Les tabulations en assembleur fonctionnent dorénavant avec XEDIT et TEDIT et non plus avec EDTEXT.

Les mots clefs de graphique (BOX, CSIZE, DRAW, FRAME, GDUMP, GEND, GINIT, IDRAW, IMOVE, LABEL, LDIR, LINETYPE, LORG, MOVE, PENDOWN, PENUP, PEN, PLOTTER, TICLEN, XAXIS et YAXIS) de Pierre David ont été ajoutés.

### Modifications et améliorations

Le mot-clef FIND, malgré les corrections successives, ne donnait toujours pas satisfaction. Il a été entièrement réécrit, et de plus il a bénéficié des chaînes génériques de XEDIT et de deux paramètres supplémentaires. Par Pierre David et Janick Taillandier.

Le paramètre *adresse* dans les mots-clefs OPCODE\$, NEXTOP\$, PEEK\$, POKE, et SYSEDIT peut maintenant être exprimé en décimal, pour simplifier les calculs et éviter des conversions superflues.

Le mot-clef LEAVE autorise maintenant un paramètre optionnel spécifiant le nombre de boucles englobantes dont il faut sortir.

### Note

Toutes ces améliorations ou corrections ont été faites en conservant la compatibilité des programmes écrits avec l'ancienne version de JPC Rom. Ainsi, vos programmes écrits avec l'ancien JPC Rom sont-ils totalement compatibles avec le nouveau JPC Rom.

## **Chapitre 6**

## **Référence**

# ADBUF\$

ADBUF\$ (adresse de buffer) renvoie l'adresse du buffer spécifié par son numéro d'identification.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

ADBUF\$ ( *identificateur de buffer* )

## Exemple

A\$=ADBUF\$("BFC")

Place l'adresse du "Lex Buffer" dans la variable A\$.

## Paramètre d'entrée

Élément	Description	Restriction
identificateur de buffer	Expression alphanumérique contenant des chiffres hexadécimaux.	3 chiffres en majuscules ou minuscules.

## Opération

### Les buffers :

Les buffers sont des emplacements dans la mémoire du HP-71 destinés à stocker des informations plus volatiles que celles contenues dans les fichiers, ou utilisées seulement par le système.

Ils sont utilisés par les programmes en assembleur et par le système d'exploitation. La table ci-dessous liste quelques uns des buffers utilisés par le système :

Id	Description
808	Chaîne utilisée par STARTUP
83B	Paramètres du graphique
83C	Paramètres du graphique (raster)
83D	Valeur de MARGIN
83E	Chaîne utilisée par ENDUP
BFB	Jeu de caractères défini par CHARSET
BFC	Adresses des fichiers Lex reconnus

Les buffers sont constitués d'un en-tête de longueur fixe (7 quartets), puis d'une zone de données contenant les informations utiles. Le codage de la zone d'en-tête est :

1 quartet : le nombre d'adresses à actualiser quand la mémoire bouge,

3 quartets : le numéro d'identification du buffer,

3 quartets : la longueur en quartets de la partie «données» du buffer.

Les buffers sont des zones très mobiles. Leur adresse change souvent, notamment quand :

- un fichier est créé, détruit, réduit ou agrandi,
- un autre buffer est détruit, réduit ou agrandi.

Le système maintient en outre des buffers temporaires, c'est à dire que leur Id n'est pas attribué définitivement, mais chaque programme (en assembleur) qui le souhaite peut chercher un Id disponible (entre E00 et FFF) et l'utiliser pour ses besoins propres. XEDIT, par exemple, utilise ce principe pour stocker sa table d'accès rapide et les chaînes génériques.

### La fonction ADBUF\$ :

ADBUF\$ retourne l'adresse du buffer dont le numéro d'identification est donné. Si le buffer n'est pas trouvé, la chaîne vide est renvoyée.

L'adresse donnée par la fonction correspond au début de la partie en-tête du buffer. La partie «données» se situe 7 quartets plus loin.

## Références

*JPC 22* (page 35) Première version de ADBUF\$ par Michel Martinet et Pierre David.

*JPC 23* (page 30) Les buffers du HP-71, par Pierre David. Article d'initiation, petits utilitaires.

*JPC 27* (page 34) deuxième version par Michel Martinet.

*Internal Design Specification* Volume I, Chapitre 3.5.3. En anglais.

## Mots-clés associés

DTH\$, HTD, PEEK\$, POKE, ADDR\$

## Auteurs

Pierre David, Michel Martinet

# ADCREATE

Le mot-clef programmable ADCREATE crée un fichier d'adresses vide.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

ADCREATE <i>fichier</i> ADCREATE <i>fichier</i> , <i>mot de passe</i>
--

## Exemples

ADCREATE ESSAI Crée un fichier d'adresses de nom ESSAI, sans mot de passe.

ADCREATE A\$, "passe" Crée un fichier d'adresses, et lui affecte le mot de passe «passe».

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Nom de fichier avec un spécificateur de périphérique optionnel.
mot de passe	Expression alphanumérique. Défaut : Aucun mot de passe	Seuls les 8 premiers caractères sont pris en compte.

## Opération

### Les fichiers d'adresses

Le mot-clef KA est un mode *carnet d'adresses* interactif. KA stocke les adresses dans un fichier de type ADRS. KA étant exclusivement interactif, JPC Rom contient donc les fonctions ADCREATE, ADGET, ADPUT, ADELETE, ADSIZE et ADFIND pour accéder par programme aux adresses mémorisées.

Les fichiers d'adresses peuvent être vus comme un ensemble de fiches contenant chacune une adresse. Par exemple :

```

-----
|               Nom               |
|-----|
|               Nom               |---|
|-----| 1 | |
|---|---|---|
|               Nom               |---| 2 |
|-----| 1 |---|
|               Tél               |---| 3 |
|-----| 2 |---|
|               Ligne 1           |---| 4 |
|-----| 3 |---|
|               Ligne 2           |---|
|-----| 4 |---|
|               Ligne 3           |---|re |
|-----|
|               Ligne 4           |---|
|-----|re |
|               Note              |---|
|-----|
|               Critère           |
|-----|
  
```

Dans cet exemple, le fichier est constitué de trois fiches. Voyons maintenant de quoi se composent les fiches.

## Les fiches

Chaque fiche est composée de 8 lignes, organisées comme suit :

- le nom et le prénom, séparés par un caractère /,
- le numéro de téléphone,
- 4 lignes pour mémoriser l'adresse,
- une ligne pour noter des informations générales, et
- une ligne pour stocker un critère, que vous pouvez utiliser pour vos programmes personnels.

Le première ligne contient le nom et le prénom, séparés par un caractère barre oblique (/). Les fonctions de gestion du carnet d'adresses le placeront pour vous si vous omettez de le mettre.

Les fiches sont toujours triées par ordre alphabétique. Ce tri est automatiquement fait lors de l'insertion dans le fichier, soit avec KA, soit avec ADPUT.

## Les fonctions de gestion d'adresses

Elles sont au nombre de 6 :

- ADCREATE crée un fichier de type ADRS en lui affectant ou non un mot de passe,
- ADGET lit une adresse (une fiche) et la place dans un tableau alphanumérique,
- ADPUT stocke une fiche dans le fichier d'adresses,
- ADDELETE détruit une fiche connaissant son numéro,
- ADSIZE renvoie le nombre de fiches dans le fichier,
- et ADFIND localise une fiche dans le fichier et renvoie son numéro.

Toutes ces fonctions admettent un mot de passe optionnel. Si le mot de passe est présent dans le fichier, l'utilisateur doit fournir le mot de passe avec chaque fonction. Si le mot de passe n'est pas mémorisé dans le fichier, le paramètre optionnel n'est pas pris en compte (autrement dit, sa présence n'est pas vérifiée).

## Le mot-clef ADCREATE

ADCREATE crée un fichier d'adresses (de type ADRS) vide, en lui affectant éventuellement un mot de passe initial.

ADCREATE refuse de créer le fichier si celui-ci existe déjà, et affiche le message d'erreur : JPC ERR:File Exists.

La taille occupée par un fichier d'adresses peut être calculé par la formule :

30,5 octets + somme des tailles des fiches

La taille d'une fiche peut être calculée par la formule :

10 octets + nombre de caractères dans la fiche

## Références

Programme AGENDA pour HP-75 par Pierre David.

## Mots-clés associés

ADDELETE, ADFIND, ADGET, ADPUT, ADSIZE, KA

## Auteur

Pierre David

# ADDELETE

Le mot-clef programmable ADDELETE efface une fiche dans un fichier d'adresses.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

ADDELETE *fichier* , *numéro*  
ADDELETE *fichier* , *numéro* , *mot de passe*

## Exemples

ADDELETE ESSAI, 5

Efface la fiche numéro 5 dans le fichier d'adresses ESSAI, sans mot de passe.

ADDELETE A\$, I+1, P\$

Efface la fiche numéro I+1 du fichier d'adresse identifié par la variable A\$ et possédant le mot de passe contenu dans la variable P\$.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Le fichier doit être en mémoire vive.
numéro	Expression numérique arrondie à un entier.	Doit être compris entre 1 et le nombre de fiches.
mot de passe	Expression alphanumérique. Défaut : Pas de mot de passe.	Seuls les 8 premiers caractères sont pris en compte.

## Opération

Le mot-clef ADDELETE supprime dans le fichier d'adresse la fiche dont le numéro est fourni.

ADDELETE refuse d'effacer la fiche si :

- le fichier n'est pas en mémoire vive,
- le fichier n'est pas du type ADRS,
- le fichier contient un mot de passe et le mot de passe fourni n'est pas le bon,
- le numéro de fiche n'est pas valide.

Se reporter au mot-clef ADCREATE pour plus d'explications sur les fichiers d'adresses.

## Références

Programme AGENDA pour HP-75 par Pierre David.

## Mots-clés associés

ADCREATE, ADFIND, ADGET, ADPUT, ADSIZE, KA

## Auteur

Pierre David

La fonction ADFIND cherche un nom dans un fichier d'adresses.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

ADFIND ( *fichier* , *chaîne* )  
 ADFIND ( *fichier* , *chaîne* , *mot de passe* )

## Exemples

A=ADFIND(ESSAI, "Dupond", P\$)

Renvoie le numéro de la fiche correspondant au nom « Dupond » dans le fichier ESSAI avec le mot de passe P\$.

ADDELETE A\$,ADFIND(A\$, "Dup. ")

Efface la fiche correspondant au premier nom commençant par « Dup » dans le fichier d'adresses A\$.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique.	Le fichier doit être en mémoire vive.
chaîne	Expression alphanumérique.	Aucune.
mot de passe	Expression alphanumérique. Défaut : Pas de mot de passe.	Seuls les 8 premiers caractères sont pris en compte.

## Opération

La fonction ADFIND renvoie le numéro de la fiche correspondant au nom fourni avec le paramètre *chaîne*.

Ce numéro de fiche peut ensuite être utilisé avec les fonctions ADGET et ADDELETE pour récupérer une fiche dans un tableau ou pour l'effacer du fichier.

La chaîne de recherche répond aux mêmes règles que pour le mode *recherche* de KA :

- Nom seul : recherche sur le nom seulement. Le premier nom trouvé est renvoyé, peu importe le prénom.
- Nom et prénom (séparés par le caractère /) : recherche sur le nom *et* le prénom.
- Chaîne se terminant par un point : la recherche est abrégée. Les noms n'ont pas besoin d'être introduits en entier. La fiche renvoyée est la première après la chaîne trouvée. Il n'y a pas d'erreur si la chaîne abrégée n'est pas exactement trouvée.

La recherche ne tient pas compte de la différence entre caractères majuscules et minuscules.

Voir KA pour plus d'explications sur la recherche.

Si ADFIND échoue pour une raison ou pour une autre, un nombre négatif est renvoyé. La valeur absolue de ce nombre est le numéro de l'erreur qui a causé l'échec.

Se reporter au mot-clef ADCREATE pour plus d'explications sur les fichiers d'adresses.

## Références

Programme AGENDA pour HP-75 par Pierre David.

## Mots-clés associés

ADCREATE, ADDELETE, ADGET, ADPUT, ADSIZE, KA

## Auteur

Pierre David

# ADGET

Le mot-clef programmable ADGET lit une fiche et la stocke dans un tableau alphanumérique.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

ADGET <i>fichier</i> , <i>tableau</i> , <i>numéro</i> ADGET <i>fichier</i> , <i>tableau</i> , <i>numéro</i> , <i>mot de passe</i>
--

## Exemples

ADGET ESSAI, T\$, 5

Lit la fiche numéro 5 du fichier ESSAI et la stocke dans le tableau alphanumérique T\$.

ADGET A\$, T\$, I+1, P\$

Lit la fiche de numéro I+1 du fichier d'adresse identifié par la variable A\$ et possédant le mot de passe contenu dans la variable P\$.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Le fichier doit être en mémoire vive.
tableau	Nom de tableau alphanumérique.	Doit comporter exactement 8 éléments.
numéro	Expression numérique arrondie à un entier.	Doit être compris entre 1 et le nombre de fiches.
mot de passe	Expression alphanumérique. Défaut : Pas de mot de passe.	Seuls les 8 premiers caractères sont pris en compte.

## Opération

Le mot-clef ADGET lit la fiche spécifiée dans un fichier d'adresses et la stocke dans un tableau alphanumérique, de manière à être exploitée par un programme utilisateur.

Attention : le tableau doit contenir exactement 8 éléments. Chaque élément doit être de longueur suffisante pour mémoriser une ligne complète. La taille limite ne peut excéder 91 caractères.

Exemple de programme exploitant le mot-clef ADGET pour réaliser l'impression des adresses du fichier ADRS :

```
100 F$="ADRS"           ! nom du fichier
110 OPTION BASE 1
120 DIM T$(8)[91]
130 FOR I=1 TO ADSIZE(F$)
140   ADGET F$, T$, I
150   PRINT T$(1)       ! le nom
160   PRINT T$(3)       ! adresse 1
170   PRINT T$(4)       ! adresse 2
180   PRINT T$(5)       ! adresse 3
190   PRINT T$(6)       ! adresse 4
200 NEXT I
```

ADGET refuse de lire la fiche si :

- le fichier n'est pas en mémoire vive,
- le fichier n'est pas du type ADRS,
- le fichier contient un mot de passe et le mot de passe fourni n'est pas le bon,
- le numéro de fiche n'est pas valide.
- le tableau ne contient pas assez d'éléments,
- un des champs de la fiche ne rentre pas dans un des éléments du tableau.

Se reporter au mot-clef ADCREATE pour plus d'explications sur les fichiers d'adresses.

## Références

Programme AGENDA pour HP-75 par Pierre David.

## Mots-clés associés

ADCREATE, ADDELETE, ADFIND, ADPUT, ADSIZE, KA

## Auteur

Pierre David

# ADPUT

Le mot-clef programmable ADPUT écrit une fiche dans un fichier d'adresses.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

ADPUT <i>fichier</i> , <i>tableau</i>
ADPUT <i>fichier</i> , <i>tableau</i> , <i>mot de passe</i>

## Exemples

ADPUT ESSAI, T\$

Ecrit la fiche mémorisée dans le tableau T\$ dans le fichier ESSAI.

ADPUT A\$, T\$, P\$

Ecrit la fiche T\$ dans le fichier d'adresse identifié par la variable A\$ et possédant le mot de passe contenu dans la variable P\$.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Le fichier doit être en mémoire vive.
tableau	Nom de tableau alphanumérique.	Doit comporter exactement 8 éléments.
mot de passe	Expression alphanumérique. Défaut : Pas de mot de passe.	Seuls les 8 premiers caractères sont pris en compte.

## Opération

Le mot-clef ADPUT écrit une fiche dans le fichier d'adresses identifié par le paramètre *fichier*.

La fiche est placée automatiquement dans l'ordre alphabétique.

Attention : le tableau doit contenir exactement 8 éléments, et chaque ligne ne peut contenir plus de 91 caractères.

ADPUT refuse de stocker la fiche si :

- le fichier n'est pas en mémoire vive,
- le fichier n'est pas du type ADRS,
- le fichier contient un mot de passe et le mot de passe fourni n'est pas le bon,
- le tableau ne contient pas assez ou trop d'éléments,
- il n'y a pas assez de mémoire,

Se reporter au mot-clef ADCREATE pour plus d'explications sur les fichiers d'adresses.

## Références

Programme AGENDA pour HP-75 par Pierre David.

## Mots-clés associés

ADCREATE, ADDELETE, ADFIND, ADGET, ADSIZE, KA

## Auteur

Pierre David

La fonction ADSIZE renvoie le nombre de fiches dans un fichier d'adresses.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

ADSIZE ( <i>fichier</i> )
ADSIZE ( <i>fichier</i> , <i>mot de passe</i> )

## Exemples

N=ADSIZE("ESSAI")

Stocke dans la variable N le nombre de fiches du fichier d'adresses ESSAI sans mot de passe.

FOR I=1 TO ADSIZE(A\$,P\$)

Fait une boucle sur toutes les fiches du fichier A\$, connaissant le mot de passe contenu dans la variable P\$.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique.	Le fichier doit être en mémoire vive.
mot de passe	Expression alphanumérique. Défaut : Pas de mot de passe.	Seuls les 8 premiers caractères sont pris en compte.

## Opération

La fonction ADSIZE renvoie le nombre de fiches présentes dans le fichier d'adresses spécifié.

Si ADSIZE échoue pour une raison ou pour une autre, un nombre négatif est renvoyé. La valeur absolue de ce nombre est le numéro de l'erreur qui a causé l'échec.

ADSIZE refuse de retourner le nombre de fiches si :

- le fichier n'est pas en mémoire vive,
- le fichier n'est pas du type ADRS,
- le fichier contient un mot de passe et le mot de passe fourni n'est pas le bon,

Se reporter au mot-clef ADCREATE pour plus d'explications sur les fichiers d'adresses.

## Références

Programme AGENDA pour HP-75 par Pierre David.

## Mots-clés associés

ADCREATE, ADDELETE, ADFIND, ADGET, ADPUT, KA

## Auteur

Pierre David

# ARR

ARR (Arrangements) renvoie le nombre d'arrangements possibles de p éléments pris dans un ensemble de n éléments.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input checked="" type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

ARR ( n , p )

## Exemple

A=ARR(4,3)

Met 24 dans la variable A.

## Paramètres d'entrée

Élément	Description	Restriction
n	Expression numérique.	Nombre entier compris entre 0 et $10^{12}-1$ .
p	Expression numérique.	Nombre entier compris entre 0 et n.

## Opération

$$ARR(n,p) = A_n^p = n! / (n-p)!$$

ARR (n, p) renvoie le nombre d'arrangements de p éléments dans un ensemble de n éléments. Cette fonction est très utilisée dans le domaine des probabilités.

Pour des raisons de précision, ARR n'utilise pas la fonction factorielle, mais un algorithme multiplicatif qui permet d'étendre le domaine de validité de la fonction. Ceci explique aussi un temps d'exécution un peu long pour de grandes valeurs.

## Références

JPC 25 (page 50) première version par Laurent Istria.

JPC 41 (page 32) deuxième version par Guy Toublanc.

## Mots-clés associés

COMB, FACT

## Auteurs

Laurent Istria, Guy Toublanc

ASC\$ (ASCII string) renvoie la chaîne débarrassée des codes ASCII non affichables.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

ASC\$ ( chaîne )
------------------

## Exemple

```
DISP ASC$("AbC"&CHR$(27))
```

Affiche la chaîne "AbC.". Le point remplace le caractère de code 27 (ESC) non affichable.

## Paramètre d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.

## Opération

### L'alphabet ASCII :

L'alphabet ASCII (American Standard Code for Information Interchange) est un système de codage des lettres largement répandu en informatique.

Dans cette norme, les caractères affichables sont les caractères dont le code est compris entre 32 et 126. Les codes de 0 à 31, ainsi que 127 servent à la transmission des informations, et ne sont pas des caractères affichables. Enfin, les caractères dont le code est supérieur à 128 ne sont pas définis.

### La fonction ASC\$ :

ASC\$ renvoie la chaîne fournie en paramètre en remplaçant les caractères non affichables par des caractères "." (point).

## Références

JPC 22 (page 31) première version par Michel Martinet.

JPC 27 (page 34) deuxième version par Michel Martinet.

## Mots-clés associés

ATH\$, HTA\$

## Auteurs

Pierre David, Michel Martinet

# ATH\$

ATH\$ (Ascii To Hexadecimal) renvoie la chaîne hexadécimale correspondant à la chaîne fournie en paramètre.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

ATH\$ ( chaîne )
ATH\$ ( chaîne , mode )

## Exemples

A\$=ATH\$ ("ABCDE")

Place la représentation hexadécimale interne "1424344454" dans la variable A\$.

A\$=ATH\$ ("ABCDE", 1)

Place la représentation hexadécimale standard "4142434445" dans la variable A\$.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
mode	Expression numérique nulle ou non nulle. Défaut : 0	Aucune.

## Opération

ATH\$ renvoie la chaîne de codes hexadécimaux représentant la chaîne fournie en paramètre.

Cette représentation peut se faire de deux manières différentes, suivant le paramètre *mode* :

Si *mode* = 0, valeur logique *faux* (valeur par défaut), la représentation se fait en inversant les chiffres hexadécimaux deux par deux. Ainsi, le caractère "A" (code hexadécimal 41) sera transformé en "14". Cette représentation est conforme à la représentation interne des données dans le HP-71.

Si *mode* est différent de 0, valeur logique *vrai*, la représentation est standard. Le caractère "A" (code hexadécimal 41) sera bien transformé en "41".

## Références

JPC 22 (page 31) première version par Michel Martinet.

JPC 27 (page 34) deuxième version par Michel Martinet.

A paraître : troisième version par Pierre David.

## Mots-clés associés

HTA\$, ASC\$

## Auteurs

Pierre David, Michel Martinet

ATTN (ATTeNtion) active ou désactive l'action de la touche [ATTN] pour interrompre les programmes.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
ATTN ON ATTN OFF	

## Exemples

```
10 ATTN OFF
20 REPEAT
30   K$=KEYWAIT$
40   DISP K$
50 UNTIL K$="#43"
60 ATTN ON
```

Définit une boucle qui affiche les touches appuyées jusqu'à ce que l'utilisateur appuie sur [ATTN].

```
ATTN OFF @ BEEP INF, INF
```

N'essayez pas cet exemple! Vous ne pourrez l'arrêter que par INIT 1.

## Opération

### La touche [ATTN] :

En temps normal, la touche [ATTN] interrompt l'exécution des programmes. Pendant l'exécution de certaines fonctions du module HP-IL, du module Mathématiques ou de JPC Rom, [ATTN] pressée deux fois interrompt la fonction.

### L'ordre ATTN :

ATTN OFF désactive l'action de la touche [ATTN]. Ceci signifie qu'un appui sur cette touche ne peut plus interrompre les programmes ou les fonctions. Cette touche est alors traitée comme n'importe quelle autre touche du HP-71 (code #43).

Cependant, lors d'une introduction de données ou de commandes, la touche [ATTN] efface la ligne que l'on ait exécuté ATTN OFF ou non. ATTN OFF ne fait que désactiver l'interruption des programmes par la touche [ATTN].

Attention : le seul moyen d'interrompre un programme est la séquence d'initialisation de premier niveau INIT 1, qui restaure l'environnement et les variables du programme principal.

ATTN OFF désactive l'action de [ATTN], mais ceci n'a pas d'effet lors d'un INPUT ou LINPUT. Pour masquer l'effet de cette touche, il faut lui assigner une définition nulle. Ceci se fait, par exemple, avec les lignes suivantes :

```
10 DEF KEY "#43", "" ;
20 INPUT A$
```

et en mettant le HP-71 en mode USER. Ainsi, la touche [ATTN] est désactivée lorsque le INPUT est actif.

Une alternative à cette définition de touche est l'utilisation de l'ordre FINPUT.

ATTN ON réactive l'action de la touche [ATTN].

## Références

JPC 23 (page 38) par Pierre David et Michel Martinet.

## Mots-clés associés

DEF KEY, FINPUT, KEY\$, USER

## Auteurs

Pierre David, Michel Martinet

# BELL

BELL (sonnerie) fait sonner l'imprimante si elle en a la capacité.

- Ordre
- Fonction
- Opérateur

- Exécution au clavier
- Mode CALC
- IF...THEN...ELSE
- Opération d'unité

BELL

## Exemple

```
IF DEVADDR("HP82905B")>0 THEN BELL
```

Si l'imprimante est une HP82905B, alors la faire sonner.

## Opération

BELL fait sonner le périphérique désigné par le dernier ordre `PRINTER IS`, si celui-ci en a la capacité.

La ThinkJet ne peut émettre de sonnerie.

### Codes envoyés à l'imprimante :

Caractère de code 7

## Références

*JPC 26* (page 39) première version par Pierre David.

*JPC 40* (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

## Mots-clés associés

BOLD, MODE, PAGELEN, PCR, PFF, PLF, UNDERLINE, WRAP

## Auteur

Pierre David

BOLD active ou désactive le mode « caractères gras » sur l'imprimante.

- Ordre
- Fonction
- Opérateur

- Exécution au clavier
- Mode CALC
- IF...THEN...ELSE
- Opération d'unité

BOLD ON  
BOLD OFF

## Exemples

BOLD ON @ PRINT "JPC"

Active le mode «caractères gras», et imprime la chaîne "JPC".

BOLD OFF @ PRINT "JPC"

Désactive le mode «caractères gras», et imprime la chaîne "JPC".

## Opération

BOLD ON active le passage en caractères gras sur le périphérique désigné par PRINTER IS. BOLD OFF repasse en caractères normaux. L'effet de cet ordre dépend du périphérique utilisé. Il est destiné aux imprimantes reconnaissant le langage de contrôle PCL de Hewlett-Packard.

### Codes envoyés à l'imprimante :

BOLD ON : ESC ( s 1 B

BOLD OFF : ESC ( s 0 B

## Références

JPC 26 (page 39) première version par Pierre David.

JPC 40 (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

## Mots-clés associés

BELL, MODE, PAGELEN, PCR, PFF, PLF, PRINT, PRINTER IS, UNDERLINE, WRAP

## Auteur

Pierre David

# BOX

BOX trace un rectangle.

- Ordre
- Fonction
- Opérateur

- Exécution au clavier
- Mode CALC
- IF...THEN...ELSE
- Opération d'unité

BOX  $x1, y1, x2, y2$

## Exemples

BOX 10, 20, 50, 70

trace un rectangle entre les points (10,20) et (50,70).

BOX X, Y, X+L, Y+H

trace un rectangle à partir du point (X,Y) de largeur L et de hauteur H.

## Paramètre d'entrée

Élément	Description	Restriction
$x1, y1, x2, y2$	Expression numériques arrondies à des entiers.	-32767 à +32767

## Opération

BOX trace un rectangle spécifié par les deux points  $(x1,y1)$  et  $(x2,y2)$ .

Le rectangle est tracé suivant le dernier ordre LINETYPE exécuté.

## Mots-clés associés

FRAME, LINETYPE

## Auteur

Pierre David

CASE est un élément de la structure SELECT ... CASE ... END SELECT.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

CASE *élément* , ...  
CASE *opérateur relationnel* *élément* , ...  
CASE *élément* TO *élément* , ...  
CASE ELSE

## Exemples

CASE 8,5 TO 7,<0,>=10

Sélectionne ce cas si l'expression du SELECT est égale à 8 ou si elle est comprise entre 5 et 7 ou si elle est négative ou si elle est supérieure à 10.

CASE >"Z", "A" TO "BCD", "0" TO "9"

Sélectionne ce cas si l'expression du SELECT est supérieure à "Z" ou si elle est comprise entre "A" et "BCD" ou entre "0" et "9".

## Paramètres d'entrée

Élément	Description	Restriction
élément	Expression numérique ou alphanumérique.	Toutes les expressions doivent être du même type.
opérateur relationnel	<, =, >, <=, >=, <>, # et ?	Aucune.

## Opération

CASE est un des constituants de la structure de choix SELECT ... END SELECT.

CASE offre un choix d'expressions. Si l'une d'entre elles est comparée positivement à l'expression du SELECT, l'exécution continue à l'ordre suivant le CASE sélectionné.

## Références

JPC 52 (page 10) première version par Pierre David et Janick Taillandier.  
Basic 5.0 pour HP 9000 séries 200/300.

## Mots-clés associés

SELECT ... END SELECT

## Auteurs

Pierre David, Janick Taillandier

# CENTER\$

CENTER\$ centre la chaîne fournie en argument en ajoutant des espaces en tête.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

CENTER\$ ( chaîne , largeur )

## Exemple

A\$=CENTER\$("Chaîne à centrer", 22)

Place, dans la variable A\$, 3 espaces suivis de la chaîne fournie en paramètre.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
largeur	Expression numérique arrondie à un entier.	1 à 524287.

## Opération

CENTER\$ ajoute des espaces en tête de la chaîne fournie en paramètre, de façon à ce que cette chaîne soit au centre de *largeur* caractères.

La chaîne est préalablement débarrassée de tous ses espaces superflus (voir REDUCE\$).

## Références

JPC 21 (page 34) première version du programme de formatage en Basic par Pierre David.

JPC 26 (page 50) deuxième version du programme en Basic avec les fonctions en assembleur par Pierre David et Michel Martinet.

## Mots-clés associés

CESURE, FORMAT\$, RED\$, REDUCE\$, SPACES\$

## Auteurs

Pierre David, Michel Martinet

CESURE renvoie le premier emplacement où on peut couper la chaîne fournie en paramètre.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

CESURE ( chaîne , largeur )

## Exemple

A=CESURE("PPC Paris",7)

Place 3 dans la variable A : la césure peut intervenir au troisième caractère.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
largeur	Expression numérique arrondie à un entier.	1 à 1048575.

## Opération

CESURE explore la chaîne à partir du caractère spécifié par l'argument *largeur* en revenant vers le début jusqu'à trouver un endroit où opérer une césure.

CESURE reconnaît les symboles de ponctuation courants : caractères *point d'interrogation*, *point d'exclamation*, *point virgule*, *deux points*, *point* et *parenthèse ouvrante*. L'algorithme est conçu pour ne pas couper la chaîne avant un de ces signes.

## Références

JPC 21 (page 34) première version du programme de formattage en Basic par Pierre David.

JPC 26 (page 50) deuxième version du programme en Basic avec les fonctions en assembleur par Pierre David et Michel Martinet.

## Mots-clés associés

CENTER\$, FORMAT\$, RED\$, REDUCE\$, SPACE\$

## Auteurs

Pierre David, Michel Martinet

# COMB

COMB (combinaisons) renvoie le nombre de combinaisons d'un ensemble de p éléments parmi n.

- |                                    |                        |
|------------------------------------|------------------------|
| <input type="checkbox"/> Ordre     | ■ Exécution au clavier |
| ■ Fonction                         | ■ Mode CALC            |
| <input type="checkbox"/> Opérateur | ■ IF...THEN...ELSE     |
|                                    | ■ Opération d'unité    |

COMB ( n , p )

## Exemple

A=COMB(4,3)

Place 4 dans la variable A.

## Paramètres d'entrée

Élément	Description	Restriction
n	Expression numérique.	Nombre entier compris entre 0 et $10^{12}-1$ .
p	Expression numérique.	Nombre entier compris entre 0 et n.

## Opération

$$\text{COMB}(n,p) = C_n^p = n! / (p! * (n-p)!)$$

COMB calcule le nombre de combinaisons de n éléments pris p à p.

Pour des raisons de précision, COMB n'utilise pas la fonction factorielle, mais un algorithme multiplicatif qui permet d'étendre le domaine de validité de la fonction. Ceci explique aussi un temps d'exécution un peu long pour de grandes valeurs.

## Références

JPC 25 (page 50) première version par Laurent Istria.

JPC 41 (page 32) deuxième version par Guy Toublanc.

## Mots-clés associés

ARR, FACT

## Auteurs

Laurent Istria, Guy Toublanc

CONTRAST retourne la valeur du contraste courant.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

CONTRAST

## Opération

CONTRAST sans paramètre retourne la valeur du contraste courant. Cette valeur est modifiable par l'ordre CONTRAST suivi d'un paramètre entre 0 et 15.

## Références

*JPC 22* (page 42) première version par Laurent Istria.

*JPC 24* (page 41) deuxième version par Jean-Jacques Moreau.

*Manuel du module Forth / Assembler* (page 52). Un exemple de primitive Forth renvoyant le contraste courant.

## Mots-clés associés

CONTRAST

## Auteurs

Laurent Istria, Jean-Jacques Moreau

# CSIZE

CSIZE (Character SIZE) spécifie la taille et l'aspect des caractères tracés par LABEL.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

CSIZE hauteur [, rapport ]

## Exemples

CSIZE 20,0.6

sélectionne une hauteur de caractères de 20 unités graphiques et une largeur de  $20 \times 0.6 = 12$  unités graphiques.

CSIZE 10\*I

sélectionne une hauteur de  $10 \times I$  unités graphiques, et une largeur de  $6 \times I$  unités graphiques.

CSIZE 20,1.2

les caractères seront tracés avec une hauteur de 20 unités graphiques et une largeur de  $20 \times 1.2 = 24$  unités graphiques.

## Paramètres d'entrée

Élément	Description	Restriction
hauteur	Expression numérique arrondie à un entier.	-32767 à +32767
rapport	Expression numérique. Défaut : 0.6	Aucune.

## Opération

Les caractères imprimés par l'instruction LABEL sont définis par plusieurs paramètres déterminant leur forme et leur taille réelles. De plus, chaque caractère se compose d'un symbole et de l'espace qui l'entoure, le séparant des autres caractères.

Le paramètre *hauteur* est la somme, en unités graphiques, de la hauteur du caractère et de l'interligne d'impression.

Le paramètre *rapport* est le rapport entre la largeur et la hauteur de l'espace-caractère. Une valeur faible indique un caractère étroit et une valeur élevée indique un caractère large.

L'exécution de GINIT donne par défaut les valeurs suivantes : *hauteur* = 20 unités graphiques, *rapport d'aspect* = 0,6. Ces paramètres définissent une *largeur* de 12 unités graphiques.

## Mots-clés associés

LABEL

## Auteur

Pierre David

DATEADD (DATE ADDition) renvoie la date correspondant à la date donnée plus un nombre de jours spécifié.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

DATEADD ( *date* , *nombre de jours* )

## Exemples

A=DATEADD (1.011986, 364)

Place 31.121986 (31 décembre 1986) dans la variable A, si le mode DMY est actif.

DATEADD (DATE\$, -1)

Renvoie la date correspondant à hier.

## Paramètres d'entrée

Élément	Description	Restriction
date	Expression numérique interprétée suivant le format courant, ou expression alphanumérique.	doit être comprise entre le 15 octobre 1582 et le 31 décembre 9999.
nombre de jours	Expression numérique arrondie à un entier.	négatif ou positif.

## Opération

DATEADD renvoie la date correspondant à la date donnée plus un certain nombre de jours.

Pour une description complète des formats de date, voir la fonction DATESTR\$.

## Références

*JPC 28* (page 40) première version par Laurent Istria.

*JPC 31* (page 35) deuxième version par François Le Grand.

*JPC 49* (Page 24) troisième version par Pierre David et Janick Taillandier.

*Manuel du module Horloge du HP-41.*

L'ancienne appellation de DATEADD était DATE+.

## Mots-clés associés

DATE\$, DATESTR\$, DDAYS, DMY, MDY

## Auteurs

Pierre David, Laurent Istria, François Le Grand, Janick Taillandier

# DATESTR\$

DATESTR\$ (DATE to STRing) convertit une date au format alphanumérique "aaaa/mm/jj".

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

DATESTR\$ ( date )

## Exemple

A\$=DATESTR\$ (14.071789)

Place "1789/07/14" dans la variable A\$.

## Paramètre d'entrée

Élément	Description	Restriction
date	Expression numérique interprétée suivant le format courant, ou expression alphanumérique.	doit être comprise entre le 15 octobre 1582 et le 31 décembre 9999.

## Opération

### Les formats des dates :

Le HP-71 «de base» dispose de deux formats de dates :

*Le format alphanumérique*

Les dates exprimées dans ce format sont des chaînes alphanumériques composées de deux ou quatre chiffres pour l'année, de deux chiffres pour le mois et de deux chiffres pour le jour. On peut les symboliser par : "aaa/mm/jj" ou "aa/mm/jj".

Par exemple, "1987/05/15" ou "87/05/15" sont des dates valides (15 mai 1987).

Si l'année est sur deux chiffres, elle correspondra à 19aa si aa>=60, à 20aa ou aa<60.

Les fonctions de date de JPC Rom utilisent ces deux formats alphanumériques.

*Le format numérique de base*

Les dates sont exprimées sous la forme d'un nombre du type : *aa**jjj*, où *aa* est l'année et *jjj* le numéro du jour dans l'année.

Par exemple, le 15 mai 1987 est représenté par le nombre 87135. L'année est 1987, et le 15 mai est le 135<sup>ème</sup> jour de l'année.

Ce format est malcommode d'emploi. Il est principalement utilisé, sur un HP-71 «de base» pour faire des calculs de dates.

Avec JPC Rom et ses fonctions de dates, ce format devient inutile. Il a été remplacé par un autre format numérique, identique à celui utilisé par le module horloge du HP-41.

*Le format numérique de JPC Rom*

C'est un format qui permet d'introduire les dates à l'européenne où à l'américaine. Le choix entre ces deux options est fait en utilisant les mots clefs DMY et MDY.

En format DMY (Day Month Year, ou Jour Mois Année), les dates sont introduites suivant le modèle *jj.mmaaaa*. Ainsi, notre 15 mai 1987 sera représenté de la façon suivante :

15.051987

En format américain, ou MDY (Month Day Year), les dates sont introduites suivant le modèle *mm.jjaaaa*. Le 15 mai 1987 sera donc le nombre :

5.151987

Le choix entre ces deux modes est représenté par un l'indicateur binaire -53, valant 0 pour le mode MDY (mode par défaut) et 1 pour le mode DMY. Vous pouvez changer de mode avec les mots clefs DMY ou MDY.

### Les formats utilisés :

Les fonctions de dates de JPC Rom admettent donc deux types de dates :

- les dates au format alphanumérique ("aaa/mm/jj" ou "aa/mm/jj"), ou

# DATESTR\$ (suite)

- les dates au format numérique (*jj.mmaaaa* en mode DMY, ou *mm.jjaaaa* en mode MDY).

## La fonction DATESTR\$ :

La fonction DATESTR\$ est une fonction de conversion d'une date au format numérique JPC Rom (*jj.mmaaaa* ou *mm.jjaaaa*) en une date au format alphanumérique ("*aaaa/mm/jj*").

Vous pouvez donc l'utiliser en complément des fonctions de calculs de dates pour isoler plus facilement les composantes de la date, par exemple.

## Références

*JPC 49* (Page 24) troisième version de DATELEX comportant DATESTR\$ par Pierre David et Janick Taillandier.

## Mots-clés associés

DATE\$, DMY, MDY, SETDATE

## Auteurs

Pierre David, Janick Taillandier

# DBLIST

DBLIST (Display Basic LIST) liste un programme Basic sous forme structurée.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

```
DBLIST [ INDENT indentation ][ TO destination ]
DBLIST fichier [ INDENT indentation ][ TO destination ]
DBLIST fichier , ligne départ [ INDENT indentation ][ TO destination ]
DBLIST fichier , ligne départ , ligne fin [ INDENT indentation ][ TO destination ]
```

## Exemples

DBLIST MONSUB INDENT 3

Liste le programme MONSUB, de la première à la dernière ligne, en indentant les structures par trois espaces.

DBLIST MONSUB,10

Liste la ligne 10 du programme MONSUB, sans indentation des structures.

DBLIST MONSUB,10,100 INDENT 2 TO LISTE

Liste le programme MONSUB, de la ligne 10 à la ligne 100, en indentant les structures de deux espaces et en envoyant le résultat dans le fichier LISTE.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets. Défaut : fichier courant.	Nom de fichier avec spécificateur d'unité en option.
ligne départ	Constante entière identifiant une ligne de programme. Défaut : première ligne du programme.	1 à 9999.
ligne fin	Constante entière identifiant une ligne de programme. Défaut : ligne de départ si spécifiée; sinon, dernière ligne du programme.	Ligne de départ jusqu'à 9999.
indentation	Expression numérique arrondie à un entier. Défaut : 0	0 à 255.
destination	Expression alphanumérique ou chaîne sans guillemets. Défaut : Le listage va sur le périphérique désigné par DISPLAY IS.	Nom de fichier avec spécificateur d'unité en option.

## Opération

L'exécution de DBLIST provoque un listage du fichier BASIC sous forme « structurée » sur l'appareil désigné par le dernier ordre DISPLAY IS ou sur l'écran interne si aucun appareil n'a été spécifié.

DBLIST liste le programme Basic en cadrant les numéros de ligne à 4 caractères (en complétant par des espaces s'ils font moins de 4 caractères). Ainsi, toutes les lignes de votre programme seront bien alignées, quel que soit leur numéro de ligne.

DBLIST ne liste pas les numéros des lignes ne contenant qu'une remarque (avec le symbole !, mais pas avec REM). Il met un tiret pour marquer la première remarque d'une série. L'ordre RENUMREM permet de faciliter la numérotation des remarques. Avec ce traitement, les lignes de remarques n'apparaissent plus comme n'importe quelle autre ligne Basic.

DBLIST saute une ligne avant toute définition de fonction (DEF FN), tout bloc de DATA ou tout label. De même, une ligne est sautée après les définitions de fonctions et les blocs de DATA. Les blocs constituant votre programme sont ainsi mieux séparés.

DBLIST saute une ligne, trace un trait et saute une nouvelle ligne avant tout nouveau sous-programme (SUB). Ceci permet de mieux scinder le programme en parties indépendantes.

Enfin, DBLIST permet l'indentation des structures. Le corps des structures, que ce soit avec des boucles standard (FOR...NEXT) ou de JPC Rom (WHILE...END WHILE ou les tests ou SELECT), est décalé vers la droite du nombre d'espaces *indentation* fourni en paramètre par INDENT. Par défaut, cette valeur est nulle, c'est à dire que les structures ne sont pas indentées.

L'indentation des structures fait très vite apparaître tous les défauts de structuration de vos programmes. C'est un complément indispensable des ordres de programmation structurée de JPC Rom.

L'adjonction de `TO fichier` permet de récupérer le listage non plus sur l'affichage mais dans un fichier. A titre indicatif, c'est ainsi que sont listés les programmes en Basic dans *JPC*. Le fichier est créé, puis rempli. Si le fichier existait déjà, une erreur `JPC ERR:File Exists` est générée.

La valeur courante de `DELAY` détermine le délai d'affichage de chaque ligne. La valeur de `WIDTH`, quant à elle, détermine la largeur de la ligne imprimée.

Pour arrêter un listage et afficher le curseur, il suffit d'appuyer sur la touche `[ATTN]`.

## Références

*JPC 18* (page 25) première version du programme `JPCLISTE` en Basic par Pierre David et Michel Martinet.

*JPC 38* (page 24) première version de `BLIST` par Jean-Pierre Bondu.

*JPC 53* (page 16) deuxième version de `DBLIST` par Pierre David et Janick Taillandier.

L'ancienne appellation de `DBLIST` était `BLIST`.

## Mots-clés associés

`DELAY`, `LIST`, `PBLIST`, `PLIST`, `WIDTH`, tous les mots-clefs de programmation structurée

## Auteurs

Jean-Pierre Bondu, Pierre David, Janick Taillandier

# DDAYS

DDAYS (Delta DAYS) calcule le nombre de jours entre deux dates.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

DDAYS ( *date1* , *date2* )

## Exemples

A=DDAYS(1.011986, 31.121986)

Stocke 364 jours dans la variable A.

DISP DDAYS( DATE\$, 14.071789)

Calcule et affiche le nombre de jours nous séparant du 14 juillet 1789.

## Paramètre d'entrée

Élément	Description	Restriction
date1, date2	Expressions numériques interprétées suivant le format courant, ou expressions alphanumériques.	doivent être comprises entre le 15 octobre 1582 et le 31 décembre 9999.

## Opération

DDAYS calcule le nombre de jours entre *date1* et *date2*. Si *date1* est postérieure à *date2*, le résultat est positif. Pour une description complète des formats d'introduction de dates, voir la fonction DATESTR\$.

## Références

*JPC 28* (page 40) première version par Laurent Istria.

*JPC 49* (page 24) troisième version par Pierre David et Janick Taillandier.

*Manuel du module Horloge du HP-41.*

## Mots-clés associés

DATEADD, DATESTR\$, DMY, MDY

## Auteurs

Pierre David, Laurent Istria, Janick Taillandier

DDIR (Display DIRectory) liste le répertoire du périphérique spécifié.

- |  |  |
|--|--|
| <input checked="" type="checkbox"/> Ordre<br><input type="checkbox"/> Fonction<br><input type="checkbox"/> Opérateur | <input checked="" type="checkbox"/> Exécution au clavier<br><input type="checkbox"/> Mode CALC<br><input checked="" type="checkbox"/> IF...THEN...ELSE<br><input type="checkbox"/> Opération d'unité |
|--|--|

DDIR [ TO destination ]  
 DDIR *spécificateur de fichier* [ TO destination ]  
 DDIR ALL [ TO destination ]

## Exemples

DDIR :TAPE	Liste le répertoire de l'unité de stockage de masse :TAPE .
DDIR :PORT(0) TO LISTE	Liste le contenu du port numéro 0 dans le fichier LISTE.
DDIR ALL	Liste tous les fichiers présents dans le HP-71.
DDIR ESSAI:TAPE(2) TO A\$	Liste tous les fichiers situés après le fichier ESSAI sur l'unité de stockage de masse :TAPE(2) dans le fichier dont le nom est contenu dans la variable A\$.

## Paramètres d'entrée

Élément	Description	Restriction
spécificateur de fichier	Expression alphanumérique ou chaîne sans guillemets. Défaut : :MAIN	Un spécificateur d'unité ou un nom de fichier avec spécificateur d'unité en option.
destination	Expression alphanumérique ou chaîne sans guillemets. Défaut : Le listage est effectué sur l'écran.	Nom de fichier avec spécificateur d'unité en option.

## Opération

Le mot-clef DDIR liste sur le périphérique d'affichage le répertoire de l'unité spécifiée. Le listage est similaire à celui produit par l'ordre standard CAT.

L'affichage se fait en respectant les valeurs spécifiées par l'ordre DELAY. Pour une bonne lecture, nous conseillons un DELAY  $x, 8$  qui évitera le défilement de l'affichage.

### La spécification du fichier

La syntaxe de DDIR vous permet de sélectionner un périphérique ou une partie de répertoire.

#### *Spécificateur de périphérique seul*

Si vous ne fournissez que le nom d'un périphérique, seul le répertoire de ce périphérique sera listé. Par exemple :

- DDIR :PORT(0.01) liste le répertoire du port numéro 0.01,
- DDIR :TAPE liste le répertoire du support de mémoire de masse HP-IL,
- DDIR :PORT liste le contenu de tous les ports du HP-71,
- DDIR :MAIN liste exclusivement le contenu de la mémoire principale.

#### *Spécificateur de fichier et de périphérique*

Si vous spécifiez à la fois le fichier et le périphérique, le listage débute à partir de ce fichier, et continue jusqu'au dernier fichier du périphérique. Par exemple :

- DDIR ESSAI:MAIN liste le répertoire de la mémoire principale après le fichier ESSAI inclus,
- DDIR ESSAI:TAPE fait de même sur l'unité de mémoire de masse,
- DDIR ESSAI:PORT(0.01) fait de même sur le port numéro 0.01,
- DDIR ESSAI:PORT cherche le fichier dans les ports, et liste le répertoire du port trouvé.

# DDIR (suite)

## Cas particuliers

DDIR ALL liste le répertoire de tous les fichiers présents dans le HP-71.

DDIR liste exclusivement le répertoire de la mémoire principale.

DDIR suivi d'un spécificateur de fichier seul (sans spécificateur de périphérique) cherche le fichier dans toute la mémoire, puis liste le reste de l'unité (port ou mémoire principale) trouvée.

## La redirection de la sortie

Lorsque DDIR est suivi de TO, puis d'un spécificateur de fichier, le listage est stocké dans ce fichier. Aucune ligne n'est affichée à l'écran. Utiliser DDIR ou PDIR est strictement équivalent lorsqu'une redirection est demandée.

Si le fichier existe déjà, une erreur ERR:File Exists est générée.

Le format des informations stockées dans le fichier est identique à celui fourni par la fonction CAT\$. Se reporter à cette dernière pour plus de détails.

Cette caractéristique est identique à celle fournie par les ordres PDIR, DBLIST et PBLIST.

## Applications

La redirection est avantageuse, car elle vous permet d'effectuer une action sur tous les fichiers d'un périphérique donné. Par exemple :

```
100 DIM P$[8],F$[43],T$[8],A
110 T$="TMP"           ! fichier temporaire
120 FINPUT P$,"Périphérique: :MAIN","15PU",A
130 IF NOT A THEN END
140 SFLAG -1 @ PURGE T$ @ CFLAG -1
150 DDIR ":%P$ TO T$
160 ASSIGN #1 TO T$
170 LOOP
180  READ #1;F$
190  F$[POS(F$," ")]=""      ! supprime caractères inutiles
   - l'action à effectuer
200  SECURE F$&":"&P$      ! par exemple...
   - fin de l'action
210 END LOOP
```

Cet exemple simple exécute une action (voir ligne 200) pour chaque fichier du périphérique demandé. En remplaçant l'action par une autre, vous pouvez très simplement copier tous les fichiers d'un port dans un autre port, effacer des fichiers, changer des noms, etc.

## Références

*A paraître* : première version par Jean-Jacques Dhénin.

## Mots-clés associés

CAT\$, CAT, DBLIST, PBLIST, PDIR

## Auteur

Jean-Jacques Dhénin

DMY (Day Month Year) valide l'introduction des dates au format numérique *jj.mmaaaa*.

- Ordre
- Fonction
- Opérateur

- Exécution au clavier
- Mode CALC
- IF...THEN...ELSE
- Opération d'unité

DMY

## Opération

Dans le mode sélectionné par DMY, les dates utilisées dans les fonctions de calcul de dates de JPC Rom peuvent être introduites sous la forme *jj.mmaaaa* (format numérique).

Le format alphanumérique, quant à lui, est toujours utilisable.

Par exemple, pour calculer le jour correspondant au 14 juillet 1789, il faut faire en mode DMY :

`DOW$(14.071789)`, ou `DOW$("1789/07/14")`

Pour plus de détails sur les formats d'introduction des dates, consulter la fonction `DATESTR$`.

## Références

*JPC 28* (page 40) première version par Laurent Istria.

*JPC 49* (page 24) troisième version par Pierre David et Janick Taillandier.

*Manuel du module Horloge du HP-41*.

## Mots-clés associés

`DATEADD`, `DATESTR$`, `DDAYS`, `DOW`, `DOW$`, `MDY`

## Auteurs

Pierre David, Laurent Istria, Janick Taillandier

# DOW

DOW (Day Of Week) renvoie le jour de la semaine correspondant à la date spécifiée sous la forme d'un numéro de jour.

<input type="checkbox"/> Ordre	■ Exécution au clavier
■ Fonction	■ Mode CALC
<input type="checkbox"/> Opérateur	■ IF...THEN...ELSE
	■ Opération d'unité

DOW
DOW ( date )

## Exemples

A=DOW(1.011986)

Place dans la variable A le jour de la semaine correspondant au 1<sup>er</sup> janvier 1986.

A=DOW(ATE\$)

Renvoie le numéro du jour de la semaine correspondant à aujourd'hui. Une méthode plus astucieuse consiste à faire simplement A=DOW.

DISP DOW

Affiche le numéro du jour courant.

## Paramètre d'entrée

Élément	Description	Restriction
date	Expression numérique interprétée suivant le format courant, ou expression alphanumérique. Défaut : aujourd'hui	doit être comprise entre le 15 octobre 1582 et le 31 décembre 9999.

## Opération

A la différence de DOW\$, DOW renvoie le jour de la semaine correspondant à une date donnée sous une forme numérique. Vous pouvez ainsi incorporer cette date plus facilement dans vos calculs.

Référez-vous à la fonction DATESTR\$ pour une complète description des formats de dates.

Par exemple :

```
100 SELECT DOW
110 CASE 0
120   A$="Dimanche"
130 CASE 1
140   A$="Lundi"
150 CASE 2
160   A$="Mardi"
170 CASE 3
180   A$="Mercredi"
190 CASE 4
200   A$="Jeudi"
210 CASE 5
220   A$="Vendredi"
230 CASE 6
240   A$="Samedi"
250 END SELECT
260 DISP DATE$;" : ";A$
```

0 correspond à Dimanche, 1 à Lundi... et 6 à Samedi.

## Références

JPC 17 (page 25) programme de calcul du jour de la semaine en Basic par Pierre David.

JPC 28 (page 40) première version par Laurent Istria.

JPC 49 (page 24) troisième version par Pierre David et Janick Taillandier.

*Manuel du module Horloge du HP-41.*

## **Mots-clés associés**

DATE\$, DATESTR\$, DMY, DOW\$, MDY, SETDATE

## **Auteurs**

Pierre David, Laurent Istria, Janick Taillandier

# DOW\$

DOW\$ (Day Of Week) renvoie le nom du jour spécifié, ou de la date courante.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

DOW\$ DOW\$ ( date )
-------------------------

## Exemples

A\$=DOW\$(1.011986)

Place la chaîne wednesday dans la variable A\$.

DISP DOW\$

Affiche le nom du jour courant.

## Paramètre d'entrée

Élément	Description	Restriction
date	Expression numérique interprétée suivant le format courant, ou expression alphanumérique. Défaut : aujourd'hui	doit être comprise entre le 15 octobre 1582 et le 31 décembre 9999.

## Opération

DOW\$ renvoie le jour de la semaine correspondant à la date spécifiée.

Si aucune date n'est fournie, DOW\$ renvoie le jour de la semaine correspondant à la date courante.

Les noms des jours sont en anglais. Leur signification est :

Sunday	Dimanche
Monday	Lundi
Tuesday	Mardi
Wednesday	Mercredi
Thursday	Jeudi
Friday	Vendredi
Saturday	Samedi

Les noms des jours correspondent à des messages contenus dans le lex JPC Rom. Vous pouvez utiliser la fonction MSG\$ (module Forth/Assembler ou Editeur de textes, ou disponible à la User's Library) pour visualiser tous les jours de la semaine :

Dimanche correspond au message 225008+0, Lundi au message 225008+1, et ainsi de suite jusqu'à Samedi au message 225008+6.

Du fait que les noms des jours sont dans une table de messages dans un Lex, ils peuvent être traduits par l'intermédiaire d'un Lex traducteur.

## Références

JPC 17 (page 25) programme de calcul du jour de la semaine en Basic par Pierre David.

JPC 28 (page 40) première version par Laurent Istria.

JPC 49 (page 24) troisième version par Pierre David et Janick Taillandier.

Manuel du module Horloge du HP-41.

## Mots-clés associés

DATE\$, DATESTR\$, DMY, DOW, MDY, SETDATE

**Auteurs**

Pierre David, Laurent Istria, Janick Taillandier

# DRAW

DRAW trace une ligne entre le point courant et le point spécifié.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

DRAW  $x, y$

## Exemples

DRAW 10, 20

trace une ligne droite entre le point courant et le point de coordonnées (10,20).

DRAW  $X+R*\text{COS}(T), Y+R*\text{SIN}(T)$

trace un segment de droite entre le point courant et le point de coordonnées  $(x + r \cos t, y + r \sin t)$ .

## Paramètre d'entrée

Élément	Description	Restriction
x, y	Expressions numériques arrondies à des entiers.	-32767 à +32767

## Opération

DRAW trace un trait entre la position courante de la plume et le point spécifié.

Le trait est tracé suivant le type de ligne défini par le précédent ordre LINETYPE.

Si vous spécifiez un point en dehors de la zone de traçage, la plume trace effectivement le trait de la position courante dans la direction du point spécifié, mais s'arrête lorsqu'elle rencontre la limite de la zone. Si vous effectuez ensuite un traçage vers un point dans la zone de traçage, la plume se déplace sur la limite de la zone au point où le trait couperait la zone, puis trace le reste du trait jusqu'au nouveau point dans les limites. Si les deux extrémités d'un trait se trouvent en dehors des limites, seule la partie du trait située dans les limites est tracée.

## Mots-clés associés

LINETYPE, IDRAW, MOVE

## Auteur

Pierre David

EDIT vous permet de réunir des fichiers LEX, ou d'éditer des fichiers résidant sur un périphérique. EDIT n'est pas programmable.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input type="checkbox"/> IF...THEN...ELSE <input type="checkbox"/> Opération d'unité
--	---

```

EDIT
EDIT fichier1
EDIT fichier1 TO fichier2
  
```

## Exemples

EDIT AREUH:TAPE

Copie le fichier AREUH résidant sur :TAPE vers la mémoire centrale, puis en fait le fichier courant.

EDIT AREUH:TAPE TO TOTO:PORT(0)

Copie le fichier AREUH résidant sur :TAPE vers le port indépendant numéro 0 en changeant son nom en TOTO, puis en fait le fichier courant.

EDIT STRINGLX

Edite le fichier LEX STRINGLX.

## Paramètres d'entrée

Élément	Description	Restriction
fichier1	Expression alphanumérique ou chaîne sans guillemets. Défaut : Fichier de travail système	optionnel.
fichier2	Expression alphanumérique ou chaîne sans guillemets. Défaut : fichier de même nom en mémoire principale.	Le spécificateur de périphérique doit être en mémoire vive.

## Opération

### La recopie et l'édition des fichiers :

Si le premier spécificateur de fichier indique une unité de mémoire de masse extérieure, le fichier est d'abord copié.

Si le deuxième spécificateur est fourni, le fichier est copié dans ce fichier.

Le fichier est ensuite édité, c'est à dire qu'il devient le fichier de travail courant.

Le comportement de EDIT sur des fichiers externes se résume à un ordre COPY suivi d'un EDIT standard sur ce fichier.

Si le type du fichier recopié est invalide, c'est à dire différent de Basic, Keys ou Lex, la copie est effectuée, puis l'erreur ERR:Invalid Filetype est renvoyée.

### Le chaînage de fichiers Lex :

Le fichier édité peut être un fichier Lex. C'est la première étape du chaînage de Lex. Voir MERGE pour plus de détails.

### Attention !

Lorsque vous éditez un fichier LEX, celui-ci devient le fichier de travail courant. Si vous exécutez l'ordre PURGE sur ce fichier, le fichier de travail courant n'est pas remis sur le fichier de travail système (workfile), et cela donne des comportements étranges.

Pour remédier à cela, soyez sûr de faire EDIT pour éditer le fichier de travail système à la fin de votre séance de chaînage.

## Références

JPC 31 (page 54) édition des fichiers sur périphériques par Jean-Pierre Bondu.

JPC 23 (page 47) programme de chaînage de fichiers Lex en Basic par Michel Martinet.

JPC 37 (page 22) chaînage de fichiers Lex en assembleur par Pierre David et Michel Martinet.

# **EDIT (suite)**

## **Mots-clés associés**

COPY, EDIT, MERGE

## **Auteurs**

Jean-Pierre Bondu, Pierre David, Michel Martinet

ENDUP définit une chaîne de commandes à exécuter lorsque le HP-71 se met hors tension.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

ENDUP *chaîne de commande*

## Exemple

ENDUP "BEEP@'Au revoir...'"

Le HP-71 émettra un signal sonore, puis affichera "Au revoir..." lors de chaque extinction.

## Paramètre d'entrée

Élément	Description	Restriction
chaîne de commande	Expression alphanumérique.	0 à 95 caractères.

## Opération

Lorsque vous spécifiez une chaîne de commandes par ENDUP, vous pouvez utiliser toute chaîne d'instructions que vous pouvez, par ailleurs, exécuter au clavier.

Lorsque vous exécutez ENDUP, le HP-71 stocke la chaîne de commandes spécifiée telle qu'elle est entrée, sans vérifier les erreurs de syntaxe. L'ordinateur ne conserve qu'une chaîne ENDUP à un instant donné. Lorsque vous mettez le HP-71 hors tension, la chaîne ENDUP est exécutée si elle est exempte d'erreur. Sinon, l'ordinateur affiche un message d'erreur puis se met dans un état où un appui sur [ATTN] met le HP-71 hors tension.

La chaîne spécifiée est conservée dans un buffer. Voir le mot-clef ADBUF\$ pour plus de renseignements sur les buffers et leur utilisation.

Note : la chaîne spécifiée par ENDUP n'est pas exécutée lorsque le HP-71 se met hors tension en mode CALC ou avec KA.

## Références

JPC 25 (page 43) première version par Jean-Jacques Moreau.

JPC 31 (page 29) deuxième version par Jean-Jacques Moreau.

## Mots-clés associés

ADBUF\$, ENDUP\$, STARTUP\$, STARTUP

## Auteur

Jean-Jacques Moreau

# ENDUP\$

ENDUP\$ renvoie la chaîne de commandes spécifiée par ENDUP.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

ENDUP\$

## Exemple

A\$=ENDUP\$

Place la commande qui doit être exécutée à l'extinction du HP-71 dans la variable A\$.

## Opération

ENDUP\$ renvoie la chaîne de commandes devant être exécutée à l'extinction du HP-71. La longueur de cette chaîne ne peut excéder 95 caractères.

ENDUP\$ renvoie une chaîne nulle si aucune commande n'avait été spécifiée par ENDUP.

## Références

*JPC 25* (page 43) première version par Jean-Jacques Moreau.

*JPC 31* (page 29) deuxième version par Jean-Jacques Moreau.

## Mots-clés associés

ENDUP, STARTUP, STARTUP\$

## Auteur

Jean-Jacques Moreau

ENTRY\$ (point d'entrée) renvoie l'adresse du point d'entrée du mot-clef spécifié.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

ENTRY\$ ( <i>mot-clef</i> )
ENTRY\$ ( <i>mot-clef</i> , <i>numéro d'ordre</i> )

## Exemples

A\$=ENTRY\$ ("ENTRY\$")

Place l'adresse de la routine d'exécution de ENTRY\$ dans la variable A\$.

DISP ENTRY\$ ("EDIT", 2)

Renvoie l'adresse du deuxième EDIT, c'est à dire le standard. Les fichiers Lex sont cherchés avant les fonctions standard.

## Paramètres d'entrée

Élément	Description	Restriction
mot-clef	Expression alphanumérique.	Le mot-clef doit exister.
numéro d'ordre	Expression numérique arrondie à un entier. Défaut : 1	Le mot-clef doit exister.

## Opération

ENTRY\$ renvoie l'adresse du point d'entrée de la fonction spécifiée. L'adresse de la routine d'exécution est synonyme d'adresse du point d'entrée.

La fonction ENTRY\$ est surtout utilisée conjointement avec le débogueur (HP-82178A) pour fournir un point d'arrêt facile à utiliser.

Attention toutefois : les fichiers dans la mémoire principale du HP-71 sont constamment déplacés. Par exemple, toute suppression de fichier, tout ajout ou retrait dans un fichier situé avant le fichier Lex contenant la fonction change l'adresse de son point d'entrée. Vous pouvez éviter ce problème en plaçant vos programmes en cours d'étude dans des ports indépendants.

Le point d'entrée est l'adresse de la routine d'exécution, c'est à dire l'adresse spécifiée par la pseudo-opération ENTRY de l'assembleur du module Forth / Assembleur.

Si le deuxième paramètre est fourni, ENTRY\$ cherche la fonction dans les autres fichiers Lex valides, puis dans les fonctions standard du HP-71.

Si le mot-clef n'existe pas, ou si le deuxième paramètre est supérieur au nombre de répétitions de la fonction dans votre HP-71, ENTRY\$ renvoie l'erreur ERR: Invalid Arg.

Les mots-clefs de plus de 8 caractères nécessitent un traitement spécial par le système. Ainsi, des mots comme UNDERLINE ou RANDOMIZE sont reconnus comme UNDERLIN ou RANDOMIZ. Ce sont ces fonctions qui reconnaissent le E final. ENTRY\$ ne peut connaître ces caractères supplémentaires. Pour les trouver, cette fonction reconnaît UNDERLIN et ignore le E final. Ainsi, ENTRY\$ ("RANDOMIZE") comme ENTRY\$ ("RANDOMIZ - - -") ignorent les caractères supplémentaires et renvoient la même adresse.

Le mot-clef trouvé est le plus long mot-clef comprenant les caractères spécifiés, les autres sont ignorés. Ainsi, ENTRY\$ ("MEMORY") renvoie l'adresse du point d'entrée de la fonction MEM.

## Références

JPC 31 (page 22) première version par Jean-Jacques Moreau.

Manuel du module Forth / Assembleur (page 63).

Internal Design Specification, Volume I. En anglais.

# ENTRY\$ (suite)

## Mots-clés associés

ADDR\$, LEX, PEEK\$, TOKEN

## Auteur

Jean-Jacques Moreau

ESC\$ (ESCape) renvoie la chaîne précédée du caractère « échappement ».

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

ESC\$
ESC\$ ( chaîne )

## Exemples

PRINT ESC\$("Y")

Place un périphérique d'impression tel que la ThinkJet en mode moniteur, c'est à dire qu'il affiche tous les caractères reçus.

PRINT ESC\$("\*b80W")&G\$

Sur la ThinkJet ou la LaserJet, envoie une ligne de graphique.

DISP ESC\$&"j";

Sur l'interface vidéo HP-92198B, passe en jeu de caractères Roman8.

## Paramètre d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique. Défaut : Chaîne vide.	Aucune.

## Opération

### Les séquences d'échappement :

Les séquences d'échappement sont utilisées par bon nombre d'équipements informatiques pour commander des périphériques.

Le HP-71 procède de la même manière pour les périphériques raccordés à la boucle HP-IL, de même que pour son afficheur interne.

Une séquence d'échappement est préfixée par un caractère de code 27, appelé *code escape*, abrégé par *ESC*. Ceci est reconnu par le périphérique comme un début de commande, et non comme une donnée normale.

Ce code *ESC* est suivi d'une chaîne de caractères codant la commande. Si le périphérique la reconnaît, il la traite.

Prenons l'exemple de la ThinkJet. Si vous faites :

```
PRINT "LE HP-71";
```

l'imprimante imprimera «LE HP-71». Si maintenant vous faites :

```
PRINT CHR$(27)&"&d";
```

l'imprimante traitera les 4 caractères reçus comme l'ordre de passer en mode «souligné», et ne les imprimera pas. C'était une séquence d'échappement.

Vous n'avez pas besoin de connaître les séquences d'échappement les plus fréquentes pour la ThinkJet. Vous pouvez aussi utiliser les ordres BOLD, PAGELEN, UNDERLINE et WRAP.

### La fonction ESC\$ :

La fonction ESC\$ ajoute un code *escape* en tête de la chaîne spécifiée. Si aucune chaîne n'est fournie, ESC\$ équivaut à CHR\$(27).

## Références

JPC 26 (page 39) première version par Pierre David.

JPC 40 (page 16) deuxième version par Pierre David.

Consultez aussi les manuels de vos périphériques...

# ESC\$ (suite)

## Mots-clés associés

BOLD, CHR\$, PAGELEN, UNDERLINE, WRAP

## Auteur

Pierre David

EXECUTE exécute la chaîne de commandes spécifiée et arrête le programme.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input type="checkbox"/> IF...THEN...ELSE                |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

EXECUTE *chaîne de commandes*

## Exemple

```
10 EXECUTE "FREEPORT(0)@RUN, 'A' "  
20 'A' :
```

Rend le port interne numéro 0 indépendant, et continue l'exécution au label 'A'.

## Paramètre d'entrée

Élément	Description	Restriction
chaîne de commandes	Expression alphanumérique.	de 0 à 95 caractères.

## Opération

EXECUTE exécute la chaîne fournie en paramètre, puis stoppe l'exécution du programme.

Cela permet de rendre « programmables » des fonctions qui ne le sont pas.

N'essayez pas d'utiliser EXECUTE dans un sous-programme. Cela a pour effet d'annuler tous les environnements appelants. De même, ne l'utilisez pas dans une structure de boucle telle LOOP ... END LOOP ou une structure de choix multiligne (IF ou SELECT).

Le programme est encore considéré comme *en train de s'exécuter* jusqu'à ce que la chaîne complète soit exécutée. Ceci permet d'utiliser CONT pour reprendre l'exécution du programme.

## Références

JPC 31 (page 29) deuxième version de ENDUPLEX par Jean-Jacques Moreau.

## Mots-clés associés

ENDUP, STARTUP

## Auteur

Jean-Jacques Moreau

# EXIT

EXIT sort prématurément d'une boucle FOR ... NEXT.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

EXIT *variable de boucle*

## Exemple

```
10 FOR I=1 TO INF
20 IF FNC(I) THEN EXIT I
30 NEXT I @ BEEP
```

Sort de la boucle FOR ... NEXT et continue l'exécution après le NEXT I et émet un signal sonore si FNC(I) est différent de 0.

## Paramètre d'entrée

Élément	Description	Restriction
variable de boucle	Variable numérique simple.	Aucune.

## Opération

EXIT sort proprement d'une boucle FOR ... NEXT. Les informations nécessaires à la gestion de la boucle sont effacées.

La sortie normale d'une boucle se fait sur l'instruction NEXT lorsque le contenu de la variable de boucle atteint ou dépasse la valeur limite.

Il existe des cas où l'on désire sortir prématurément d'une boucle de ce type. Par exemple, pour traiter les cas d'erreur. L'ordre EXIT fait alors cette sortie proprement. Un exemple d'exécution pourrait être ce petit programme qui calcule 10 racines carrées, à moins que l'un des arguments soit négatif :

```
10 DATA 1, 2, 3, 4, 5, -6, 7, 8, 9, 10
20 FOR I=1 TO 10
30 READ X
40 IF X<0 THEN EXIT I
50 DISP SQR(X)
60 NEXT I
70 DISP "FINI"
```

## Références

JPC 30 (page 49) première version par Janick Taillandier.

## Mots-clés associés

FOR ... NEXT, LEAVE,

## Auteur

Janick Taillandier

FILEPOS (FILE POSition) cherche une chaîne générique dans un fichier Texte.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

```
FILEPOS ( fichier , motif )
FILEPOS ( fichier , motif , début )
FILEPOS ( fichier , motif , début , fin )
FILEPOS ( fichier , motif , début , fin , colonne )
FILEPOS ( canal , motif )
FILEPOS ( canal , motif , début )
FILEPOS ( canal , motif , début , fin )
FILEPOS ( canal , motif , début , fin , colonne )
```

## Exemples

DISP FILEPOS ("ESSAI", "JPC Rom")

Cherche la chaîne "JPC Rom" dans le fichier ESSAI et retourne le numéro de l'enregistrement trouvé, ou -1 si la recherche a échoué.

L=FILEPOS (5, "[0-9]", 25, 50)

Recherche dans le fichier correspondant au canal #5, la première occurrence d'un chiffre en début d'enregistrement, entre les lignes 25 et 50.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique.	Le fichier doit être en mémoire vive.
canal	Expression numérique arrondie à un entier.	1 à 255, et le fichier doit être en mémoire vive.
motif	Expression alphanumérique.	Expression générique valide.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575.
fin	Expression numérique arrondie à un entier. Défaut : dernier enregistrement du fichier.	début à 1048575.
colonne	Expression numérique arrondie à un entier. Défaut : 1	1 à 65535.

## Opération

FILEPOS recherche la première occurrence de *motif* dans le fichier spécifié. Si une occurrence est trouvée, le numéro de l'enregistrement est retourné. Si aucune occurrence n'est trouvée, la valeur -1 est renvoyée. Les paramètres *début* et *fin* permettent de restreindre la recherche dans une partie du fichier.

**Attention** : les paramètres *début* et *fin* de FILEPOS sont exprimés en *numéros d'enregistrement*. L'enregistrement 0 correspond à la ligne 1 du fichier, de manière à être compatible avec les ordres standard du HP-71 (READ #, RESTORE #, etc.).

La recherche tient compte des caractères génériques étendus (voir XEDIT pour plus de précision sur les caractères génériques). Le tableau ci-dessous résume les caractères spéciaux :

Caractère	Signification
-----	-----
\	annule la signification du caractère suivant
^	début de ligne
\$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^ ]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

# FILEPOS (suite)

Le paramètre *colonne* spécifie le numéro de la colonne à partir de laquelle la recherche commence. La recherche commencera à la ligne et à la colonne spécifiées, et continuera sur les lignes suivantes à partir de la première colonne. Ceci est utile lorsque vous désirez reprendre une recherche sur une ligne alors qu'une occurrence a déjà été trouvée. Le paramètre *colonne* permet alors de « sauter » cette occurrence, et de reprendre la recherche après.

**Note** : les modules *Forth / Assembler* et *Text Editor* offrent la fonction SEARCH. La fonctionnalité de FILEPOS est équivalente, mais se distingue par trois points essentiels :

- la recherche utilise les chaînes génériques étendues (celles de XEDIT et pas celles de TEDIT),
- la syntaxe est bien plus simple, claire et mnémotechnique, et enfin,
- la valeur de retour est, elle aussi, plus simple.

Si vous désirez chercher une chaîne dans un fichier, et que cette chaîne est susceptible de contenir des caractères spéciaux, il faut annuler la signification de ces caractères. Il suffit de les faire précéder par un \, ce qui donne la commande suivante :

```
FILEPOS (fichier, GENRPLC$ (motif, "[\\^$\\.\\[\\]*", "\\&"))
```

## Références

*A paraître* : Première version de FILEPOS par Pierre David et Janick Taillandier.

*Unix User's Reference Manual* par AT&T.

## Mots-clés associés

ASSIGN #, CREATE, FIND, GENLEN, GENPOS, GENRPLC, PRINT #, READ #, XEDIT

## Auteurs

Pierre David, Janick Taillandier

FILESIZE renvoie la taille du fichier spécifié.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input type="checkbox"/> Opération d'unité

FILESIZE ( <i>fichier</i> )
-----------------------------

## Exemple

A=FILESIZE("ESSAI")

Renvoie la taille du fichier ESSAI s'il est présent, 0 sinon.

## Paramètre d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique.	Nom de fichier avec spécificateur de périphérique en option.

## Opération

FILESIZE renvoie la taille du fichier en octets, ou 0 si le fichier n'est pas présent en mémoire ou sur le support indiqué. On peut ainsi tester facilement l'existence d'un fichier, que ce soit en mémoire ou sur un périphérique externe. On fera donc un test similaire à :

```
1000 IF FILESIZE(F$&":TAPE") THEN
1010   COPY :TAPE TO F$
1020 END IF
```

La taille renvoyée est la taille *totale* du fichier. Cette taille comprend la place occupée par l'en-tête. Cette partie comprend le nom du fichier, son type, la date de création et d'autres informations utiles pour le système. Cette taille est donc différente de celle renvoyée par le catalogue CAT ou CAT\$.

L'intérêt d'utiliser cette taille est qu'elle correspond à la place disponible fournie par MEM. Pour copier un fichier d'un support magnétique vers une mémoire indépendante, il suffit d'un test du type :

```
IF MEM(0)>=FILESIZE("TOTO:TAPE") THEN COPY ...
```

## Références

JPC 23 (page 36) mot-clef FILE?

A paraître : FILESIZE par Henri Kudelski.

## Mots-clés associés

ADDR\$, CAT

## Auteur

Henri Kudelski

# FIND

FIND trouve une chaîne de caractère dans un programme Basic. FIND est non programmable.

<input checked="" type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

FIND chaîne
FIND chaîne , ligne départ
FIND chaîne , ligne départ , ligne fin

## Exemples

FIND "OSUB 1210"

Cherche la première occurrence de la chaîne "OSUB 1210" après la ligne courante, et y positionne le curseur.

FIND "[^\$]( [^ ]\*)=" , 1

Cherche, à partir de la première ligne du fichier, une affectation à un tableau numérique.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
ligne départ	Constante entière identifiant une ligne de programme. Défaut : ligne suivant la ligne courante.	1 à 9999.
Constante entière identifiant une ligne de programme.	Ligne de départ jusqu'à 9999. Défaut : dernière ligne du programme.	

## Opération

FIND cherche une chaîne de caractères dans le programme Basic édité, à partir de la ligne courante, ou à partir de la ligne *ligne départ* si spécifiée, et jusqu'à la dernière ligne, ou la ligne *ligne fin* si spécifiée.

Si FIND trouve la chaîne, il affiche la ligne concernée et positionne le curseur sur le premier caractère de la chaîne trouvée.

Si la chaîne n'est pas entre la ligne courante et la dernière ligne du programme, l'erreur : JPC ERR:Not Found est renvoyée.

FIND ne peut pas trouver la chaîne dans la première ligne du programme, à moins que le programme vienne d'être édité à l'aide de l'ordre EDIT ou que le paramètre *ligne départ* soit fourni.

La chaîne de recherche peut contenir des caractères génériques. Le tableau ci-dessous résume ces caractères. Pour plus de détails, voir XEDIT.

Caractère	Signification
-----	-----
\	annule la signification du caractère suivant
^	début de ligne
\$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^ ]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

## Références

JPC 31 (page 25) première version par Jean-Jacques Moreau.

JPC 45 (page 19) deuxième version par Janick Taillandier.

A paraître : troisième version par Janick Taillandier et Pierre David.

Fonction FETCH du HP-75.

## **Mots-clés associés**

FETCH, FILEPOS, GENPOS, XEDIT

## **Auteurs**

Jean-Jacques Moreau, Janick Taillandier

# FINPUT

FINPUT (Formatted INPUT) crée un masque de saisie, et attend l'introduction de l'utilisateur.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

```
FINPUT input , prompt , attn  
FINPUT input , prompt , format , attn
```

## Exemple

```
10 DIM I$[8]  
20 FINPUT I$,"Fichier: ",A  
30 IF A=0 THEN ...
```

L'utilisateur entre le nom du fichier (pas plus de 8 caractères), et FINPUT le place dans la variable I\$. Si L'utilisateur appuie sur la touche [ATTN], la variable A est mise à 0.

## Paramètres d'entrée

Élément	Description	Restriction
input	Nom de variable ou de tableau alphanumérique existant.	La variable ou le tableau doit être créé avant d'utiliser FINPUT.
prompt	Expression ou nom de tableau alphanumérique.	Ne doit pas contenir de caractères non affichables.
format	Expression ou nom de tableau alphanumérique. Défaut : STR\$(LEN(prompt))&"PU"	Chaîne de longueur non nulle, composée des caractères 'U' et 'P' et de chiffres indiquant un format.
attn	Nom de variable ou élément de tableau numérique.	Aucune.

## Opération

### Les champs protégés :

En quelques mots, les champs protégés servent lors d'un INPUT ou LINPUT à isoler des caractères et empêcher l'utilisateur du programme de les effacer ou d'écrire autre chose à la place.

A titre d'exemple, pour entrer une date, l'affichage devra avoir l'aspect suivant :

Date: Dy/Mo/Yr

L'utilisateur devra remplacer les seuls caractères Dy (Day), Mo (Month) et Yr (Year) par leur valeur. Les autres ne doivent pas changer. Voici le programme pour atteindre ce but :

```
100 E$=ESC$("<")           ! Eteint le curseur  
110 A$=ESC$(">")           ! Allume le curseur  
120 D$=A$&" Jr"&E$/"&A$&"Mo"&E$/"&A$&"Yr"  
130 DISP E$&"Date: "&D$&E$;           ! Affichage  
140 INPUT "" ;I$           ! Introduction de la date
```

Première observation : ce programme est illisible malgré les commentaires.

Deuxième observation : l'affichage du masque de saisie est lent.

Troisième observation : si une date est rentrée, puis la touche [ATTN] pressée, le mois s'efface et le curseur revient au début. Un deuxième appui sur [ATTN] ne provoque aucune réaction. La touche [ATTN] est inactive, il n'y a aucun moyen d'arrêter le programme. La seule solution est d'appuyer sur [ENDLINE]. Et la date est validée telle quelle, malgré une volonté évidente d'arrêter!

Quatrième observation : si la date est rentrée, rien n'empêche de continuer à rentrer des caractères. Comment y remédier?

Le problème vient de ce qu'aucun caractère n'est plus protégé après la date. Le HP-71 n'a donc plus aucune raison de bloquer le reste de l'affichage. Pour y arriver, il faut calculer : il faut afficher un masque de 14 caractères, et les autres doivent être tous protégés. Les autres, cela fait 96-14, c'est à dire 82 espaces à protéger. il faut donc ajouter quelques lignes :

```
121 DIM S$[82]
```

```
122 S$=""
123 S$[82]=" "
```

Puis remplacer la ligne 130 par :

```
129 WIDTH INF
130 DISP E$&"Date: "&D$&E$&S$;
```

A l'exécution, il y a un clignotement indésirable de l'affichage, puis tout s'affiche. Il n'y a enfin aucun moyen de rentrer de caractères au-delà de la date.

Mais il y a un nouveau problème : un appui sur la touche [->] après la deuxième année provoque un dérobement de l'affichage sur la gauche. Mieux encore : un appui sur sur [g] [->] aboutit à un écran vide après un temps assez long.

## Utilisation de FINPUT :

### FINPUT mono-ligne

Dans sa forme la plus simple, FINPUT est une extension de l'ordre LINPUT pour manipuler facilement les champs protégés.

Le programme Basic qui a servi d'exemple se résume alors à la forme suivante :

```
100 DIM I$[6]
110 FINPUT I$,"Date: Dy/Mo/Yr", "6P2UP2UP2UP",A
```

Dans cet exemple, il faut constater que :

- I\$ est la chaîne de destination. Elle doit être créée avant d'utiliser FINPUT.
- Ensuite vient le *prompt*. C'est ce qui apparaîtra sur l'écran. Tous les caractères à afficher y figurent, protégés ou non.
- Le paramètre suivant est le *format*. Examinons de près ce que cette expression contient : le 6P signifie que les six premiers caractères sont protégés (*Protected*). Le 2U précise que les deux caractères suivants sont «non protégés» (*Unprotected*). Le P indique ensuite que le caractère suivant est protégé, et ainsi de suite... Le P final signifie que le reste de l'affichage est protégé. Il n'y a donc pas besoin de mettre 82P pour terminer la ligne.
- Enfin vient la variable *attn*. En sortie, elle vaudra 0 si la touche [ATTN] a été utilisée pour sortir de FINPUT.

Comme vous pouvez le constater, le problème de l'utilisation des champs protégés est simplifié à l'extrême. Mais, ce n'est pas tout ce qu'apporte FINPUT. Parmi ses nombreux avantages, nous pouvons citer :

- gestion de la touche [ATTN] : lors d'un FINPUT, la touche [ATTN] sert une première fois pour revenir à l'affichage par défaut (la chaîne *prompt*). La deuxième fois, [ATTN] sort de FINPUT en mettant la variable *attn* à 0. Le programme n'est pas interrompu, et il est facile de gérer la touche [ATTN]. Un simple test : IF NOT A THEN . . . suffit.
- gestion des touches [->] et [g] [->] : les touches [->] et [g] [->] n'ont plus l'effet désagréable qu'elles avaient lors du premier exemple.
- gestion des «variables courtes» : dans l'exemple ci-dessus, si la variable I\$ avait été déclaré à moins de 6 caractères, par exemple 3, il n'aurait pas été possible d'en rentrer plus de 3. FINPUT met cette sécurité supplémentaire. Les programmes ne peuvent plus s'arrêter avec l'erreur : "String Overflow" !

### FINPUT sans chaîne de format :

Dans bien des cas, il n'y a pas besoin d'une gestion sophistiquée de l'affichage. Par exemple, pour rentrer le nom d'un fichier. Avec INPUT, vous feriez :

```
100 INPUT "Fichier: ";F$
```

Mais un nom de fichier en mémoire ne peut faire plus de 8 caractères. Avec FINPUT, le programme devient :

```
100 DIM F$[8]
110 FINPUT F$,"Fichier: ",A
120 IF NOT A THEN END
```

Il est maintenant impossible de rentrer plus de 8 caractères, et si l'utilisateur désire se rétracter en appuyant sur [ATTN], le programme l'autorise de la manière la plus simple.

Ce *format* est donc facultatif. Si il n'est pas présent, FINPUT le crée. Par défaut, cette chaîne de format est : STR\$(LEN(*prompt*))&"PU". C'est à dire que tous les caractères de la chaîne de *prompt* sont protégés. Les suivants sont tous non protégés, jusqu'à la dimension maximum de la chaîne résultat.

# FINPUT (suite)

*FINPUT sur plusieurs lignes :*

Mais la principale caractéristique de FINPUT est la gestion de plusieurs lignes de saisie. C'est, en quelque sorte, un «masque d'écran» complet.

Si un programme doit entrer la date et l'heure, ce pourrait être :

```
100 DIM D$(6),H$(6)
110 FINPUT D$,"Date: Dy/Mo/Yr","6P2UP2UP2UP",A
120 IF NOT A THEN END
130 ! traitement de la date
:
200 FINPUT H$,"Heure: Hr:Mn:Sc","7P2UP2UP2UP",A
210 IF NOT A THEN END
220 ! traitement de l'heure
```

Mais il y a une autre solution :

```
100 OPTION BASE 1 ! Pour démarrer les tableaux à 1
110 DIM I$(2)[6],M$(2),P$(2)
120 DATA Date: Dy/Mo/Yr,Heure: Hr:Mn:Sc
130 DATA 6P2UP2UP2UP,7P2UP2UP2UP
160 READ M$ ! Lit les 2 messages
170 READ P$ ! Lit les 2 chaînes de protection
180 FINPUT I$,M$,P$,A
190 IF NOT A THEN END
200 ! traitement de la date (I$(1))
210 ! traitement de l'heure (I$(2))
```

Cette dernière solution est beaucoup plus pratique que la première quand on doit entrer de grandes quantités de données. L'introduction se fait en une seule opération.

Les touches de curseur vertical servent à passer d'une ligne à l'autre. La touche [ENDLINE] sert à valider chaque ligne. La touche [RUN] sert à valider la ligne et à sortir de FINPUT.

**Important :** il y a deux possibilités de sortir de FINPUT en validant la fiche :

- la touche [RUN], qui valide la ligne courante, et
- deux pressions successives sur [ENDLINE] lorsque le curseur est sur le dernier champ.

La variable *attn* contient alors le numéro de la ligne sur laquelle on est sorti. La valeur 0 indique toujours une sortie par [ATTN].

Cette forme d'utilisation de FINPUT permet de saisir des fiches complètes en une seule opération. Le programmeur n'a donc plus à gérer tous les déplacements à l'intérieur de cette fiche, FINPUT le fait à sa place !

## Résumé :

La variable *input* doit être créée avant de commencer à exécuter FINPUT.

La chaîne *prompt* ne doit contenir que des caractères affichables, c'est à dire des caractères dont le code n'est ni 0 (NULL), ni 27 (ESC), ni 13 (CR), ni 10 (LF), ni 8 (BS).

La spécification de format doit contenir les lettres "U" et "P" (en majuscules ou minuscules) précédées ou non d'un facteur de répétition pour indiquer les caractères protégés ou non. La chaîne doit être non nulle, et le format résultant ne doit pas dépasser 96 caractères. Ainsi, si 96P est légal, 97P ne l'est pas, non plus que 95P2U.

Les variables simples sont assimilées à des tableaux à un seul élément.

## L'utilisation :

Sous FINPUT, la signification des touches est la suivante :

[ATTN]

Si des caractères ont été entrés, restauration de l'affichage par défaut, c'est à dire du *prompt* tel quel.

Une deuxième fois : sortie de FINPUT.

[f][OFF]

Sortie directe de FINPUT.

[ENDLINE]

Validation de la ligne courante et passage à la ligne suivante. Si il n'y a qu'une seule ligne, sort de FINPUT.

Une deuxième fois sur la dernière ligne : sort de FINPUT en validant la fiche.

[RUN]

Validation de la ligne courante et sort de FINPUT. Si il n'y a qu'une seule ligne, [RUN] est identique à [ENDLINE].

[↑], [↓], [g][↑] et [g][↓]

Change de ligne sans valider la ligne courante. Si une seule ligne, remet l'affichage par défaut.

*Contenu des variables en sortie :*

Les variables *prompt* et *format* ne sont jamais modifiées.

En cas de sortie normale (par [ENDLINE] ou [RUN]), la variable *attn* contient le numéro logique de la ligne sur laquelle on est sorti. Ce numéro varie entre 1 et la dimension du tableau, même si le tableau avait été déclaré en `OPTION BASE 0`.

La variable *destination* contient les valeurs introduites.

En cas de sortie prématurée (par [ATTN] ou [f][OFF]), la variable *attn* contient 0. La variable *destination* n'est pas changée.

## Références

JPC 43 (page 16) FINPUT par Pierre David et Janick Taillandier.

## Mots-clés associés

INPUT, LINPUT, DISP

## Auteurs

Pierre David, Janick Taillandier

# FKEY

FKEY (First KEY) pousse une touche en premier dans la mémoire tampon de touches.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

FKEY touche
-------------

## Exemple

```
10 DISP "Résultat =";R  
20 K$=KEYWAIT$  
30 FKEY K$  
40 INPUT X$
```

Affiche le résultat contenu dans la variable R, puis attend une pression de touche pour continuer l'exécution. La touche n'est pas perdue, et sera le premier caractère de l'introduction de données suivante.

## Paramètre d'entrée

Élément	Description	Restriction
touche	Expression alphanumérique.	Moins de cinq caractères.

## Opération

FKEY ajoute la touche spécifiée en tête de la mémoire tampon (qui contient 15 pressions de touches) à la différence de PUT qui l'ajoute en queue.

Si la mémoire tampon est pleine, la touche la plus ancienne (c'est à dire celle qui a été frappée en premier) est perdue.

FKEY permet d'établir un système de priorités, dans lequel certaines instructions (touches assignées) seront traitées avant d'autres entrées au clavier.

## Références

JPC 24 (page 35) première version par Jean-Pierre Bondu.

## Mots-clés associés

KEY\$, KEYWAIT\$, PUT

## Auteur

Jean-Pierre Bondu

FORMAT\$ insère des espaces supplémentaires dans la chaîne de façon à ce qu'elle fasse exactement le nombre de caractères spécifié.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

FORMAT\$ ( chaîne , largeur )

## Exemple

A\$=FORMAT\$("P P C",9)

Répartit quatre espaces dans la chaîne spécifiée.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
largeur	Expression numérique arrondie à un entier.	1 à 1048575.

## Opération

FORMAT\$ réduit d'abord la chaîne (voir REDUCE\$). Des espaces sont ensuite ajoutés de façon à ce que la chaîne finale fasse exactement la largeur spécifiée.

Les espaces sont ajoutés entre les mots.

Ceci facilite grandement la réalisation d'une justification des textes à gauche et à droite.

## Références

JPC 21 (page 34) première version du programme de formattage en Basic par Pierre David.

JPC 26 (page 50) deuxième version du programme en Basic avec les fonctions en assembleur par Pierre David et Michel Martinet.

## Mots-clés associés

CESURE, REDUCE\$

## Auteurs

Pierre David, Michel Martinet

# FPRIM

FPRIM (First PRIME number) renvoie le premier nombre premier à partir d'un nombre donné.

- |                                    |                        |
|------------------------------------|------------------------|
| <input type="checkbox"/> Ordre     | ■ Exécution au clavier |
| ■ Fonction                         | ■ Mode CALC            |
| <input type="checkbox"/> Opérateur | ■ IF...THEN...ELSE     |
|                                    | ■ Opération d'unité    |

FPRIM ( *argument* )  
FPRIM ( *argument* , *direction* )

## Exemples

A=FPRIM(300)

Place le nombre 307, premier nombre premier à partir de 300, dans la variable A.

A=FPRIM(300, 350)

Place le nombre 307, premier nombre premier entre 300 et 350, dans la variable A.

DISP FPRIM(300, 250)

Affiche le nombre 293, plus grand nombre premier inférieur à 300 et supérieur à 250.

## Paramètres d'entrée

Élément	Description	Restriction
argument	Expression numérique.	Doit être un nombre entier non nul compris entre $-10^{12}+1$ et $10^{12}-1$ .
direction	Expression numérique. Défaut : $10^{12} * \text{SGN}(\text{argument})$ .	Doit être un nombre entier compris entre $-10^{12}+1$ et $10^{12}-1$ .

## Opération

FPRIM cherche le premier nombre premier à partir de l'argument spécifié, ou renvoie l'argument si il est lui-même premier. Le deuxième paramètre, *direction*, indique si la recherche doit être faite vers les nombres supérieurs (*direction* > *argument*) ou vers les nombres inférieurs (*direction* < *argument*).

*Direction* est également utilisée comme limite supérieure ou inférieure de la recherche. FPRIM renvoie 0 si il n'y a pas de nombre premier compris entre *argument* et *direction*.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur JPC ERR:Function Interrupted.

## Références

JPC 35 (page 21) première version de DIVILEX par Guy Toublanc.

JPC 38 (page 18) deuxième version par Guy Toublanc.

JPC 48 (page 23) troisième version par Guy Toublanc.

L'ancienne appellation de FPRIM était FPRM.

## Mots-clés associés

PRIM, NPRIM, DIV

## Auteur

Guy Toublanc

FRAC\$ (FRACtion) approxime un nombre réel par une fraction.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

FRAC\$ ( *nombre réel* )  
 FRAC\$ ( *nombre réel* , *précision* )

## Exemples

A\$=FRAC\$(1.25)

Renvoie la chaîne "5/4" dans la variable A\$.

DISP FRAC\$(PI,2)

Affiche la chaîne "22/7", approximation de PI à 10<sup>-2</sup> près.

## Paramètres d'entrée

Élément	Description	Restriction
nombre réel	Expression numérique.	Aucune.
precision	Expression numérique arrondie à un entier. Défaut : si $ n  > 1$ , 10, sinon 10-exposant de n.	-99 à 99

## Opération

FRAC\$ approxime le nombre réel  $x$  passé en paramètre en une fraction rationnelle exprimée sous forme d'une chaîne alphanumérique.

La précision est optionnelle. Si elle est omise, ou si elle vaut 0, elle est prise par défaut à 10<sup>-10</sup> si  $|x| > 1$ , ou 10<sup>-10+exposant de x</sup> sinon.

Si le paramètre *précision* est positif, la précision vaut 10<sup>-*précision*</sup>.

Si le paramètre *précision* est négatif, il s'agit du nombre d'itérations de l'algorithme des réduites successives.

## Références

JPC 20 (page 55) première version du programme en Basic par Pierre David.

JPC 42 (page 21) FRAC\$ première version par Guy Toublanc.

## Mots-clés associés

EXPONENT

## Auteur

Guy Toublanc

# FRAME

FRAME trace un cadre autour de la zone de traçage courante.

<input checked="" type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

FRAME
-------

## Exemple

```
100 IF F THEN FRAME
```

Trace un cadre autour de la zone de traçage si le contenu de la variable F est non nul.

## Opération

Après le traçage, la plume est positionnée dans le coin inférieur gauche, et elle est levée.

Le cadre est tracé suivant le dernier ordre LINETYPE exécuté.

## Mots-clés associés

LINETYPE, BOX

## Auteur

Pierre David

## GDUMP (Graphic DUMP)

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input type="checkbox"/> Opération d'unité               |

GDUMP

## Opération

GDUMP imprime le contenu du fichier GRAPHILE sur l'appareil désigné par l'ordre PLOTTER IS.

L'appareil désigné doit accepter les graphiques définis dans le PCL (Printer Control Language de Hewlett-Packard). Les imprimantes ThinkJet, QuietJet et LaserJet répondent à GDUMP.

Le fichier GRAPHILE n'est pas modifié par GDUMP. C'est à dire qu'une session graphique peut contenir plusieurs ordres GDUMP. Ceci peut être utile pour imprimer les étapes intermédiaires du traçage.

Si l'ordre PLOTTER IS n'a pas été exécuté, GDUMP déclenche l'erreur : "Plotter not reachable".

## Mots-clés associés

PLOTTER IS, GINIT

## Auteur

Pierre David

# GEND

GEND (Graphic END) termine une session graphique ouverte par GINIT.

- Ordre
- Fonction
- Opérateur

- Exécution au clavier
- Mode CALC
- IF...THEN...ELSE
- Opération d'unité

GEND

## Opération

GEND achève une session graphique ouverte par un ordre GINIT. Le fichier GRAPHILE est effacé de la mémoire, ainsi que les zones nécessaires au stockage des informations d'état.

GEND ne modifie pas la spécification de l'appareil désigné par PLOTTER IS.

GEND interdit toute opération autre que PLOTTER IS ou GINIT. Une tentative d'exécution d'une fonction graphique déclenche l'erreur :

GRPH ERR :Graph not initialized

## Mots-clés associés

GINIT, PLOTTER IS

## Auteur

Pierre David

GENLEN (GENeric LENgth) cherche une chaîne générique dans une chaîne alphanumérique et retourne la longueur de l'occurrence trouvée.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

```
GENLEN ( chaîne , motif )
GENLEN ( chaîne , motif , début )
```

## Exemples

```
DISP GENLEN ("JPC Rom", "Rom")
```

Affiche 3, c'est à dire la longueur de la chaîne "Rom" à l'intérieur de la chaîne "JPC Rom".

```
A=GENLEN (L$, "[0-9][0-9]*", X+1)
```

Place dans la variable A la longueur du premier nombre trouvé dans la chaîne L\$ à partir du caractère X+1.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
motif	Expression alphanumérique.	Expression générique valide.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575

## Opération

La fonction GENLEN recherche la chaîne générique *motif* dans la chaîne *chaîne*. Si une occurrence est trouvée, sa longueur (éventuellement nulle) est renvoyée, sinon la valeur 0 est renvoyée.

Attention : la valeur 0 est ambiguë, car elle peut indiquer à la fois une occurrence de taille nulle (ce qui peut arriver lorsque vous utilisez le caractère générique \* par exemple), et à la fois une occurrence non trouvée. Pour lever l'ambiguïté, nous vous conseillons de tester au préalable la présence de l'occurrence avec GENPOS, puis de n'utiliser GENLEN que lorsque vous êtes sûr qu'il y a une occurrence. Cette ambiguïté est préférable, car ainsi les deux fonctions soeurs GENPOS et GENLEN renvoie la même valeur en cas d'occurrence non trouvée, de même que la fonction standard POS.

Les caractères génériques utilisables dans la chaîne *motif* sont résumés dans le tableau ci-dessous. Pour plus de détails, voir XEDIT.

Caractère	Signification
-----	-----
\	annule la signification du caractère suivant
^	début de ligne
\$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^ ]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

Si le paramètre numérique *début* est présent, il spécifie à partir de quel caractère de *chaîne* la recherche doit commencer. Par défaut, *motif* est cherché dans toute la chaîne *chaîne*. Attention : le caractère générique ^ (début de chaîne) correspond au début de la chaîne et non à la position indiquée par *début*. Ainsi, la commande GENLEN("ABCD", "^B", 2) retournera 0.

## Références

*A paraître* : Première version de GENPOS par Pierre David et Janick Taillandier.

*Unix User's Reference Manual* par AT&T.

# GENLEN (suite)

## Mots-clés associés

XEDIT, FILEPOS, FIND, GENPOS, GENRPLC, LEN

## Auteurs

Pierre David, Janick Taillandier

GENPOS (GENeric POSition) cherche une chaîne générique dans une chaîne alphanumérique et retourne la position de l'occurrence trouvée.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

GENPOS ( chaîne , motif )  
 GENPOS ( chaîne , motif , début )

## Exemples

DISP GENPOS ("JPC Rom", "Rom")

Affiche 4, c'est à dire la position de la chaîne "Rom" à l'intérieur de la chaîne "JPC Rom".

A=GENPOS (L\$, "[0-9]", X+1)

Place dans la variable A la position du premier chiffre dans la chaîne L\$ à partir du caractère X+1.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
motif	Expression alphanumérique.	Expression générique valide.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575

## Opération

La fonction GENPOS recherche la chaîne générique *motif* dans la chaîne *chaîne*. Si une occurrence est trouvée, sa position dans la chaîne *chaîne* est renvoyée, sinon la valeur 0 est renvoyée.

Les caractères génériques utilisables dans la chaîne *motif* sont résumés dans le tableau ci-dessous. Pour plus de détails, voir XEDIT.

Caractère	Signification
-----	-----
\	annule la signification du caractère suivant
^	début de ligne
\$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^ ]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

Si le paramètre numérique *début* est présent, il spécifie à partir de quel caractère de *chaîne* la recherche doit commencer. Par défaut, *motif* est cherché dans toute la chaîne *chaîne*. Attention : le caractère générique ^ (début de chaîne) correspond au début de la chaîne et non à la position indiquée par *début*. Ainsi, la commande GENPOS("ABCD", "^B", 2) retournera 0.

## Références

*A paraître* : Première version de GENPOS par Pierre David et Janick Taillandier.

*Unix User's Reference Manual* par AT&T.

## Mots-clés associés

XEDIT, FILEPOS, FIND, GENLEN, GENRPLC, POS

# GENPOS (suite)

## Auteurs

Pierre David, Janick Taillandier

GENRPLC\$ (GENeric RePLaCe) cherche un motif générique dans une chaîne alphanumérique, et substitue la sous-chaîne trouvée par la chaîne de remplacement.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

GENRPLC\$ ( chaîne , motif , remplacement )  
 GENRPLC\$ ( chaîne , motif , remplacement , début )

## Exemples

DISP GENRPLC\$ ("Math Rom", "Math", "JPC")

Affiche la chaîne "JPC Rom", c'est à dire la chaîne d'origine avec toutes les occurrences de "Math" remplacées par "JPC".

A\$=GENRPLC\$ (L\$, "[0-9][0-9]\*", "-&", X+1)

Place dans la variable A\$ la chaîne origine avec tous les nombres rendus positifs : un signe "-" est placé devant chaque nombre. La chaîne de remplacement signifie : un signe "-" et l'occurrence trouvée.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
motif	Expression alphanumérique.	Expression générique valide.
remplacement	Expression alphanumérique.	Aucune.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575

## Opération

La fonction GENRPLC\$ substitue, dans la chaîne *chaîne*, toutes les occurrences de la chaîne *motif* par la chaîne *remplacement*.

La fonction GENRPLC\$ est similaire à la fonction REPLACE\$. La fonction REPLACE\$ est, elle-même, le regroupement de deux fonctions plus anciennes, REPLACE\$ et RPLC\$. Cette fonction REPLACE\$ n'est laissée dans JPC Rom que pour des raisons de compatibilité ascendante, et vous êtes plutôt encouragés à utiliser GENRPLC\$. Le tableau ci-dessous résume les différences entre ces trois fonctions :

Fonction	Ancien nom	Caractéristiques
REPLACE\$	REPLACE\$	Remplacement simple avec caractère joker
REPLACE\$	RPLC\$	Chaînes génériques compatibles avec EDTEXT
GENRPLC\$	GENRPLC\$	Chaînes génériques compatibles avec XEDIT

Note : l'ancienne fonction REPLACE\$ correspond au cas où l'actuelle fonction REPLACE\$ est appelée avec un quatrième paramètre alphanumérique représentant le joker. Dans les autres cas d'appel, c'est l'ancienne fonction RPLC\$ qui est exécutée.

Les chaînes *motif* et *remplacement* peuvent contenir des caractères génériques. Le tableau ci-dessous résume ces caractères. Pour une explication détaillée, voir XEDIT.

Les caractères génériques de la chaîne *motif* sont :

Caractère	Signification
\	annule la signification du caractère suivant
^	début de ligne
\$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères

# GENRPLC\$ (suite)

[^ ]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

Dans la chaîne *remplacement*, les caractères génériques sont :

Caractère	Signification
-----	-----
\	annule la signification du caractère suivant
&	substitue l'occurrence trouvée

Si le paramètre numérique *début* est présent, il spécifie à partir de quel caractère de *chaîne* la recherche et la substitution doivent commencer. Par défaut, *motif* est cherché dans toute la chaîne *chaîne*. Attention : le caractère générique ^ (début de chaîne) correspond au début de la chaîne et non à la position indiquée par *début*. Ainsi, la commande GENRPLC\$("ABCD", "^B", "X", 2) retournera la chaîne "ABCD".

Si vous désirez remplacer une chaîne susceptible de contenir des caractères spéciaux dont vous voulez annuler la signification, c'est à dire si vous désirez opérer un remplacement non générique, vous ferez :

```
GENRPLC$ (chaîne, GENRPLC$ (motif, "[\\^$\\.\\[\\]*", "\\&"),
          GENRPLC$ (remplacement, "[\\&]", "\\&"))
```

## Références

*A paraître* : Première version de GENRPLC\$ par Pierre David et Janick Taillandier.

*Unix User's Reference Manual* par AT&T.

## Mots-clés associés

XEDIT, GENLEN, GENPOS, REPLACES

## Auteurs

Pierre David, Janick Taillandier

GINIT (Graphic INITialisation) initialise une session graphique.

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> Ordre<br><input type="checkbox"/> Fonction<br><input type="checkbox"/> Opérateur | <input checked="" type="checkbox"/> Exécution au clavier<br><input type="checkbox"/> Mode CALC<br><input checked="" type="checkbox"/> IF...THEN...ELSE<br><input checked="" type="checkbox"/> Opération d'unité |
|--|---|

GINIT "RASTER", *dimension*

## Paramètre d'entrée

Élément	Description	Restriction
hauteur	Expression numérique arrondie à un entier.	1 à 1048575

## Opération

GINIT ouvre une session graphique. Toutes les fonctions graphiques deviennent autorisées.

Le paramètre hauteur définit la taille de la zone de traçage. Les coordonnées sur l'axe des x varient entre 0 et 639, les coordonnées sur l'axe des y varient entre 0 et hauteur-1.

GINIT crée un fichier GRAPHILE pour conserver une image de la zone de traçage. Il réserve aussi de la mémoire pour les informations d'état telles que position de la plume, type de ligne courant, taille des caractères, etc.

La place occupée par le fichier GRAPHILE est définie par :

$$place\ occupée = 19 + hauteur * 80\ octets$$

Les conditions de traçage par défaut sont :

- Position initiale de la plume = (0,0)
- Type de tracé = ligne pleine
- Taille des caractères = 20 de hauteur, 12 de largeur
- Origine des labels = 1
- Direction des labels = 0 degrés
- Longueur des marques pour les axes = 4 unités graphiques

## Mots-clés associés

GEND, PLOTTER IS

## Auteur

Pierre David

# GLINE

GLINE (Graphic LINE) prépare le tracé d'une ligne graphique sur les imprimantes ThinkJet ou LaserJet.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

GLINE *x* , *longueur* , *premier* , *taille* , *écart*

## Exemple

GLINE x1,x2-x1+1,1,1,0

Trace une ligne entre les points x1 et x2 en utilisant la variable G\$.

## Paramètres d'entrée

Élément	Description	Restriction
x	Expression numérique arrondie à un entier.	1 à 640.
longueur	Expression numérique arrondie à un entier.	0 à 640.
premier	Expression numérique arrondie à un entier.	0 à 640.
taille	Expression numérique arrondie à un entier.	0 à 640.
écart	Expression numérique arrondie à un entier.	0 à 640.

## Opération

### Les graphiques sur les imprimantes « raster » :

Les imprimantes capables de dessiner des graphiques du type « raster » sont, par exemple, les ThinkJet, LaserJet ou QuietJet.

Les graphiques sur ces imprimantes sont construits ligne par ligne. Il faut donc un espace de 80 caractères pour recevoir une image des 640 points de cette ligne. GLINE et GPSET utilisent donc une variable de 80 caractères nommée explicitement G\$, dimensionnée au préalable par un ordre DIM G\$[80].

Avant le début de chaque nouvelle ligne, il faudra remplir cette variable avec des CHR\$(0) pour l'initialiser. Ceci peut se faire par un simple segment de programme du type :

```
100 DIM G$[80]
110 G$=SPACE$(0,80)
```

### L'impression de la ligne graphique :

L'impression se fait à l'aide d'une séquence d'échappement (voir ESC\$), dépendante de l'imprimante dont vous disposez. Avec une ThinkJet, vous ferez :

```
PRINT ESC$("*r640S"); @ PWIDTH INF
```

pour initialiser le mode graphique. Notez que cette initialisation ne devra être faite qu'une seule fois pour toute la durée de votre programme.

Vous imprimerez ensuite chaque ligne graphique en faisant :

```
PRINT ESC$("*b80W")&G$;
```

Attention : faites de préférence un PWIDTH INF avant d'imprimer le graphique, pour éviter que le HP-71 n'insère une séquence de fin de ligne (voir ENDLIN) au milieu de vos données graphiques.

Reportez-vous au manuel de votre imprimante pour plus de détails sur le fonctionnement des graphiques.

### L'utilisation de GLINE :

GLINE sert à tracer une droite dans l'«image» G\$, à partir du point x, de longueur *longueur*. La caractéristique de GLINE est de tracer des jolis motifs. Par exemple :

```
premier
v  taille  écart
<---><-----><----->
```

---

```
|      xxxxxxxxxxxx      xxxxxxxxxxxx      xxxxxxxxxxxx
|      xxxxxxxxxxxx      xxxxxxxxxxxx      xxxxxxxxxxxx
| xxxxxxxxxxxx      xxxxxxxxxxxx      xxxxxxxxxxxx
|
```

Ceci était un petit exemple de motif. Chaque ligne est construite en donnant 5 paramètres :

- 1) La coordonnée du premier point : *x*
- 2) la longueur de la ligne : *longueur*
- 3) le premier écart : *premier*
- 4) la taille d'un trait : *taille*
- 5) l'écart entre les traits : *écart*.

Pour dessiner un motif comme celui-ci, il suffit de faire varier *premier* :

$premier = \text{MOD} ( premier - 2 , écart )$

Pour tracer une ligne droite normale, prenez :

$premier = taille = 1$

$écart = 0$

## Références

JPC 35 (page 38) GLINE et le sous programme GRAPH par Pierre David.

JPC 42 (page 29) exemple d'utilisation par Eric Gengoux.

## Mots-clés associés

ENDLINE, ESC\$, GPSET, PRINT, PWIDTH, Fonctions graphiques de JPC Rom

## Auteur

Pierre David

# GPSET

GPSET (Graphic Point SET) prépare le tracé d'un point sur les imprimantes ThinkJet ou LaserJet.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

GPSET x

## Exemple

```
FOR X=X1 TO X2 @ GPSET X @ NEXT X
```

Trace une ligne de x1 à x2.

## Paramètre d'entrée

Élément	Description	Restriction
x	Expression numérique arrondie à un entier.	1 à 640.

## Opération

GPSET trace un point dans l'«image» de la ligne G\$. Pour plus d'explications, voir GLINE.

## Références

JPC 35 (page 38) GPSET et le sous programme GRAPH par Pierre David.

JPC 42 (page 29) exemple d'utilisation par Eric Gengoux.

## Mots-clés associés

ENDLINE, ESC\$, GLINE, PRINT, PWIDTH, Fonctions graphiques de JPC Rom

## Auteur

Pierre David

HMS (Hour Minute Second) convertit des heures ou des degrés en leur équivalent au format HMS.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

HMS ( *argument* )

## Exemple

A=HMS(121.5)

121 degrés décimaux équivalent à 121 degrés 30'.

## Paramètre d'entrée

Élément	Description	Restriction
argument	Expression numérique.	Aucune.

## Opération

### Le format HMS :

Les fonctions HMS, HR, HMSADD et HMSSUB opèrent sur des quantités de temps (ou des quantités angulaires) exprimées à l'aide de nombres réels sous la forme HMS (heures, minutes, secondes).

Le format HMS est *h.mmssd*, où :

- *h* = zéro ou plusieurs chiffres représentant la partie entière du nombre,
- *mm* = deux chiffres représentant le nombre de minutes,
- *ss* = deux chiffres représentant le nombre de secondes, et
- *d* = zéro ou plusieurs chiffres représentant la partie fractionnaire décimale de secondes.

Ces nombres sont indépendants du mode angulaire courant (voir OPTION ANGLE).

### La fonction HMS :

HMS convertit un nombre réel représentant des heures décimales (ou des degrés) en format HMS.

HMS est la fonction inverse de HR.

## Références

*JPC 25* (page 52) première version par Michel Martinet.

*JPC 50* (page 29) deuxième version par Guy Toublanc.

*Manuel d'utilisation du HP-41.*

*Manuel de référence du HP-28C.*

## Mots-clés associés

HMSADD, HMSSUB, HR, fonctions trigonométriques

## Auteurs

Michel Martinet, Guy Toublanc

# HMSADD

HMSADD (Hour Minute Second ADDition) renvoie la somme des deux arguments interprétés en format HMS.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

HMSADD ( *arg1* , *arg2* )

## Exemple

A=HMSADD(12.3456, -20.1721)

Place dans la variable A la somme de 12 heures 34'56" et de -20 heures 17'21", soit -7.4225 ou -7 heures, 42 minutes et 25 secondes.

## Paramètre d'entrée

Élément	Description	Restriction
arg1, arg2	Expressions numériques interprétées selon le format HMS.	Aucune.

## Opération

HMSADD additionne les deux arguments interprétés comme des nombres sexagésimaux au format HMS. La valeur retournée est elle aussi dans le format HMS.

Voir la fonction HMS pour le détail du format HMS.

## Références

*JPC 25* (page 52) première version par Michel Martinet.

*JPC 50* (page 29) deuxième version par Guy Toublanc.

*Manuel d'utilisation du HP-41.*

*Manuel de référence du HP-28C.*

L'ancienne appellation de HMSADD était HMS+.

## Mots-clés associés

HMS, HMSSUB, HR, fonctions trigonométriques

## Auteurs

Michel Martinet, Guy Toublanc

HMSSUB (Hour Minute Second SUBstract) renvoie la soustraction des deux arguments interprétés en format HMS.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

HMSSUB ( *arg1* , *arg2* )

## Exemple

A=HMSSUB(12.3456,20.1721)

Place dans la variable A la différence de 12 heures 34'56" et de 20 heures 17'21".

## Paramètre d'entrée

Élément	Description	Restriction
arg1, arg2	Expressions numériques interprétées selon le format HMS.	Aucune.

## Opération

HMSSUB soustrait les deux arguments interprétés comme des nombres sexagésimaux au format HMS. La valeur retournée est elle aussi au format HMS.

Voir la fonction HMS pour les détails sur le format HMS.

## Références

*JPC 25* (page 52) première version par Michel Martinet.

*JPC 50* (page 29) deuxième version par Guy Toublanc.

*Manuel d'utilisation du HP-41.*

*Manuel de référence du HP-28C.*

L'ancienne appellation de HMSADD était HMS+.

## Mots-clés associés

HMS, HMSADD, HR, fonctions trigonométriques

## Auteurs

Michel Martinet, Guy Toublanc

# HR

HR (HouR) convertit un nombre réel exprimé au format HMS en son équivalent décimal.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

HR ( *argument* )

## Exemple

A=HR(121.3)

121 degrés 30' correspondent à 121.5 degrés décimaux.

## Paramètre d'entrée

Élément	Description	Restriction
argument	Expression numérique interprétée selon le format HMS.	Aucune.

## Opération

HR convertit les nombres du format HMS en nombres réels décimaux, directement utilisable par les fonctions trigonométriques.

La fonction HR est l'inverse de la fonction HMS. Voir cette dernière pour les détails sur le format HMS.

## Références

*JPC 25* (page 52) première version par Michel Martinet.

*JPC 50* (page 29) deuxième version par Guy Toublanc.

*Manuel d'utilisation du HP-41.*

*Manuel de référence du HP-28C.*

## Mots-clés associés

HMS, DEGREES, HMSADD, HMSSUB, fonctions trigonométriques

## Auteurs

Michel Martinet, Guy Toublanc



# IDRAW

IDRAW (Incremental DRAWing) trace un segment de droite à partir de la position courante de la plume jusqu'à la position spécifiée par les incréments en x et y.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

IDRAW *incrément x, incrément y*

## Exemple

IDRAW 10,20

trace un vecteur (10,20) à partir de la position courante de la plume.

## Paramètre d'entrée

Élément	Description	Restriction
incrément x, incrément y	Expressions numériques arrondies à des entiers.	-32767 à +32767

## Opération

IDRAW est semblable à DRAW, à ceci près que les paramètres sont des incréments et non des positions absolues. L'origine est la plus récente position de la plume.

Comme pour DRAW, aucun trait ne peut être tracé en dehors de la zone de tracé. Si un tracé dépasse cette limite, la position théorique de la plume sera changée, mais le trait s'arrêtera à la frontière.

Le trait est tracé suivant le type de ligne en vigueur.

## Mots-clés associés

DRAW, IMOVE, LINETYPE

## Auteur

Pierre David

# IF ... THEN ... ELSE ... END IF

La structure IF ... THEN ... ELSE ... END IF étend la structure standard au traitement multi-lignes.

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> Ordre<br><input type="checkbox"/> Fonction<br><input type="checkbox"/> Opérateur | <input checked="" type="checkbox"/> Exécution au clavier<br><input type="checkbox"/> Mode CALC<br><input checked="" type="checkbox"/> IF...THEN...ELSE<br><input checked="" type="checkbox"/> Opération d'unité |
|--|---|

```
IF expression logique THEN
  segment de programme
END IF
ou bien :
IF expression logique THEN
  segment de programme
ELSE
  segment de programme
END IF
```

## Exemples

```
100 IF X=0 THEN
110   INPUT X
120 ELSE
130   X=X+1
140 END IF
150 :
```

Si le contenu de la variable X est nul, le INPUT est exécuté, sinon la variable X est incrémentée.

```
100 IF X=0 THEN
110   INPUT X
120 END IF
130 :
```

Si le contenu de la variable X est nul, le INPUT est exécuté, sinon l'exécution continue à la ligne 130.

## Paramètres d'entrée

Élément	Description	Restriction
expression logique	Expression numérique évaluée comme vraie si différente de zéro et fausse si égale à zéro.	Aucune.
segment de programme	Tout nombre de lignes de programme contiguës.	Aucune.

## Opération

La structure IF ... ELSE ... END IF (ou IF ... END IF) de JPC Rom est une extension de la structure IF standard.

La différence est faite quand il y a une fin d'ordre après le THEN, c'est à dire une fin de ligne, un caractère @, une remarque (!) ou un ELSE mono-ligne.

Par exemple :

```
10 IF X=0 THEN           IF de JPC Rom
10 IF X=0 THEN @ BEEP    "
10 IF X=0 THEN ! Remarque "

10 IF X=0 THEN 'BEEP'    IF standard
10 IF X=0 THEN 10       "
10 IF X=0 THEN BEEP     "
```

Les programmes conçus avec des IF monolignes sont exécutables même lorsque JPC Rom n'est plus dans le HP-71.

Si l'évaluation de l'*expression logique* est vraie (différente de zéro), l'exécution se poursuit au premier ordre suivant le THEN. Si le ELSE est présent, l'exécution continue après le END IF, en sautant le block délimité par ELSE et END IF.

Si l'évaluation de l'*expression logique* est fausse (nulle), l'exécution continue immédiatement après le ELSE s'il est présent, ou après le END IF sinon.

# IF ... THEN ... ELSE ... END IF (suite)

Les segments de programme peuvent eux-mêmes contenir des structures de type boucle ou choix. L'emboîtement doit être correct, sinon l'erreur JPC ERR:Structure Mismatch est renvoyée.

## Références

*JPC 52* (page 10) première version par Pierre David et Janick Taillandier.  
Basic 5.0 pour HP 9000 séries 200/300.

## Mots-clés associés

IF ... THEN, SELECT ... END SELECT

## Auteurs

Pierre David, Janick Taillandier

IMOVE (Incremental MOVE) déplace la plume depuis la position courante de la plume jusqu'à la position spécifiée par les incréments en x et y.

<input checked="" type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

IMOVE <i>incrément x, incrément y</i>
---------------------------------------

## Exemple

IMOVE 30,40

déplace la plume de 30 unités graphiques selon l'axe des x, et de 40 unités graphiques selon l'axe des y.

## Paramètre d'entrée

Élément	Description	Restriction
incrément x, incrément y	Expressions numériques arrondies à des entiers.	-32767 à +32767

## Opération

IMOVE est semblable à MOVE, à ceci près que les paramètres sont des incréments et non des positions absolues.

Si un ordre IMOVE déplace la plume en dehors des limites du tracé, le prochain trait ne commencera qu'à partir de la frontière de la zone de traçage.

## Mots-clés associés

MOVE, IDRAW, LINETYPE

## Auteur

Pierre David

# INVERSE

INVERSE inverse des portions de l'affichage graphique.

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> Ordre<br><input type="checkbox"/> Fonction<br><input type="checkbox"/> Opérateur | <input checked="" type="checkbox"/> Exécution au clavier<br><input type="checkbox"/> Mode CALC<br><input checked="" type="checkbox"/> IF...THEN...ELSE<br><input checked="" type="checkbox"/> Opération d'unité |
|--|---|

INVERSE  
INVERSE *début* , *fin*

## Exemples

IF KEYDOWN("I") THEN INVERSE

Inverse l'affichage si la touche [I] est pressée.

INVERSE 10,121

Inverse l'affichage de la colonne 10 à la colonne 121, c'est à dire en laissant 10 colonnes inchangées de part et d'autre.

## Paramètres d'entrée

Élément	Description	Restriction
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 131
fin	Expression numérique arrondie à un entier. Défaut : 131	début à 131

## Opération

INVERSE inverse la totalité ou une portion de l'affichage graphique. les points blancs passent à noir, et les points noirs passent à blanc.

Le contenu de la mémoire d'écran (telle que renvoyée par DISP\$) n'est pas modifié.

Les paramètres optionnels précisent les première et dernière colonnes de l'inversion.

## Références

JPC 19 (page 25) routine d'inversion de l'afficheur graphique en Forth par Jean-Pierre Bondu.

JPC 24 (page 37) première version par Jean-Jacques Moreau.

JPC 25 (page 59) programme en Forth renvoyant l'adresse d'une colonne graphique par Jean-Pierre Bondu.

*A paraître* : deuxième version par Pierre David et Janick Taillandier.

*Internal Design Specification* Volume I, chapitre 3.2.1.

## Mots-clés associés

GDISP\$, GDISP

## Auteurs

Pierre David, Jean-Jacques Moreau, Janick Taillandier



## KA (suite)

Maintenant que votre fichier contient trois adresses (appuyez sur la touche [f][CAT] et maintenez-la enfoncée pour vérifier), nous allons voir comment nous déplacer dans les adresses.

Essayez les touches de curseur [↑] et [↓] pour vous déplacer dans la fiche. Il est normal que vous voyez des lignes vides. Ce sont celles que vous avez rentrées plus haut. Vous pouvez vous rendre directement en haut ou en bas en préfixant les touches de curseur par [g] (touche préfixe bleue).

Si une ligne dépasse 22 caractères, vous pouvez la voir dans sa totalité en faisant dérouler l'affichage avec les touches [←] et [→] (ou les mêmes préfixées par la touche [g]).

Pour se déplacer de fiche en fiche, vous utiliserez les touches [(] et [)] (parenthèses), ou les mêmes préfixées par la touche bleue [g] pour aller à une extrémité du fichier. Remarquez que les fiches sont triées par ordre alphabétique.

Placez vous maintenant au début du fichier (touche [g][ ]). Nous allons chercher l'adresse de Jacques Selere. Appuyez sur la touche [S], rentrez [E][L][E][R][E] et appuyez sur [ENDLINE].

Vous êtes maintenant positionné sur la fiche de Jacques Selere. Remarquez que la recherche ne tient pas compte des différences entre majuscules et minuscules.

Continuons dans la recherche. Vous désirez vous rendre sur la fiche dont le nom commence par « BR ». Appuyez sur la touche [B], entrez les caractères R et . (point) et appuyez sur [ENDLINE]. C'est la recherche abrégée.

Supposons pour continuer que vous vouliez supprimer la fiche de Jacques Selere. Rendez-vous sur cette fiche, et appuyez sur [f][DELETE]. Confirmation vous est demandée, et vous appuyez sur la touche [Y] pour enregistrer la suppression.

Enfin, nous allons modifier une fiche. Allez sur la fiche de Harry Caux et appuyez sur la touche [f][EDIT]. Vous pouvez maintenant modifier chaque champ de la fiche. Vous sortez de ce mode en enregistrant les modifications en appuyant sur [RUN] ou deux fois sur [ENDLINE] sur la dernière ligne.

### Les fichiers d'adresses

KA est un gestionnaire interactif de carnet d'adresses. Les adresses sont contenues dans un fichier, de type ADRS. Le fichier d'adresses par défaut se nomme ADRS, et est cherché automatiquement lorsque KA est exécuté sans paramètre.

Lorsque le fichier d'adresses indiqué est sur un support externe (tel qu'unité de disquette, par exemple), KA le copie d'abord en mémoire, puis cherche à lire les adresses qui s'y trouvent.

### Les adresses

Un fichier d'adresses contient des fiches, chacune contenant une adresse et éventuellement d'autres informations. Une fiche est identifiée par le nom qui lui est associé.

Il n'y a pas d'autre limite au nombre de fiches gérées par KA que la mémoire disponible dans votre HP-71B.

Chaque fiche contient 8 lignes, organisée comme suit :

- le nom et le prénom,
- le numéro de téléphone,
- 4 lignes pour mémoriser l'adresse,
- une ligne pour noter des informations, et
- une ligne pour stocker un critère, que vous pouvez utiliser pour vos programmes personnels.

La première ligne contient le nom et le prénom, séparés par un caractère barre oblique (/). S'il n'y est pas, KA l'ajoute automatiquement à la fin de votre ligne.

Chaque ligne peut occuper jusqu'à 90 caractères.

### Fonctionnement de KA

Lorsque vous entrez dans KA, vous vous trouvez en mode consultation de fichier. Dans ce mode, vous pouvez consulter toutes vos adresses.

En cas de difficulté, appuyez sur la touche [ATTN]. Cela a pour effet de sortir de KA lorsqu'on est en mode *consultation*, ou de revenir au mode *consultation* dans les autres modes.

#### Le mode consultation

Dans ce mode, vous pouvez vous déplacer de fiche en fiche avec les touches [(] et [)]. Les touches [g][ (] et [g][ )] servent à se rendre directement au début et à la fin du fichier respectivement.

Vous pouvez également vous déplacer dans une fiche avec les touches [↑] et [↓]. Vous pouvez également utiliser les touches [g][↑] et [g][↓] pour vous rendre sur le premier champ ou sur le dernier respectivement.

Les touches [0] à [7] permettent en outre de se rendre directement sur une ligne sans passer par les lignes intermédiaires. La touche [f][CAT] (maintenue enfoncée) affiche le nombre de fiches introduites dans le fichier, et une estimation du

nombre de fiches que l'on peut encore rentrer. Attention : ce dernier chiffre est purement indicatif. C'est une estimation basée sur la taille moyenne d'une fiche et sur la place restant dans la mémoire ou le port dans lequel réside le fichier d'adresses.

La touche [f] [DELETE] demande l'effacement d'une fiche. Un message vous demande confirmation. Si vous appuyez sur la touche [Y] (yes), la fiche sera détruite. Sinon, la fiche ne sera pas détruite.

## Mode édition de fiche

Le mode *consultation* ne permettait que la visualisation de fiches déjà introduites. Le mode *édition* permet de modifier une fiche (touche [f] [EDIT]) ou d'en introduire une nouvelle (touche [f] [INPUT]).

En mode *consultation*, vous rentrez en mode *édition* par la touche :

- [f] [EDIT] (la fiche éditée est la fiche courante), ou
- [f] [INPUT] (la fiche éditée est vide).

Vous pouvez dès lors modifier ou saisir la fiche éditée. Les touches disponibles sont :

- [↑], [↓], [g][↑] ou [g][↓] : déplacement sur une autre ligne sans valider la ligne courante,
- [ENDLINE] : validation d'une ligne et passage à la ligne suivante,
- [ATTN] : efface la ligne courante,
- deux fois sur [ATTN] : sort du mode *édition* sans enregistrer aucune modification,
- deux fois [ENDLINE] sur la dernière ligne, ou [RUN] : validation de la fiche et insertion dans le fichier.

Une fois validée, la fiche est insérée automatiquement dans le fichier dans l'ordre alphabétique.

## Mode recherche

Le mode *recherche* permet de rechercher un nom dans tout le fichier. Depuis le mode *consultation*, il suffit d'appuyer sur une des touches [A] à [Z] pour introduire le premier caractère du nom. Vous pouvez ensuite taper le reste du nom. Lorsque vous appuyez sur [ENDLINE], le nom (et éventuellement le prénom) sont recherchés dans tout le fichier. Un message d'erreur (Not Found) vous indique lorsque la recherche a été vaine.

Le mode *recherche* permet trois sortes de recherches :

- La recherche sur le nom :

Lorsque vous rentrez un nom et pas de prénom (c'est à dire sans caractère /), le nom est recherché dans le fichier, de la fiche courante à la fin, puis du début jusqu'à la fiche courante. Le premier nom trouvé devient la fiche courante.

- La recherche sur le nom et le prénom :

La fiche est cherchée suivant le même parcours que précédemment. Si une fiche porte le même nom, mais pas le même prénom, elle ne sera pas trouvée.

- La recherche abrégée :

C'est le type de recherche le plus souvent utilisé. Il suffit de taper le début du nom, puis un point. KA va alors rechercher la première fiche dont le nom commence par les caractères rentrés. Si aucune fiche ne répond au critère, la suivante est éditée.

Il n'y a pas de distinction entre lettres majuscules et minuscules au niveau de la recherche.

La recherche abrégée est la plus simple et la plus rapide pour trouver une fiche. C'est la plus largement utilisée dans l'utilisation courante de KA.

## Mot de passe

Chaque fichier d'adresses peut contenir un mot de passe. Dès lors, chaque appel de KA demandera votre mot de passe, et vous devrez le fournir également aux fonctions programmables de gestion du carnet d'adresses (voir ADCREATE).

Vous pouvez modifier le mot de passe (et c'est la seule méthode pour le faire) associé à un fichier en appuyant sur la touche [f] [KEY] en mode *consultation*. Un nouveau mot de passe vous est demandé.

- si vous appuyez sur [ATTN], le mot de passe n'est pas changé,
- si vous ne rentrez rien, et appuyez sur [ENDLINE], votre mot de passe est supprimé, et enfin
- si vous rentrez un mot de passe (jusqu'à 8 caractères), et le validez avec [ENDLINE], celui-ci est enregistré.

Si vous utilisez un interface vidéo HP-IL, le mot de passe ne sera pas affiché sur l'écran lorsque vous l'introduirez.

## Références

Mode d'emploi du programme AGENDA pour HP-75.

# KA (suite)

## Mots-clés associés

ADCREATE, ADDELETE, ADFIND, ADGET, ADPUT, ADSIZE

## Auteur

Pierre David

KEYWAIT\$ attend une pression de touche et renvoie la chaîne représentant cette touche.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

KEYWAIT\$

## Exemples

A\$=KEYWAIT\$

Attend une pression de touche et place dans A\$ le code de la touche.

```
10 LOOP
20 DISP KEYWAIT$
30 END LOOP
```

Fait une boucle et affiche les touches pressées au clavier.

```
10 SELECT KEYWAIT$
20 CASE "A" TO "Z"
30 DISP "LETTRE"
40 CASE "#46"
50 DISP "TOUCHE [RUN]"
60 CASE ELSE
70 DISP "AUTRE"
80 END SELECT
```

Attend une pression de touche, et affiche ensuite "LETTRE" si la touche appuyée était une lettre, "[RUN]" si c'était la touche [RUN], ou "AUTRE" si ce n'était pas une de celles-ci.

## Opération

KEYWAIT\$ place le HP-71 dans un état de faible consommation électrique et attend une pression de touche.

La chaîne représentant la touche est similaire à la chaîne renvoyée par la fonction standard KEY\$.

L'utilisation de KEYWAIT\$ permet d'éviter la programmation de boucles pour attendre un caractère telles que :

```
10 K$=KEY$ @ IF K$="" THEN 10
```

## Références

*JPC 20* (page 50) désassemblage de KEYWAIT\$ par Pierre David et Michel Martinet.

## Mots-clés associés

ATTN, FKEY, KEY\$, PUT

## Auteur

Hewlett-Packard

# LABEL

LABEL trace des caractères alphanumériques à partir de la position courante de la plume.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

LABEL *chaîne alphanumérique* [ ; ]

## Exemples

LABEL "EXEMPLE DE TRACE";

Ecrit la chaîne "EXEMPLE DE TRACE" dans la zone de traçage, et laisse la plume après la dernière lettre.

LABEL A\$&STR\$(I)

Ecrit la chaîne spécifiée dans la zone de traçage, et place la plume au début de la ligne suivante.

## Paramètre d'entrée

Élément	Description	Restriction
Chaîne alphanumérique	Expression alphanumérique.	Aucune.

## Opération

Les titres et labels peuvent être écrits en tout endroit de la zone de tracé.

LABEL les écrit à partir de la position courante de la plume, en utilisant le type de ligne numéro 1 (ligne pleine).

Une séquence de fin de ligne est envoyée après le dernier caractère imprimé. Elle est supprimée si un point-virgule (;) est ajouté après la chaîne.

### Attributs des labels :

L'ordre CSIZE détermine la hauteur et le rapport d'aspect des caractères.

L'ordre LORG détermine la position du label, en accord avec la position courante de la plume lors de l'exécution de LABEL.

L'ordre LDIR détermine la direction dans laquelle sont tracés les labels.

Le jeu de caractères est le jeu de caractères Roman8.

## Mots-clés associés

CSIZE, LDIR, LORG, ENDLINE

## Auteur

Pierre David

LDIR (Label DIRection) spécifie l'angle des labels par rapport à l'axe des  $x$ .

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

LDIR *angle*

## Exemples

LDIR 0 Les labels seront tracés horizontalement (valeur par défaut défaut).

LDIR ASIN(.5) L'angle entre la direction des labels et l'axe des  $x$  est de 30 degrés.

## Paramètre d'entrée

Élément	Description	Restriction
angle	Expression numérique, interprétée suivant le mode trigonométrique en vigueur.	0 à 360 degrés, ou 0 à $2*PI$

## Opération

L'ordre LDIR définit l'angle de rotation pour l'impression des labels. L'angle est interprété dans le mode trigonométrique courant (DEGREES ou RADIANS).

Cet angle mesure la rotation entre l'axe des  $x$  (horizontal) et la direction des labels.

L'exécution de GINIT initialise cet angle à 0.

## Mots-clés associés

DEGREES, RADIANS, LABEL

## Auteur

Pierre David

# LEAVE

LEAVE sort proprement d'une boucle structurée de type WHILE, REPEAT ou LOOP.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

LEAVE  
LEAVE *profondeur*

## Exemples

```
IF I=INF THEN LEAVE
```

Sort de la boucle si la valeur de la variable I est infinie.

```
10 WHILE X<0  
20   WHILE Y<0  
30     Z=FNN(X,Y)  
40     IF Z<0 THEN LEAVE 2  
50   END WHILE  
60 END WHILE
```

Si la valeur de la variable Z devient négative, le test en ligne 40 provoque la sortie prématurée des deux boucles WHILE englobantes.

## Paramètre d'entrée

Élément	Description	Restriction
profondeur	Expression numérique arrondie à un entier. Défaut : 1	1 au nombre deboucles englobantes.

## Opération

LEAVE sort prématurément d'une boucle WHILE ... END WHILE ou REPEAT ... UNTIL.

Pour les boucles de type LOOP ... END LOOP, LEAVE représente la seule manière de sortir.

Si le paramètre optionnel *profondeur* est fourni, LEAVE sortira du nombre de boucles englobantes spécifié.

## Références

JPC 31 (page 38) première version par Janick Taillandier.

JPC 52 (page 10) deuxième version par Pierre David et Janick Taillandier.

## Mots-clés associés

EXIT, LOOP ... END LOOP, REPEAT ... UNTIL, WHILE ... END WHILE

## Auteurs

Pierre David, Janick Taillandier

LEX désactive ou active un fichier Lex.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
LEX <i>fichier</i> OFF LEX <i>fichier</i> ON	

## Exemples

LEX STRINGLX OFF

Désactive le fichier LEX STRINGLX. Les fonctions de ce Lex ne sont plus disponibles, et le type du fichier est changé en D-LEX.

LEX STRINGLX ON

Active le fichier D-LEX STRINGLX. Les fonctions de ce Lex redeviennent disponibles.

## Paramètre d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Nom de fichier avec un spécificateur de périphérique optionnel.

## Opération

### La désactivation des fichiers Lex

La vitesse de traitement du HP-71 est liée au nombre de fichiers Lex en mémoire. Plus il y a de fichiers, moins le HP-71 réagit rapidement. Il vaut mieux avoir un seul gros fichier Lex regroupant 90 fonctions que 9 petits regroupant 10 fonctions chacun. Si votre HP-71 comprend une multitude de petits fichiers Lex, vous souhaitez peut être en désactiver quelques uns, quitte à les réactiver quand vous en aurez besoin.

D'autre part, si des fichiers Lex comprennent deux mots-clefs ayant les mêmes numéros d'identification (ID) et de token, vous pouvez utiliser l'un ou l'autre de ces mots-clefs en désactivant le Lex que vous ne désirez pas utiliser.

Les fichiers en mémoire morte ou en module Eprom ne sont pas désactivables.

### L'ordre LEX ON/OFF

LEX *fichier* OFF change le type de *fichier* de LEX en type D-LEX, de telle sorte qu'il ne soit plus actif. Ses fonctions ne sont plus utilisables, jusqu'à ce qu'un ordre LEX *fichier* ON soit exécuté sur ce même fichier.

### Les fichiers D-LEX

Les fichiers désactivés apparaissent sous le type D-LEX lors d'un CAT ou CAT\$ tant que JPC Rom est connecté sur votre HP-71. Ces fichiers peuvent être copiés sur support de mémoire de masse, mais ils peuvent être recopiés dans la mémoire du HP-71 seulement lorsque JPC Rom est dans votre HP-71.

## Références

JPC 24 (page 30) première version par Michel Martinet.

A paraître : nouvelle version par Pierre David et Janick Taillandier.

Les anciennes appellations de LEX étaient ENABLE et DISABLE

## Mots-clés associés

EDIT, MERGE

# LEX (suite)

## Auteurs

Pierre David, Michel Martinet, Janick Taillandier

LINETYPE sélectionne le type et la longueur du motif pour le tracé de lignes, axes et rectangles.

<input checked="" type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

LINETYPE *type* [, *longueur du motif*]

## Exemples

LINETYPE 1

Définit la ligne pleine comme type de ligne

LINETYPE 6, 48

Définit le motif numéro 6, ainsi qu'une longueur de motif de 48 unités graphiques.

## Paramètres d'entrée

Élément	Description	Restriction
type	Expression numérique arrondie à un nombre entier.	0 à +32767
longueur du motif	Expression numérique arrondie à un nombre entier. Défaut : 16	0 à +32767

## Opération

LINETYPE choisit un type de ligne parmi 8 suivant le paramètre *type*. Le numéro 0 donne un trait plein.

La longueur du motif est exprimée en unités graphiques pour la totalité du motif. La valeur par défaut est 16 unités graphiques.

## Mots-clés associés

DRAW, IDRAW, BOX, FRAME, XAXIS, YAXIS

## Auteur

Pierre David

# LOOP ... END LOOP

LOOP ... END LOOP définit une boucle sans fin.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

```
LOOP
  segment de programme
END LOOP
```

## Exemple

```
10 INPUT "Nombre entier : ";X
20 LOOP
30   DISP X;X*X
40   X=X+1
50 END LOOP
```

Définit une boucle sans fin affichant la suite des nombres entiers et de leur carré. La seule manière d'arrêter cette boucle est l'appui sur la touche [ATTN].

## Paramètre d'entrée

Élément	Description	Restriction
segment de programme	Tout nombre de lignes de programme contiguës.	Aucune.

## Opération

La structure LOOP ... END LOOP permet de faire des boucles infinies, avec des sorties par LEAVE.

Le segment de programme à répéter commence après le mot-clef LOOP et se termine avant le mot-clef END LOOP. Atteindre le mot-clef END LOOP revient à faire un branchement au premier ordre suivant LOOP.

Le segment de programme peut contenir un nombre quelconque de LEAVE. La seule restriction est que ces LEAVE ne soient pas imbriqués dans des structures emboîtées.

Le segment de programme peut lui-même contenir des structures de type boucle ou choix. L'emboîtement doit être correct, sinon l'erreur : JPC ERR:Structure Mismatch est renvoyée.

## Références

JPC 52 (page 10) première version par Pierre David et Janick Taillandier.

Basic 5.0 pour HP 9000 séries 200/300.

## Mots-clés associés

ATTN, LEAVE, FOR ... NEXT, WHILE ... END WHILE, REPEAT ... UNTIL

## Auteurs

Pierre David, Janick Taillandier

LORG (Label ORiGin) définit la position des labels par rapport à la position courante de la plume.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

LORG <i>position</i>
----------------------

## Paramètre d'entrée

Élément	Description	Restriction
position	Expression numérique arrondie à un entier.	-32767 à +32767

## Opération

LORG définit la position de l'origine du label, qui détermine l'emplacement des labels par rapport à la position courante de la plume. Le paramètre de position peut prendre toute valeur entre 1 et 9 et est interprété conformément à l'illustration ci-après (le centre des croix représente la position courante de la plume).

## Mots-clés associés

LABEL, MOVE

## Auteur

Pierre David

# MAP

MAP applique une fonction de conversion aux caractères d'un fichier texte.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

```
MAP fichier , chaîne1 , chaîne2
MAP fichier , chaîne1 , chaîne2 , début
MAP fichier , chaîne1 , chaîne2 , début , fin
MAP # canal , chaîne1 , chaîne2
MAP # canal , chaîne1 , chaîne2 , début
MAP # canal , chaîne1 , chaîne2 , début , fin
```

## Exemple

MAP TOTO, "éï", "ei", 10, 50

Remplace, dans le fichier TOTO, les caractères "é" et "ï" par leur équivalent non accentué, de la ligne 10 à la ligne 50.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Nom de fichier avec spécificateur d'unité en option.
canal	Expression numérique arrondie à un entier.	1 à 255.
chaîne1, chaîne2	Expressions alphanumériques.	Les deux chaînes doivent avoir la même longueur.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575.
fin	Expression numérique arrondie à un entier. Défaut : dernière ligne du fichier.	0 à 1048575.

## Opération

MAP parcourt le fichier spécifié par son nom (*fichier*) ou par le numéro de canal qui lui est associé (*canal*).

Chaque caractère est testé pour savoir si il appartient à l'ensemble des caractères de *chaîne1*. Si c'est le cas, le caractère de *chaîne2* correspondant remplace le caractère d'origine.

Si le même caractère existe plusieurs fois dans *chaîne2*, seule la première occurrence est utilisée pour trouver le caractère correspondant de *chaîne3*.

## Références

STaK (Journal du Club Finlandais) Novembre 1986. Première version par Tapani Tarvainen.

JPC 46 (page 18) traduction de l'article de Tapani Tarvainen dans JPC.

## Mots-clés associés

ASSIGN #, MAP\$, ROMAN

## Auteur

Tapani Tarvainen

MAP\$ applique une fonction de conversion aux caractères d'une chaîne alphanumérique.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

MAP\$ ( chaîne1 , chaîne2 , chaîne3 )

## Exemple

MAP\$ ("chaîné", "éi", "ei")

Retourne "chaine", c'est à dire le paramètre initial sans accents.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne1	Expression alphanumérique.	Aucune.
chaîne2, chaîne3	Expressions alphanumériques.	Les deux chaînes doivent avoir la même longueur.

## Opération

MAP\$ parcourt *chaîne1* caractère par caractère ; si celui-ci se trouve dans *chaîne2*, il est remplacé par le caractère de *chaîne3* correspondant.

Si le même caractère existe plusieurs fois dans *chaîne2*, seule la première occurrence est utilisée pour trouver le caractère correspondant de *chaîne3*.

## Références

*STaK* (Journal du Club Finlandais) Novembre 1986. Première version par Tapani Tarvainen.

*JPC 46* (page 18) traduction de l'article de Tapani Tarvainen dans *JPC*.

## Mots-clés associés

MAP, REPLACE\$, ROMAN

## Auteur

Tapani Tarvainen

# MARGIN

MARGIN valide un signal sonore lorsque le curseur arrive à la position spécifiée.

<input checked="" type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

MARGIN
MARGIN <i>position</i>

## Exemples

MARGIN 80

Le curseur émettra un signal sonore quand il arrivera à la colonne 80.

MARGIN

Désactive le précédent MARGIN actif.

## Paramètre d'entrée

Élément	Description	Restriction
position	Expression numérique arrondie à un entier. Défaut : 0	0 à 96.

## Opération

MARGIN mémorise la position du curseur demandée, puis émettra un signal sonore à chaque fois que le curseur atteindra cette position, en supposant que le signal sonore soit actif (BEEP ON) et que JPC Rom soit dans votre HP-71.

MARGIN agit en mode d'introduction normale, aussi bien que pendant un INPUT, LINPUT ou FINPUT en mode programme, ou également en mode FORTH avec les modules Forth Assembleur ou Translator Pac.

MARGIN utilisé sans paramètre ou MARGIN 0 inhibent cette possibilité, et récupèrent l'espace alloué à la mémorisation de la position du curseur (4,5 octets).

## Références

JPC 26 (page 33) première version par Michel Martinet.

A paraître : modification pour utilisation de ressources attribuées par HP, par Pierre David et Janick Taillandier.

MARGIN pour HP-75.

## Mots-clés associés

BEEP, FINPUT, FORTH, INPUT, LINPUT

## Auteurs

Pierre David, Michel Martinet, Janick Taillandier

MAXD (MAXimum Directory size) retourne le nombre d'entrées maximum du répertoire d'un support magnétique.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input type="checkbox"/> Opération d'unité               |

MAXD ( *spécificateur d'unité* )

## Exemples

A=MAXD(" .DISK")

Place dans la variable A le nombre d'entrées maximum dans le répertoire du disque repéré par le label de volume ".DISK".

DISP MAXD(A\$)

Affiche le nombre d'entrées disponibles sur le support désigné par le contenu de la variable A\$.

N=MAXD(D)

Place dans la variable N le nombre d'entrées sur le disque situé à la position D dans la boucle.

## Paramètre d'entrée

Élément	Description	Restriction
spécificateur d'unité	Voir les définitions standard HP-IL.	Ne doit pas être une chaîne sans guillemets.

## Opération

### Les mémoires de masse :

Les périphériques reconnus comme mémoire de masse par le HP-71 utilisent tous le protocole *Filbert*. Ce sont, à l'heure actuelle, l'unité de cassettes numériques HP-82161 et l'unité de disquettes magnétiques HP-9114. En règle générale, ce sont tous les périphériques répondant au spécificateur : TAPE.

L'unité de mesure en ce qui concerne les mémoires de masse est le *secteur*. Un *secteur* est composé de 256 octets, et est l'élément de base des transferts entre contrôleur (votre HP-71) et mémoire de masse.

L'unité de cassettes numériques HP-82161 totalise 512 secteurs sur une cassette, soit 128 Ko. Une disquette double face pour le HP-9114 possède 2464 secteurs, soit 616 Ko.

Les mémoires de masse sont organisées en trois parties :

- La partie *système*. Le contrôleur HP-IL (le HP-71) l'utilise pour stocker des informations telles que date et heure d'initialisation, label de volume, taille du médium, etc. Cette partie occupe de manière fixe les secteurs 0 et 1, soit 512 octets.
- La partie *répertoire*. Cette partie contient les noms et des informations utiles sur les fichiers, telles que date et heure de création, taille, emplacement sur le support, type, etc. 32 octets sont nécessaires pour coder une entrée de fichier. Le répertoire commence normalement au secteur 2.
- La partie *fichiers*. C'est là que sont stockés les fichiers proprement dits.

La taille du répertoire est fixée à l'initialisation, par l'ordre INITIALIZE. Le paramètre est arrondi au multiple de 8 supérieur, de manière à occuper un nombre entier de secteurs. Il est stocké dans la partie *système* du support.

Par défaut, INITIALIZE utilise une taille de répertoire égale à  $1/32^{\text{ème}}$  de la taille du support (en secteurs). Le nombre d'entrées est alors :

$\text{nombre d'entrées} = \text{taille du support en octets} / 1024$

### La fonction MAXD :

MAXD renvoie le nombre maximum d'entrées dans le répertoire du support indiqué.

Ce nombre est toujours multiple de 8. Pour avoir la place occupée par le répertoire en :

secteurs : faire  $\text{MAXD} / 8$

octets : faire  $\text{MAXD} * 32$

### Interruption par [ATTN] :

## **MAXD (suite)**

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur HPIL ERR: Abort ed. Il peut être nécessaire d'exécuter RESTORE IO pour réactiver le système HP-IL.

### **Références**

*JPC 30* (page 44) première version par Michel Martinet.

Manuel d'utilisation du lecteur de cassettes numériques HP-82161A.

Manuel du module HP-IL pour le HP-71, chapitre 3 et Annexe D.

### **Mots-clés associés**

INITIALIZE, MEMD, MAXM, RREC\$, WREC

### **Auteur**

Michel Martinet

MAXM (MAXimum Medium size) renvoie la taille maximum du support en octets.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input checked="" type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input type="checkbox"/> Opération d'unité

MAXM ( <i>spécificateur d'unité</i> )
---------------------------------------

## Exemples

A=MAXM( "%16" )

Renvoie la taille maximum du support en octets.

DISP MAXM(1)

Renvoie la taille maximum du support situé dans le lecteur à l'adresse 1 sur la boucleu HP-IL.

## Paramètre d'entrée

Élément	Description	Restriction
spécificateur d'unité	Voir les définitions standard HP-IL.	Ne doit pas être une chaîne sans guillemets.

## Opération

MAXM renvoie la taille du support désigné par le *spécificateur d'unité*. Cette taille comprend la place occupée par le répertoire, ainsi que les deux enregistrements utilisés par le système.

Pour avoir la taille maximum réservée aux données, il faut faire :

MAXM(D) - 512 - MAXD(D) \* 32

Pour plus de renseignements, voir MAXD, et le manuel du module HP-IL.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur HPIL ERR: Abort ed. Il peut être nécessaire d'exécuter RESTORE IO pour réactiver le système HP-IL.

## Références

JPC 30 (page 44) première version par Michel Martinet.

Manuel d'utilisation du lecteur de cassettes numériques HP-82161A.

Manuel du module HP-IL pour le HP-71, chapitre 3 et Annexe D.

## Mots-clés associés

MAXD, MEMM, RREC\$, WREC

## Auteur

Michel Martinet

# MDY

MDY (Month Day Year) valide l'introduction des dates au format numérique *mm.jjaaaa*.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
MDY	

## Exemple

```
IF K$="N" THEN MDY
```

Si la variable K\$ vaut "N", passer en format de date américain.

## Opération

Dans le mode sélectionné par DMY, les dates utilisées dans les fonctions de calcul de dates de JPC Rom peuvent être introduites sous la forme *mm.jjaaaa* (format numérique).

Le format MDY est le format par défaut après une initialisation de la mémoire du HP-71 (INIT :3).

Le format alphanumérique est, quant à lui, toujours utilisable. Il n'est pas affecté par le mode DMY ou MDY.

Pour plus de détails sur les formats d'introduction des dates, consulter la fonction DATESTR\$.

## Références

*JPC 28* (page 40) première version par Laurent Istria.

*JPC 49* (page 24) deuxième version par Pierre David et Janick Taillandier.

## Mots-clés associés

DATESTR\$, DATEADD, DDAYS, DMY, DOW, DOW\$

## Auteurs

Pierre David, Laurent Istria, Janick Taillandier

MEMD (MEMory in Directory) renvoie le nombre d'entrées disponibles dans le répertoire du support indiqué pour de nouveaux fichiers.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input type="checkbox"/> Opération d'unité               |

MEMD ( *spécificateur d'unité* )

## Exemples

A=MEMD(" : TAPE")

Renvoie le nombre d'entrées disponibles dans le répertoire.

DISP MEMD(B+N)

Renvoie le nombre d'entrées disponibles pour de nouveaux fichiers sur le support situé à l'adresse HP-IL B+N.

## Paramètre d'entrée

Élément	Description	Restriction
spécificateur d'unité	Voir les définitions standard HP-IL.	Ne doit pas être une chaîne sans guillemets.

## Opération

### Les fichiers effacés :

Lorsque des fichiers sont effacés du support par l'ordre PURGE, les entrées correspondantes dans le répertoire sont rendues disponibles. Cela conduit à des «trous» dans le répertoire, normalement invisible pour l'utilisateur.

Cependant, ces trous peuvent devenir prépondérants. Dans ce cas, un PACKDIR est quelquefois nécessaire pour tasser le répertoire et enlever les trous.

### La fonction MEMD :

La fonction MEMD retourne le nombre d'entrées disponibles dans le répertoire. Ceci compte les entrées correspondant aux fichiers effacés par PURGE et les entrées inutilisées s'il y en a.

MEMD compte comme disponibles les entrées correspondant aux fichiers effacés : MEMD agit comme si il y avait eû un PACKDIR.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur HPIL ERR: Aborted. Il peut être nécessaire d'exécuter RESTORE IO pour réactiver le système HP-IL.

## Références

JPC 30 (page 44) première version par Michel Martinet.

Manuel d'utilisation du lecteur de cassettes numériques HP-82161A.

Manuel du module HP-IL pour le HP-71, chapitre 3 et Annexe D.

## Mots-clés associés

MAXD, MEMM, PACKDIR, RREC\$, WREC

## Auteur

Michel Martinet

# MEMM

MEMM (MEMory on Medium) renvoie la taille disponible pour stocker des fichiers sur le support spécifié.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input checked="" type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input type="checkbox"/> Opération d'unité

MEMM ( <i>spécificateur d'unité</i> )
---------------------------------------

## Exemple

A=MEMM(" :HP9114")

Renvoie la place disponible sur la première unité de disques HP-9114 rencontrée.

## Paramètre d'entrée

Élément	Description	Restriction
spécificateur d'unité	Voir les définitions standard HP-IL.	Ne doit pas être une chaîne sans guillemets.

## Opération

### Les fichiers effacés :

Lorsque des fichiers sont effacés du support par l'ordre PURGE, la place correspondante dans l'espace de stockage des fichiers est rendue disponible. Cela conduit à des espaces vides sur le support, normalement invisible pour l'utilisateur.

Cependant, ces trous peuvent devenir prépondérants. Dans ce cas, un PACK est quelquefois nécessaire pour tasser le répertoire et l'espace de stockage des fichiers, et enlever les trous.

### La fonction MEMM :

la fonction MEMM retourne la place disponible dans l'espace de stockage des fichiers pour de nouveaux fichiers, en comptant également les fichiers effacés et l'espace inutilisé.

MEMM compte comme disponible la place rendue disponible par les fichiers effacés : MEMM agit comme si il y avait eu un PACK.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur HPIL ERR:Abort ed. Il peut être nécessaire d'exécuter RESTORE IO pour réactiver le système HP-IL.

## Références

JPC 30 (page 44) première version par Michel Martinet.

Manuel d'utilisation du lecteur de cassettes numériques HP-82161A.

Manuel du module HP-IL pour le HP-71, chapitre 3 et Annexe D.

## Mots-clés associés

MAXM, MEMD, PACK, RREC\$, WREC

## Auteur

Michel Martinet

MENU gère des menus interactifs.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

MENU ( <i>nombre d'éléments</i> )
MENU ( <i>nombre d'éléments</i> , <i>premier élément</i> )

## Exemple

```

10 ATTN OFF @ M=1
20 DATA UN, DEUX, TROIS, QUATRE
30 RESTORE 20
40 M=MENU(4,M) @ ON M GOTO 60,70,80,90
50 ATTN ON @ BEEP @ END
60 DISP "ONE" @ GOTO 40
70 DISP "TWO" @ GOTO 40
80 DISP "THREE" @ GOTO 40
90 DISP "FOUR" @ GOTO 40

```

Propose un choix de 4 éléments visualisable par les touches de curseur vertical. L'utilisateur valide avec [ENDLINE], ou interrompt avec [ATTN] et saute à la ligne 40.

## Paramètres d'entrée

Élément	Description	Restriction
nombre d'éléments	Expression numérique arrondie à un entier.	0 à 1048575.
premier élément	Expression numérique arrondie à un entier. Défaut : 1	0 à 1048575.

## Opération

MENU est destiné à la gestion des menus interactifs par programme.

MENU affiche les éléments contenus dans les lignes DATA repérés par le pointeur de DATA courant.

L'utilisateur utilise alors les touches de curseur vertical, soit [↑], [↓], [g][↑] et [g][↓], pour faire défiler les éléments. Il valide son choix en appuyant sur la touche [ENDLINE]. Ceci renvoie le numéro d'ordre de l'élément correspondant. Le premier élément porte le numéro 1, et le dernier porte le numéro *nombre d'éléments*.

Une pression sur la touche [ATTN] annule l'opération MENU et passe à la ligne suivante du programme (et non à l'ordre suivant). Il est ainsi possible de gérer les interruptions de l'utilisateur.

Si la valeur *premier élément* est spécifiée, elle représente le numéro d'ordre de la première donnée à afficher. Les touches [↑] et [g][↑] permettent bien sûr de revenir aux précédents éléments.

## Références

JPC 26 (page 34) première version par Jean-Jacques Dhénin.

## Mots-clés associés

DATA, FINPUT, READ, RESTORE

## Auteur

Jean-Jacques Dhénin

# MERGE

MERGE étend le mot-clef standard pour le chaînage des fichiers Lex. MERGE n'est pas programmable.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

MERGE <i>fichier</i> MERGE <i>fichier</i> , <i>ligne début</i> MERGE <i>fichier</i> , <i>ligne début</i> , <i>ligne fin</i>
---

## Exemple

EDIT STRINGLX @ MERGE KEYWAIT

Chaîne le fichier KEYWAIT dans le fichier STRINGLX.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Nom de fichier avec un spécificateur de périphérique optionnel.
ligne début	Constante entière. Défaut : Première ligne de programme.	1 à 9999 (sans signification si Lex).
ligne fin	Constante entière. Défaut : Ligne de début.	Ligne de début à 9999 (sans signification si Lex).

## Opération

### Le chaînage de fichiers Lex :

La technique du chaînage permet de réunir plusieurs Lex en un seul fichier.

Le principal avantage du chaînage est qu'il permet de réunir en un seul fichier tous les mots-clefs et extensions du système d'exploitation requis par un logiciel. Ceci simplifie grandement le programme principal du logiciel, celui-ci n'ayant à tester la présence que d'un seul fichier Lex, au lieu de plusieurs. Le temps de chargement de ce fichier Lex combiné est réduit de manière significative par rapport au temps nécessaire pour charger les fichiers un par un.

Chaîner des fichiers Lex existant est plus facile qu'écrire un nouveau Lex spécialisé disposant de toutes les caractéristiques nécessaires. Ceci contribue également à préserver les ressources du HP-71, puisque vous pouvez chaîner des fichiers Lex d'ID différents ou des fonctions de tokens non consécutifs. Enfin, ce chaînage réduit l'encombrement de 18,5 octets pour chaque fichier chaîné, puisque les fichiers Lex partagent le même en-tête de 18,5 octets.

L'intérêt du chaînage de fichiers Lex est le même que la réunion des plusieurs sous programmes Basic en un simple fichier : facilité, et simplification de la gestion de la sauvegarde sur support.

### Mode d'emploi :

Pour réunir les fichiers F1, F2 et F3 dans le fichier F1, il faut procéder comme suit :

```
EDIT F1
MERGE F2
MERGE F3
EDIT
```

Le EDIT F1 rend le fichier Lex F1 *fichier de travail courant*. Le MERGE F2 réunit le fichier F2 dans le fichier F1. Le MERGE F3 réunit le fichier F3 dans le fichier F1 (qui est maintenant F1 + F2).

Le EDIT final remet le `workfile` comme fichier courant.

Il ne reste plus qu'à purger éventuellement F2 et F3 qui restent en mémoire, comme après un MERGE normal de programmes Basic ou de fichiers Keys.

Si vous désirez restaurer le fichier courant, vous ferez :

```
A$=CAT$(0)[1,8]
EDIT F1
```

MERGE F2  
MERGE F3  
EDIT A\$

## Attention !

Le principe de l'opération est d'autoriser l'édition d'un fichier Lex. Il y a un effet secondaire pervers, qui peut se révéler gênant. Pour cette raison, il ne faut **jamais purger le fichier Lex courant** !

Il semblerait qu'il y ait une bogue quelque part. L'ordre Basic PURGE aurait une attitude étrange face à un fichier courant de type Lex. Il ne remet pas le pointeur de fichier courant dans le `workfile`. Ceci conduit à des CAT bizarres. Un simple EDIT suffit généralement à s'en sortir...

## Références

*JPC 23* (page 47) programme de chaînage de fichier Lex en Basic par Michel Martinet.

*JPC 37* (page 22) chaînage de fichiers Lex en assembleur par Pierre David et Michel Martinet.

## Mots-clés associés

EDIT, MERGE

## Auteurs

Pierre David, Michel Martinet

# MODE

MODE change le jeu de caractères sur l'appareil d'impression courant.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input type="checkbox"/> Opération d'unité
--	--

MODE <i>argument</i>
----------------------

## Exemple

MODE 2

Sélectionne l'impression compressée sur les imprimantes HP.

## Paramètre d'entrée

Élément	Description	Restriction
argument	Expression numérique arrondie à un entier.	0 à 999.

## Opération

MODE change la densité d'impression sur l'appareil désigné par le plus récent ordre `PRINTER IS`.

Sur l'imprimante ThinkJet,

MODE 0 correspond à 80 caractères par ligne,

MODE 1 correspond à 40 caractères par ligne,

MODE 2 correspond à 142 caractères par ligne et

MODE 3 correspond à 71 caractères par ligne.

### Codes envoyés à l'imprimante :

MODE *n* : ESC & k *n* S

## Références

*JPC 26* (page 39) première version par Pierre David.

*JPC 40* (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

## Mots-clés associés

BOLD, PRINTER IS, UNDERLINE, WRAP

## Auteur

Pierre David

MOVE lève la plume et la déplace jusqu'au point spécifié par les coordonnées.

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Ordre</li> <li><input type="checkbox"/> Fonction</li> <li><input type="checkbox"/> Opérateur</li> </ul> | <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Exécution au clavier</li> <li><input type="checkbox"/> Mode CALC</li> <li><input checked="" type="checkbox"/> IF...THEN...ELSE</li> <li><input checked="" type="checkbox"/> Opération d'unité</li> </ul> |
|--|---|

MOVE  $x, y$

## Exemples

MOVE 0, 0

Déplace la plume à l'origine du tracé

MOVE X, Y+L

Déplace la plume au point de coordonnées (x,y)

## Paramètre d'entrée

Élément	Description	Restriction
x, y	Expressions numériques arrondies à des entiers.	-32767 à +32767

## Opération

MOVE déplace la plume sans tracer de trait jusqu'à la position indiquée. Si la position demandée est hors des limites de la zone de traçage, le prochain trait commencera à partir de la frontière de la zone.

## Mots-clés associés

DRAW, IMOVE

## Auteur

Pierre David

# NEXTOP\$

NEXTOP\$ (NEXT OPcode) renvoie l'adresse de la prochaine instruction assembleur.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

NEXTOP\$ ( *adresse hexadécimale* )  
NEXTOP\$ ( *adresse décimale* )

## Exemples

A\$=NEXTOP\$ ("0BD38")

```
10 A$="00000"  
20 LOOP  
30 DISP OPCode$(A$)  
40 A$=NEXTOP$(A$)  
50 END LOOP
```

Place 0BD3C, soit l'adresse de la deuxième instruction de la routine POP1S qui commence à l'adresse 0BD38, dans la variable A\$.  
Ce petit programme réalise un désassemblage automatique à partir de l'adresse 00000. La mnémonique de l'instruction est affichée, puis l'adresse est actualisée.

## Paramètres d'entrée

Élément	Description	Restriction
adresse hexadécimale	Expression alphanumérique contenant des chiffres hexadécimaux.	Jusqu'à 5 chiffres en majuscules ou en minuscules.
adresse décimale	Expression numérique arrondie à un nombre entier.	0 à 1048575.

## Opération

NEXTOP\$ renvoie l'adresse de correspondant à l'instruction suivant celle pointée par l'adresse hexadécimale fournie. Cette fonction, associée à la fonction OPCode\$, permet de désassembler très simplement du code HP-71 en langage machine.

La longueur (en quartets) de l'instruction courante peut être calculée de manière très simple par la soustraction de l'adresse hexadécimale courante et de l'adresse renvoyée par NEXTOP\$ :

$len = HTD(NEXTOP$(adresse courante)) - HTD(adresse courante)$

Attention : si l'adresse que vous spécifiez pointe dans une zone de données (une table ou un en-tête de fichier par exemple), les données seront décodées comme des instructions plutôt que des données. Ce problème peut être résolu par une utilisation du désassembleur interactif SYSEDIT.

## Mots-clés associés

OPCode\$, PEEK\$, SYSEDIT

## Auteurs

Pierre David, Jean-Jacques Dhénin, Janick Taillandier

NLOOP (Number on the LOOP) renvoie le nombre d'appareils sur la boucle HP-IL.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input checked="" type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input type="checkbox"/> Opération d'unité

NLOOP NLOOP ( <i>numéro de boucle</i> )
--

## Exemples

A=NLOOP

```
10 RESTORE IO
20 FOR I=1 TO NLOOP
30  DISP I;DEVID$(I)
40 NEXT I
```

Place dans la variable A le nombre d'appareils sur la boucle HP-IL numéro 1.

Affiche le numéro et le nom de tous les appareils sur la boucle HP-IL.

## Paramètre d'entrée

Élément	Description	Restriction
numéro de boucle	Expression numérique arrondie à un entier. Défaut : 1	1 à 3.

## Opération

NLOOP retourne le nombre de périphériques sur la boucle spécifiée. L'accès à plusieurs boucles HP-IL est possible grâce à l'adaptateur double boucles HP-82402.

Si NLOOP est utilisé en mode *adressage étendu* (flag -22 armé), un nombre de la forme *sseepp* est renvoyé, où :

*ss* est la réponse au message AES,

*ee* est la réponse au message AEP, et

*pp* est la réponse au message AAD.

Note : NLOOP représente le nombre de périphériques connu par le HP-71 à un instant donné. Le HP-71 conserve ce nombre en mémoire, il n'y a donc pas besoin d'envoyer un message sur la boucle. En cas de doute, utilisez RESTORE IO pour mettre à jour l'information.

Après un RESET HPIL, la valeur renvoyée par NLOOP est 0.

Note : en mode *non contrôleur*, le nombre ne peut être mis à jour. Pour cette raison, NLOOP ne peut renvoyer de résultat valide.

## Références

*JPC 30* (page 50) première version par Jean-François Garnier.

*JPC 37* (page 33) deuxième version par Jean-François Garnier.

*HP-IL Module Internal Design Specification*, chapitre 5.9.1.2.

*The HP-IL System : An Introductory Guide to the Hewlett-Packard Interface Loop*, par Gery Kane, Steve Harper et David Ushijima, aux éditions Mc Graw-Hill.

*The HP-IL Interface Specification* référence HP-82166-90017. D'abord difficile, mais référence la plus précise.

## Mots-clés associés

RESTORE IO, SEND

# NLOOP (suite)

**Auteur**

Jean-François Garnier

NPRIM (Number of PRIME numbers) renvoie le nombre de nombres premiers entre deux nombres.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input checked="" type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

NPRIM ( n1 , n2 )
-------------------

## Exemple

A=NPRIM(10,10000)

Renvoie 1225 en 1'30". Il y a 1225 nombres premiers entre 10 et 10000.

## Paramètre d'entrée

Élément	Description	Restriction
n1, n2	Expressions numériques.	Nombres entiers compris entre $-10^{12}+1$ et $10^{12}-1$ .

## Opération

NPRIM renvoie le nombre de nombres premiers dans un intervalle, y compris les bornes.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur JPC ERR:Function Interrupted.

## Références

JPC 35 (page 21) première version de DIVILEX par Guy Toublanc.

JPC 38 (page 18) deuxième version par Guy Toublanc.

JPC 48 (page 23) troisième version de DIVILEX par Guy Toublanc.

L'ancienne appellation de NPRIM était NPRM.

## Mots-clés associés

FPRIM, PHI, PRIM

## Auteur

Guy Toublanc

# OPCODE\$

OPCODE\$ renvoie en clair la mnémonique de l'instruction assembleur située à l'adresse spécifiée.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

OPCODE\$ ( adresse hexadécimale )  
OPCODE\$ ( adresse décimale )

## Exemples

A\$=OPCODE\$( "0BD38" )	Place la chaîne A=DAT1 7 dans la variable A\$, soit la mnémonique de la première instruction de la routine POP1S.
DISP OPCODE\$(ENTRY\$( "LOG" ) )	Affiche GOSUB #0BD8D, soit la première instruction exécutée par le fonction LOG.
A\$=OPCODE\$(A+L)	Renvoie la mnémonique de l'instruction située à l'adresse donnée par l'opération A+L.

## Paramètres d'entrée

Élément	Description	Restriction
adresse hexadécimale	Expression alphanumérique contenant des chiffres hexadécimaux.	Jusqu'à 5 chiffres en majuscules ou en minuscules.
adresse décimale	Expression numérique arrondie à un nombre entier.	0 à 1048575.

## Opération

### La fonction OPCODE\$ :

OPCODE\$ renvoie la mnémonique de l'instruction assembleur pointée par l'adresse spécifiée.

Elle fournit un moyen simple de constituer un désassembleur automatique. Par exemple, le petit programme Basic suivant est un désassembleur très rapide, puissant et compact. A titre indicatif, il remplace un programme Basic d'environ 4 Ko.

```
100 DIM I$[8],A$[5]           ! dimensionne les chaînes
110 FINPUT I$,"Keyword : ",A   ! introduction du mot-clef
120 IF NOT A THEN END
130 A$=ENTRY$(I$)             ! adresse du point d'entrée
140 ATTN OFF
150 WHILE KEY$#"#43"          ! tant que pas [ATTN]
160  DISP OPCODE$(A$)         !  afficher l'instruction
170  A$=NEXTOP$(A$)           !  adresse suivante
180 END WHILE                 !  fin tant que
190 ATTN ON
200 BEEP
```

### Les mnémoniques :

Les mnémoniques renvoyées par OPCODE\$ sont conformes à la syntaxe de l'assembleur du HP-71, à la seule exception près que les instructions conditionnelles sont regroupées sur une seule ligne, et séparées par une barre oblique (/).

Il est alors possible de les séparer. Par exemple, si l'on voulait en tenir compte dans le programme ci-dessus, on remplacerait la ligne 160 par :

```
160 O$=OPCODE$(A$)           ! l'instruction
161 P=POS(O$,"/")
162 IF P THEN                 ! Y-a-t'il une barre oblique ?
163  DISP O$[1,P-1]          ! oui : affichage du test
164  DISP O$[P+1]           !      puis du RTNYES ou GOYES
165 ELSE                      ! non : affichage de la mnémonique
166  DISP O$                 !      non modifiée
```

167 END IF

Les mnémoniques ne peuvent excéder 23 caractères. Un DIM 0\$[23] est donc recommandé en ligne 100 pour compléter les déclarations.

Les instructions mettant en oeuvre des constantes inférieures à 16, telles que P= n, A=DAT1 n, ST=1 n sont renvoyées avec les constantes en *décimal*.

Les instructions mettant en oeuvre des adresses absolues (instructions de branchement, D0=(n) etc.) sont affichées avec des constantes en *hexadécimal*, avec un symbole # devant la constante. Ceci est compatible avec les assembleurs du module Forth ou AREUH pour les machines MS/DOS ou HP-UX.

L'instruction LCHEX est un cas à part. Le libellé même de l'instruction indique une constante en *hexadécimal*. Il n'y a donc pas de symbole # pour lever une éventuelle ambiguïté.

## Références

*Manuel du module Forth / Assembler* : pages 55 et suivantes. Bonne introduction aux instructions du microprocesseur du HP-71.

*Internal Design Specification* Volume I (Chapitre 16). Description détaillée et exhaustive du jeu d'instructions.

*Internal Design Specification* Volume III. La manière dont le HP-71 utilise ses instructions...

## Mots-clés associés

NEXTOP\$, PEEK\$, SYSEDIT

## Auteurs

Pierre David, Jean-Jacques Dhénin, Janick Taillandier

# PAGELEN

PAGELEN (PAGE LENGTH) sélectionne les longueurs physique et logique sur l'appareil d'impression.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input type="checkbox"/> Opération d'unité               |

PAGELEN  
PAGELEN *taille physique*  
PAGELEN *taille physique* , *taille logique*

## Paramètres d'entrée

Élément	Description	Restriction
taille physique	Expression numérique arrondie à un entier.	0 à 999.
taille logique	Expression numérique arrondie à un entier. Défaut : 72, 66	0 à 999.

## Opération

PAGELEN sert à définir la taille des pages sur les imprimantes Hewlett-Packard.

### Généralités :

La taille physique des pages correspond à la taille d'une feuille, comptée en nombre de lignes. Cette taille peut varier si le nombre de lignes par pouces varie.

Par exemple, pour un nombre de lignes par pouces égal à 6 (valeur par défaut des imprimantes en général), les feuilles au format 12 pouces ont 72 lignes, et 66 pour le format 11 pouces.

Le format 12 pouces correspond à peu près au format A4 international. Le format 11 pouces est le format utilisé aux Etats-Unis.

La taille logique, quant à elle, délimite la zone imprimable dans une feuille. C'est le nombre de lignes que l'on peut imprimer avant que l'imprimante ne passe à la page suivante, si le mode «saut de perforations» est actif. Positionner cette taille à 66 quand la taille physique vaut 72 équivaut à créer une marge de 6 lignes, répartie entre marge haute et marge basse.

### Le mot-clef PAGELEN :

PAGELEN admet 3 formes :

La première s'utilise sans paramètre. Elle règle la taille physique à 72, la taille logique à 66, et demande un saut de perforations. PAGELEN sans aucun paramètre est donc équivalent à :

PAGELEN 72,66 @ PERF ON

La taille internationale est prise par défaut. Si vous utilisez du papier 11 pouces, il suffira de faire PERF ON, car les imprimantes sont en général en format 11 pouces par défaut.

La deuxième forme admet comme seul paramètre la taille physique du papier. Il faut faire attention, car certaines imprimantes, notamment la ThinkJet, mettent une valeur par défaut à la taille logique après avoir reçu une taille physique. Reportez-vous au manuel de votre imprimante.

La troisième forme admet les deux paramètres.

Attention : seule la première forme demande à l'imprimante un saut de perforations. Les deux dernières ne le font pas, il faudra l'exécuter vous-même par l'ordre PERF ON.

### Codes envoyés à l'imprimante :

PAGELEN : ESC & l 72 p 66 f 1 L

PAGELEN x : ESC & l x P

PAGELEN x, y : ESC & l x P ESC & l y F

## Références

JPC 26 (page 39) première version par Pierre David.

*JPC 40* (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

L'ancienne dénomination de PAGELEN était PL.

## **Mots-clés associés**

PERF, PFF

## **Auteur**

Pierre David

# PAINT

PAINT teste ou allume un point sur l'écran graphique du HP-71, et retourne l'état du point avant modification éventuelle.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

PAINT ( x , y ) PAINT ( état , x , y )
---

## Exemples

C=PAINT(X, Y)

Retourne dans la variable C la valeur du point de coordonnées X et Y.

```
10 FOR X=1 TO 132
20   A=PAINT(1, X, 5)
30 NEXT X
```

Trace une ligne horizontale sur l'écran du HP-71.

## Paramètres d'entrée

Élément	Description	Restriction
état	Expression numérique nulle ou non nulle.	Aucune
x	Expression numérique arrondie à un entier.	0<=x<=131
y	Expression numérique arrondie à un entier.	0<=y<=7

## Opération

PAINT est utilisé pour faire du graphique sur l'afficheur interne du HP-71. Cette fonction a deux objectifs :

- Changer l'état des points individuellement sur l'écran. Si *état* est nul, le point de coordonnées (x,y) est éteint. Sinon, le point est allumé.

- Renvoyer l'ancien état du point. Le résultat de PAINT est cet état.

L'origine des coordonnées est le coin supérieur gauche de la fenêtre d'affichage. Le coin inférieur droit a comme coordonnées x=131 et y=7.

Par exemple, pour tracer une droite sur l'afficheur du HP-71, on peut faire :

```
10 FOR X=0 TO 131
20   C=PAINT(1, X, 5)
30 NEXT X
```

## Références

JPC 19 (page 25) routine d'inversion de l'afficheur graphique en Forth par Jean-Pierre Bondu.

JPC 24 (page 37) première version par Jean-Jacques Moreau.

JPC 25 (page 59) programme en Forth renvoyant l'adresse d'une colonne graphique par Jean-Pierre Bondu.

A paraître : deuxième version par Pierre David et Janick Taillandier.

Internal Design Specification Volume I, chapitre 3.2.1.

## Mots-clés associés

GDISP, GDISP\$, INVERSE

## **Auteurs**

Pierre David, Jean-Jacques Moreau, Janick Taillandier

# PARPOLL

PARPOLL (Parallel POLL) retourne le résultat d'une interrogation parallèle de la boucle HP-IL.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input checked="" type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input type="checkbox"/> Opération d'unité

PARPOLL PARPOLL ( <i>numéro de boucle</i> )
--

## Exemple

```
IF PARPOLL(1) THEN GOSUB 'INTERPT'
```

Exécute la routine 'INTERPT' si un périphérique validé demande service.

## Paramètre d'entrée

Élément	Description	Restriction
numéro de boucle	Expression numérique arrondie à un entier. Défaut : 1	1 à 3.

## Opération

### L'interrogation parallèle :

L'interrogation parallèle permet à un périphérique de signaler au contrôleur (le HP-71) une demande de service. L'interrogation parallèle est le moyen le plus efficace de surveiller l'état d'un ou plusieurs périphériques.

Avant de pouvoir utiliser l'interrogation parallèle, un périphérique doit être validé par le contrôleur en utilisant le message HP-IL PPE *n* (Parallel Poll Enable). Le contrôleur attribue le numéro *n* (de 0 à 7) au périphérique par cette opération. C'est la phase d'initialisation.

Le contrôleur peut alors envoyer un message IDY 00. Si le périphérique demande un service, il positionne le bit numéro *n* de la partie donnée du message IDY, ainsi que le bit de demande de service. Le IDY 00 se transforme donc en ISR *m*, avec  $m = 2^n$ .

Certains périphériques n'ont pas la capacité de répondre aux interrogations parallèles. Consulter le manuel d'utilisation de vos périphériques pour plus de détails.

### Utilisation de PARPOLL :

Par exemple, supposons que nous voulions configurer le périphérique situé à l'adresse 3, en lui ordonnant de mettre à 1 le bit 4, ainsi que le périphérique situé à l'adresse 7 en lui ordonnant de mettre à 1 le bit 2, lorsqu'ils auront une demande de service. La routine d'initialisation sera donc :

```
100 SEND UNL UNT ! déconfigure la boucle
110 SEND LISTEN 3 ! adresse le périphérique à l'adresse 3
120 SEND CMD 128+4 ! PPE 4
130 SEND UNL ! déconfigure le périphérique à l'adresse 3
140 SEND LISTEN 7 ! adresse le périphérique à l'adresse 7
150 SEND CMD 128+2 ! PPE 2
160 SEND UNL UNT ! fin de séquence
```

Cette séquence d'initialisation étant faite, le programme peut utiliser l'interrogation parallèle :

```
500 P=PARPOLL
510 IF BIT(P,4) THEN GOSUB 'DEV3'
520 IF BIT(P,2) THEN GOSUB 'DEV7'
```

Ce segment de programme branche au sous programme 'DEV3' (respectivement 'DEV7') si le périphérique situé à l'adresse 3 (respectivement 7) demande un service.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur HPIL ERR: Abort ed. Il peut être nécessaire d'exécuter RESTORE IO pour réactiver le système HP-IL.

## Références

*JPC 30* (page 50) première version par Jean-François Garnier.

*JPC 37* (page 33) deuxième version par Jean-François Garnier.

*The HP-IL System : An Introductory Guide to the Hewlett-Packard Interface Loop*, par Gerry Kane, Steve Harper et David Ushijima, aux éditions Mc Graw-Hill.

*The HP-IL Interface Specification*, référence HP-82166-90017. D'abord difficile, mais référence la plus précise.

L'ancienne appellation de PARPOLL était PPOLL.

## Mots-clés associés

SRQ, BIT, SEND, SPOLL

## Auteur

Jean-François Garnier

# PBLIST

PBLIST (Printer Basic List) liste un programme Basic sous forme structurée sur l'unité d'impression courante.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input type="checkbox"/> Opération d'unité               |

```
PBLIST [ INDENT indentation ][ TO destination ]
PBLIST fichier [ INDENT indentation ][ TO destination ]
PBLIST fichier , ligne départ [ INDENT indentation ][ TO destination ]
PBLIST fichier , ligne départ , ligne fin [ INDENT indentation ][ TO destination ]
```

## Exemples

PBLIST MONSUB INDENT 3

Liste le programme MONSUB, de la première à la dernière ligne, en indentant les structures par trois espaces.

PBLIST MONSUB,10

Liste la ligne 10 du programme MONSUB, sans indentation des structures.

PBLIST MONSUB,10,100 INDENT 2 TO LISTE

Liste le programme MONSUB, de la ligne 10 à la ligne 100, en indentant les structures de deux espaces et en envoyant le résultat dans le fichier LISTE.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets. Défaut : fichier courant.	Nom de fichier avec spécificateur d'unité en option.
ligne départ	Constante entière identifiant une ligne de programme. Défaut : première ligne du programme.	1 à 9999.
ligne fin	Constante entière identifiant une ligne de programme. Défaut : ligne de départ si spécifiée; sinon, dernière ligne du programme.	Ligne de départ jusqu'à 9999.
indentation	Expression numérique arrondie à un entier. Défaut : 0	0 à 255.
destination	Expression alphanumérique ou chaîne sans guillemets. Défaut : Le listage va sur le périphérique désigné par PRINTER IS.	Nom de fichier avec spécificateur d'unité en option.

## Opération

Le fonctionnement de PBLIST est identique à celui de DBLIST, à l'exception du fait que la sortie passe sur le périphérique d'impression au lieu du périphérique d'affichage.

Le fait de spécifier un fichier qui n'est pas un fichier Basic génère une erreur Invalid File Type (Type de fichier invalide, erreur 63).

Le fait de spécifier une seule ligne ne fait apparaître sur la liste que cette ligne. Si vous spécifiez un intervalle et que le HP-71 ne trouve pas la ligne de départ mais trouve une ligne portant un numéro plus élevé, le listage commence par cette ligne. L'exécution de PBLIST sans spécifier de ligne génère un listage du fichier complet.

Si le périphérique d'impression est le périphérique d'affichage, la valeur courante de DELAY détermine le délai d'affichage de chaque ligne.

La valeur de PWIDTH détermine la largeur de la ligne imprimée.

Pour arrêter le listage et afficher le curseur, il suffit d'appuyer sur [ATTN] sauf si on est en mode ATTN OFF.

## Références

JPC 18 (page 25) première version du programme JPCLISTE en Basic par Pierre David et Michel Martinet.

JPC 38 (page 24) première version de PBLIST par Jean-Pierre Bondu.

JPC 53 (page 17) deuxième version de PBLIST par Pierre David et Janick Taillandier.

## **Mots-clés associés**

ATTN, DBLIST, PLIST, DELAY, PWIDTH, MODE, tous les mots-clefs de programmation structurée

## **Auteurs**

Jean-Pierre Bondu, Pierre David, Janick Taillandier

# PCR

PCR (Print Carriage Return) remet la tête d'impression de l'imprimante en première position.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input type="checkbox"/> Opération d'unité
PCR	

## Exemple

```
PCR @ PRINT TAB(55);CHR$(124)
```

Place le chariot en début de ligne, puis place une barre verticale en cinquante cinquième colonne.

## Opération

### Le retour chariot :

PCR envoie un caractère "Retour-Chariot" (code 13) au périphérique désigné par l'ordre `PRINTER IS`, ce qui a pour effet de ramener le chariot de l'imprimante à la position initiale.

Le HP-71 garde en mémoire la position théorique de la tête d'impression de l'imprimante. C'est cette valeur qui est utilisée par l'option `TAB` de `PRINT`.

Cette position est remise à 0 après l'exécution de `PCR`. Ceci permet d'utiliser l'option `TAB` plus efficacement.

Note : ce mot-clef est très utile lorsque vous envoyez des séquences d'échappement ou des codes de contrôle à une imprimante. Le HP-71 les compte en effet dans son compte de caractères envoyés, ce qui peut induire une impression prématurée.

### Codes envoyés à l'imprimante :

Caractère de code 13.

## Références

*JPC 26* (page 39) première version par Pierre David.

*JPC 40* (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

L'ancienne dénomination de `PCR` était `CR`.

## Mots-clés associés

`PFF`, `PLF`, `PRINT TAB`

## Auteur

Pierre David

PDIR (Print DIRectory) imprime le répertoire du périphérique spécifié.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input type="checkbox"/> Opération d'unité
--	--

PDIR [ TO destination ] PDIR <i>spécificateur de fichier</i> [ TO destination ] PDIR ALL [ TO destination ]
---

## Exemples

- |                           |   |
|---------------------------|---|
| PDIR :TAPE                | Liste le répertoire de l'unité de stockage de masse :TAPE .   |
| PDIR :PORT(0) TO LISTE    | Liste le contenu du port numéro 0 dans le fichier LISTE.  |
| PDIR ALL                  | Liste tous les fichiers présents dans le HP-71.   |
| PDIR ESSAI:TAPE(2) TO A\$ | Liste tous les fichiers situés après le fichier ESSAI sur l'unité de stockage de masse :TAPE(2) dans le fichier dont le nom est contenu dans la variable A\$. |

## Paramètres d'entrée

Élément	Description	Restriction
spécificateur de fichier	Expression alphanumérique ou chaîne sans guillemets. Défaut : :MAIN	Un spécificateur d'unité ou un nom de fichier avec spécificateur d'unité en option.
destination	Expression alphanumérique ou chaîne sans guillemets. Défaut : Le listage est effectué sur l'imprimante.	Nom de fichier avec spécificateur d'unité en option.

## Opération

L'utilisation de PDIR est identique à celle de DDIR, à l'exception du fait que la sortie est faite sur le périphérique d'impression au lieu du périphérique d'affichage.

Toutefois, si une redirection (par TO) est spécifiée, il n'y a plus aucune différence entre DDIR et PDIR.

## Références

*A paraître* : première version par Jean-Jacques Dhénin.

## Mots-clés associés

CAT\$, CAT, DBLIST, DDIR, PBLIST

## Auteur

Jean-Jacques Dhénin

# PEEK\$

PEEK\$ renvoie le contenu d'une zone de mémoire spécifiée par son adresse.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

PEEK\$ ( <i>adresse hexadécimale</i> , <i>nombre de quartets</i> )
PEEK\$ ( <i>adresse décimale</i> , <i>nombre de quartets</i> )

## Exemples

A\$=PEEK\$(ADDR\$("EXEMPLE"), 2)

Place la chaîne "54" dans la variable alphanumérique A\$, soit la représentation hexadécimale du premier caractère du nom du fichier.

DISP PEEK\$(X+5, 1)

Retourne le quartet situé 5 quartets après l'adresse contenue dans la variable numérique X.

## Paramètres d'entrée

Élément	Description	Restriction
adresse hexadécimale	Expression alphanumérique contenant des chiffres hexadécimaux.	Jusqu'à 5 chiffres en majuscules ou en minuscules.
adresse décimale	Expression numérique arrondie à un nombre entier.	0 à 1048575.
nombre de quartets	Expression numérique arrondie à un nombre entier.	0 à 524287.

## Opération

PEEK\$ fonctionne de manière similaire à la fonction PEEK\$ interne au HP-71, à ceci près qu'elle ne vérifie pas la protection des zones mémoire concernées.

PEEK\$ est réservé aux connaisseurs du HP-71. Pour donner un exemple d'utilisation, la séquence suivante retourne la valeur du contraste courant :

HTD(PEEK\$("2E3FE", 1))

## Références

JPC 23 (page 37) première version par Pierre David, Laurent Istria et Michel Martinet.

Internal Design Specification volume I, et plus particulièrement le chapitre 3.

## Mots-clés associés

ADBUF\$, ADDR\$, ENTRY\$, HTA\$, HTD, PEEK\$, POKE

## Auteurs

Pierre David, Laurent Istria, Michel Martinet

PEN sélectionne une plume.

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> Ordre<br><input type="checkbox"/> Fonction<br><input type="checkbox"/> Opérateur | <input checked="" type="checkbox"/> Exécution au clavier<br><input type="checkbox"/> Mode CALC<br><input checked="" type="checkbox"/> IF...THEN...ELSE<br><input checked="" type="checkbox"/> Opération d'unité |
|--|---|

PEN *numéro de plume*

## Exemples

PEN 2

Sélectionne la plume 2.

PEN X

Sélectionne la plume numéro X.

## Paramètre d'entrée

Élément	Description	Restriction
numéro de plume	Expression numérique arrondie à un entier.	-32767 à +32767

## Opération

PEN sélectionne une plume sur l'appareil désigné par PLOTTER IS. Inactif pour les graphiques de type "raster".

## Mots-clés associés

PLOTTER IS

## Auteur

Pierre David

# PENDOWN

PENDOWN baisse la plume et imprime un point à la position courante de la plume.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

PENDOWN

## Exemple

IF X THEN PENDOWN

Baisse la plume si le contenu de la variable X est différent de 0.

## Opération

PENDOWN baisse la plume. Ceci a pour effet d'imprimer un point à la position courante de la plume.

## Mots-clés associés

PENUP, PEN, MOVE, DRAW

## Auteur

Pierre David

PENUP lève la plume

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

PENUP

## Exemple

IF NOT X THEN PENUP

Lève la plume si le contenu de la variable X est nul.

## Opération

PENUP lève la plume. Cet ordre est inactif pour les graphiques de type "raster".

## Mots-clés associés

PEN, PENDOWN

## Auteur

Pierre David

# PERF

PERF active ou désactive le mode *saut de perforations* sur l'appareil d'impression courant.

- Ordre
- Fonction
- Opérateur

- Exécution au clavier
- Mode CALC
- IF...THEN...ELSE
- Opération d'unité

PERF ON  
PERF OFF

## Exemple

PERF ON

Active le mode «saut de perforations»

## Opération

### Le saut de perforations :

Les imprimantes Hewlett-Packard qui utilisent le papier en continu ont en général un mode qui leur permet de ne pas écrire sur la jonction de deux feuilles.

PERF ON active ce mode. Dès lors, l'impression de la dernière ligne d'une page provoque automatiquement l'avancement du papier jusqu'en haut de la page suivante. On évite ainsi l'impression du texte sur les perforations séparant deux pages.

PERF OFF désactive ce mode.

### Codes envoyés à l'imprimante :

PERF ON : ESC & 1 0 L

PERF OFF : ESC & 1 1 L

## Références

*JPC 26* (page 39) première version par Pierre David.

*JPC 40* (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

## Mots-clés associés

PAGELEN, PLF, PRINT

## Auteur

Pierre David

PFF (Print Form Feed) avance le papier jusqu'au début de la page suivante.

- Ordre
- Fonction
- Opérateur

- Exécution au clavier
- Mode CALC
- IF...THEN...ELSE
- Opération d'unité

PFF

## Exemple

IF K+L>=N THEN PFF

Commence une nouvelle page si  $K+L \geq N$ .

## Opération

### Le saut de page :

PFF agit sur le périphérique désigné par le dernier ordre `PRINTER IS` effectué.

PFF passe au début de la page suivante. Le début de la page correspond au début de la page logique telle que définie par le dernier ordre `PAGELEN` effectué ou aux conditions par défaut de l'imprimante.

### Codes envoyés à l'imprimante :

Retour chariot (caractère de code 13)

Avance page (caractère de code 12)

## Références

*JPC 26* (page 39) première version par Pierre David.

*JPC 40* (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

L'ancienne appellation de PFF était FF.

## Mots-clés associés

PAGELEN, PCR, PRINTER IS

## Auteur

Pierre David

# PGCD

PGCD calcule le plus grand commun diviseur de plusieurs nombres.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

PGCD ( $arg_1$ , $arg_2$ )
PGCD ( $arg_1$ , $arg_2$ , $arg_3$ )
:
PGCD ( $arg_1$ , $arg_2$ , $arg_3$ , ... $arg_{10}$ )

## Exemple

A=PGCD(385, 210, 715)

Place 5 dans la variable A.

## Paramètre d'entrée

Élément	Description	Restriction
$arg_i$	Expressions numériques.	Nombres entiers entre $-10^{12}+1$ et $10^{12}-1$ .

## Opération

PGCD retourne le plus grand commun diviseur des nombres passés en paramètre. Ceux-ci peuvent être jusqu'à 10.

## Références

JPC 35 (page 21) première version de DIVILEX par Guy Toublanc.

JPC 38 (page 18) deuxième version par Guy Toublanc.

JPC 48 (page 23) troisième version par Guy Toublanc.

## Mots-clés associés

DIV, PPCM, PHI

## Auteur

Guy Toublanc

PHI, ou indicateur d'Euler, donne le nombre de nombres relativement premiers inférieurs à l'argument.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

PHI ( *argument* )

## Exemples

A=PHI(251) Place 250 dans la variable A.

A=PHI(999) 648

A=PHI(1) 1

## Paramètre d'entrée

Élément	Description	Restriction
argument	Expression numérique.	Nombre entier non nul entre $-10^{12}+1$ à $10^{12}-1$ .

## Opération

PHI retourne la valeur de l'indicateur d'Euler, c'est à dire le nombre d'entiers entre 1 et *argument* qui sont relativement premiers avec *argument*.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur JPC ERR:Function Interrupted.

## Références

JPC 35 (page 21) première version de DIVILEX par Guy Toublanc.

JPC 38 (page 18) deuxième version par Guy Toublanc.

JPC 48 (page 23) troisième version par Guy Toublanc.

## Mots-clés associés

PRIM, NPRIM, FPRIM

## Auteur

Guy Toublanc



PLOTTER IS déclare un appareil comme unité graphique.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

PLOTTER IS *spécificateur d'unité*

## Exemples

PLOTTER IS PRINTER(2)	Définit la deuxième imprimante de la boucle comme unité graphique
PLOTTER IS :RS232	Déclare l'interface HP-IL / RS232 comme unité graphique
PLOTTER IS *	Annule la déclaration courante.

## Paramètre d'entrée

Élément	Description	Restriction
spécificateur d'unité	Voir définition standard dans le module HPIL.	Aucune.

## Opération

L'ordre PLOTTER IS déclare une unité HPIL comme unité graphique.

PLOTTER IS et GINIT sont les seuls ordres exécutables en dehors d'une session graphique.

## Mots-clés associés

GINIT, GDUMP

## Auteur

Pierre David

# POKE

POKE place une donnée à partir de l'adresse hexadécimale donnée.

<input checked="" type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

POKE <i>adresse hexadécimale</i> , <i>donnée</i>
POKE <i>adresse décimale</i> , <i>donnée</i>

## Paramètres d'entrée

Élément	Description	Restriction
adresse hexadécimale	Expression alphanumérique contenant des chiffres hexadécimaux.	Jusqu'à 5 chiffres en majuscules ou minuscules.
adresse décimale	Expression numérique arrondie à un nombre entier.	0 à 1048575.
donnée	Expression alphanumérique contenant des chiffres hexadécimaux.	Aucune.

## Opération

POKE est identique à l'ordre système POKE, mais ne vérifie pas la protection des fichiers.

**Attention** : POKE est réservé aux experts. Une utilisation sans précaution peut endommager le contenu de la mémoire, et peut conduire au Memory Lost.

## Références

*JPC 23* (page 37) première version par Pierre David, Laurent Istria et Michel Martinet.

*Internal Design Specification*, Volume I, et plus particulièrement le chapitre 3.

## Mots-clés associés

ADBUF\$, ADDR\$, ATH\$, DTH\$, PEEK\$, POKE

## Auteurs

Pierre David, Laurent Istria, Michel Martinet

POSI (POSition in an Interval) renvoie la position dans la chaîne du premier caractère compris entre deux bornes.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

POSI ( chaîne , min )  
 POSI ( chaîne , min , max )

## Exemples

A=POSI("Valeur = 1000 F", 48, 57)

Retourne la position du premier chiffre de la chaîne car 48 est le code du caractère "0", 57 est le code du caractère "9".

A=POSI(A\$, "a", "z")

Recherche la position de la première minuscule dans la chaîne A\$.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
min	Expression alphanumérique, ou expression numérique arrondie à un entier.	Si c'est un nombre, il doit être compris entre 0 et 255.
max	Expression alphanumérique, ou expression numérique arrondie à un entier. Défaut : 255.	Si c'est un nombre, il doit être compris entre 0 et 255.

## Opération

POSI renvoie la position dans la chaîne *chaîne* du premier caractère dont le code est compris entre les deux valeurs *min* et *max*.

Ces valeurs sont spécifiées soit par des valeurs numériques, auquel cas elles doivent être comprises entre 0 et 255, soit par des chaînes alphanumériques. Dans ce dernier cas, seule la valeur du premier caractère (code ASCII) est prise en compte, comme pour la fonction NUM. La chaîne vide est interprétée comme une valeur nulle.

Si la valeur *max* n'est pas fournie, le maximum est pris par défaut. Autrement dit, tout code supérieur à *min* sera pris en compte.

Si *min* > *max*, les deux valeurs sont permutées avant de commencer la recherche.

Si aucun caractère n'est trouvé, la valeur 0 est renvoyée.

Par exemple, si vous voulez chercher une lettre majuscule dans une chaîne de caractères, vous ferez :

POSI(A\$, 65, 90) ou POSI(A\$, "A", "Z") ou POSI(A\$, "A", 90)

pour chercher une lettre minuscule :

POSI(A\$, 97, 122) ou POSI(A\$, "a", "z") ou POSI(A\$, 97, "z")

Pour chercher une lettre minuscule ou majuscule :

POSI(UPRC\$(A\$), "A", "Z")

## Référence

JPC 37 (page 25) première version de POSI par Jean-Pierre Bondu.

A paraître : deuxième version par Pierre David et Janick Taillandier.

## Mots-clés associés

POS, NUM

# POSI (suite)

## Auteurs

Jean-Pierre Bondu, Pierre David, Janick Taillandier

PPCM renvoie le plus petit commun multiple des arguments.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

```
PPCM ( arg1 , arg2 )
PPCM ( arg1 , arg2 , arg3 )
:
PPCM ( arg1 , arg2 , arg3 ... arg10 )
```

## Exemple

A=PPCM(385, 210, 715)

Renvoie 30030, le plus petit commun multiple des nombres 385, 210 et 715.

## Paramètre d'entrée

Élément	Description	Restriction
arg <sub>i</sub>	Expression numérique.	Nombres entiers compris entre $-10^{12}+1$ et $10^{12}-1$ .

## Opération

PPCM renvoie le plus petit commun multiple de tous les arguments  $arg_i$ .

## Références

JPC 35 (page 21) première version de DIVILEX par Guy Toublanc.

JPC 38 (page 18) deuxième version par Guy Toublanc.

JPC 48 (page 23) troisième version par Guy Toublanc.

## Mots-clés associés

PGCD, DIV

## Auteur

Guy Toublanc

# PRIM

PRIM teste la primalité. Il renvoie 0 ou le plus grand diviseur de l'argument.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

PRIM ( *argument* )  
PRIM ( *partie haute* , *partie basse* )

## Exemples

A=PRIM(999997874844)                      Renvoie 31622743 en 3"01.

A=PRIM(100071000730, 021)                Renvoie 10007 en 11"15.

## Paramètres d'entrée

Élément	Description	Restriction
argument	Expression numérique.	Nombre entier compris entre 1 et $10^{12}-1$ .
partie haute	Expression numérique.	Nombre entier compris entre $-10^{12}+1$ et $10^{12}-1$ .
partie basse	Expression numérique.	Entier compris entre 1 et 999.

## Opération

PRIM teste la primalité de l'argument, et renvoie 0 si le nombre est premier, ou le plus petit diviseur sinon.

Les nombres sont limités à 15 chiffres significatifs. Si l'argument comprend entre 13 et 15 chiffres, il doit être entré en deux paramètres. Voir le deuxième exemple, où :

$argument = partie\ haute * 1000 + partie\ basse$

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur JPC ERR:Function Interrupted.

## Références

JPC 26 (page 37) première version de PRIM par Olivier Arbey.

JPC 35 (page 21) première version de DIVILEX par Guy Toublanc.

JPC 38 (page 18) deuxième version par Guy Toublanc.

JPC 48 (page 23) troisième version par Guy Toublanc.

## Mots-clés associés

PGCD, NPRIM, FPRIM

## Auteurs

Olivier Arbey, Guy Toublanc

RED\$ enlève les espaces en tête et en queue de la chaîne fournie en paramètre.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

RED\$ ( chaîne )

## Exemple

A\$=RED\$(" A b c ")

Enlève les espaces en début et en fin de la chaîne, mais laisse les 3 espaces entre "b" et "c".

## Paramètre d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.

## Opération

RED\$ supprime tous les espaces en tête et en queue de la chaîne fournie en paramètre.

Exemple d'utilisation :

```
10 DIM C$[50]
20 I=0
30 REPEAT
40   I=I+1
50   C$=CAT$(I)
60   C$=RED$(C$[1,8]) ! isole le nom du fichier, enlève les espaces
70   DISP C$
80 UNTIL LEN(C$)=0    ! jusqu'à plus de nom de fichier
```

## Référence

JPC 21 (page 31) première version de RED\$ par Michel Martinet.

JPC 22 (page 35) deuxième version par Michel Martinet et Pierre David.

JPC 27 (page 34) troisième version par Michel Martinet.

## Mots-clés associés

REDUCE\$

## Auteurs

Pierre David, Michel Martinet

# REDUCE\$

REDUCE\$ réduit la chaîne, c'est à dire enlève tous les espaces superflus.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

REDUCE\$ ( chaîne )

## Exemple

A\$=REDUCE\$( " P P C "

Enlève les espaces en tête et en queue de la chaîne, et réduit le nombre d'espaces entre les mots à un.

## Paramètre d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.

## Opération

REDUCE\$ réduit une chaîne, c'est à dire en ôte tous les espaces superflus. Sont considérés comme tels :

- tous les espaces en début ou en fin de chaîne,
- tous les espaces doublés, triplés, ou plus généralement répétés entre deux mots.

## Références

JPC 21 (page 34) première version du programme de formattage en Basic par Pierre David.

JPC 26 (page 50) deuxième version du programme en Basic avec les fonctions en assembleur par Pierre David et Michel Martinet.

## Mots-clés associés

CESURE, RED\$

## Auteurs

Pierre David, Michel Martinet

RENUMREM (RENUMber REMarks) renumérote un programme en tenant compte des lignes de remarques.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

```

RENUMREM
RENUMREM nouveau début
RENUMREM nouveau début , incrément
RENUMREM nouveau début , incrément , ancien début
RENUMREM nouveau début , incrément , ancien début , ancienne fin
    
```

## Exemples

RENUMREM 1000,10,1000,2000

Renumérote le programme à partir de la ligne 1000, de 10 en 10, en commençant à la ligne 1000 jusqu'à la ligne 2000.

RENUMREM

Renumérote tout le programme à partir de la ligne 10, de 10 en 10.

## Paramètres d'entrée

Élément	Description	Restriction
nouveau début	Constante entière. Défaut : 10	1 à 9999.
incrément	Constante entière. Défaut : 10	1 à 9999.
ancien début	Constante entière. Défaut : début du fichier.	1 à 9999.
ancienne fin	Constante entière. Défaut : fin du fichier.	1 à 9999.

## Opération

Lorsqu'un programme Basic est listé avec DBLIST ou PBLIST, les numéros des lignes de remarque (REM ou !) ne sont pas imprimés. Ceci donne *apparemment* une numérotation quelque peu erratique.

RENUMREM renumérote le programme Basic courant de la même manière que RENUMBER, mais traite différemment les lignes de remarques.

Les lignes de remarques sont « collées » le plus possible à la ligne précédente, c'est à dire renumérotées de 1 en 1. Ainsi, le programme :

```

10 PRINT TAB(18);"HP-71"
30 ! première ligne de commentaire
40 ! deuxième ligne de commentaire
42 ! troisième ligne de commentaire
45 ! quatrième ligne de commentaire
50 IF KEY$="" THEN 50
    
```

deviendra, après un RENUMREM 100,10 :

```

100 PRINT TAB(18);"HP-71"
101 ! première ligne de commentaire
102 ! deuxième ligne de commentaire
103 ! troisième ligne de commentaire
104 ! quatrième ligne de commentaire
110 IF KEY$="" THEN 110
    
```

Les lignes de commentaires ont été *tassées* contre la ligne 100. La ligne 100 et 110 sont bien numérotées. Ce programme, listé par DBLIST ou PBLIST donnera :

# RENUMREM (suite)

```
100 PRINT TAB(18);"HP-71"  
  - première ligne de commentaire  
  deuxième ligne de commentaire  
  troisième ligne de commentaire  
  quatrième ligne de commentaire  
110 IF KEY$="" THEN 110
```

Ceci rend les programmes plus faciles à taper sans les commentaires, et le listage est bien plus lisible.

## Références

*JPC 18* (page 25) première version du programme JPCLISTE en Basic par Pierre David et Michel Martinet.

*JPC 38* (page 24) première version de RENUMREM par Jean-Pierre Bondu.

## Mots-clés associés

DBLIST, PBLIST, RENUMBER

## Auteur

Jean-Pierre Bondu

# REPEAT ... UNTIL

REPEAT ... UNTIL (Répéter ... jusqu'à) définit une boucle qui est répétée jusqu'à ce que l'expression dans UNTIL soit vraie.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

REPEAT <i>segment de programme</i> UNTIL <i>expression logique</i>
--

## Exemples

```
10 REPEAT
20  CALL AA(I, N)
30  I=I+1
40 UNTIL I+2>N
```

Exécute le sous programme "AA" jusqu'à ce que la condition soit vraie.

```
10 DATA FICHER1, ESSAI, EXEMPLE,
20 REPEAT
30  READ F$
40  DISP F$
50 UNTIL F$=""
```

Lit les données et les affiche, jusqu'à ce qu'il n'y en ait plus (la dernière DATA est vide).

## Paramètres d'entrée

Élément	Description	Restriction
segment de programme	Tout nombre de lignes de programme contiguës.	Aucune.
expression logique.	Expression numérique évaluée comme vraie si différente de 0.	Aucune.

## Opération

La boucle REPEAT ... UNTIL exécute le *segment de programme* de manière répétitive, jusqu'à ce que l'expression située dans le UNTIL soit vraie, c'est à dire non nulle.

L'exécution commence au premier ordre suivant le REPEAT, et continue jusqu'au UNTIL où un test est effectué. Si le test est faux, un branchement est réalisé à l'ordre suivant immédiatement le REPEAT.

Quand le test conditionnel est vrai, l'exécution du programme continue au premier ordre suivant le UNTIL.

La boucle est effectuée au moins une fois.

L'ordre LEAVE peut être utilisé pour sortir prématurément (et proprement) de la boucle.

Le segment de programme peut lui-même contenir des structures de type boucle ou choix. L'emboîtement doit être correct, sinon l'erreur JPC ERR:Structure Mismatch est renvoyée.

## Références

JPC 31 (page 38) première version par Janick Taillandier.

JPC 52 (page 10) deuxième version par Pierre David et Janick Taillandier.

Basic 5.0 pour HP 9000 séries 200/300.

## Mots-clés associés

LEAVE, LOOP ... END LOOP, WHILE ... END WHILE

# REPEAT ... UNTIL (suite)

## Auteurs

Pierre David, Janick Taillandier

# REPLACES\$

REPLACES\$ remplace une chaîne par une autre dans la chaîne spécifiée, avec ou sans les conventions de l'éditeur de textes HP.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

```
REPLACES$ ( chaîne , motif1 , motif2 )
REPLACES$ ( chaîne , motif1 , motif2 , début )
REPLACES$ ( chaîne , motif1 , motif2 , joker )
```

## Exemples

```
REPLACES$ ("A B D E F ", " ", "")
```

Enlève tous les espaces dans la chaîne spécifiée, et renvoie donc la chaîne "ABCDEF".

```
REPLACES$ ("ABxCDxEF", ".x", "", ".")
```

Affiche la chaîne "ACEF".

```
A$=REPLACES$("X1 X2 X3", "X\.", "X")
```

Place la chaîne "X X X" dans la variable A\$.

```
A$=REPLACES$("X1 X2 X3", "X\.", "X", 3)
```

Place la chaîne "X1 X X" dans la variable A\$, c'est à dire la chaîne originale traitée à partir du 3<sup>ème</sup> caractère.

## Paramètres d'entrée

Élément	Description	Restriction
chaîne	Expression alphanumérique.	Aucune.
motif1	Expression alphanumérique.	Aucune.
motif2	Expression alphanumérique.	Aucune.
début	Expression numérique arrondie à un entier. Défaut : 1	0 à 1048575.
joker	Expression alphanumérique.	Chaîne vide ou non vide.

## Opération

Ce mot-clef est rendu caduque par la fonction GENRPLC\$. Vous êtes vivement invités à utiliser GENRPLC\$. La fonction REPLACES\$ n'est là que pour assurer la compatibilité avec les programmes existants.

REPLACES\$ remplace, dans la chaîne *chaîne*, toutes les occurrences de la chaîne *motif1* par la chaîne *motif2*.

REPLACES\$ définit des conventions lors du remplacement. Elles sont identiques à celles de l'éditeur de textes EDTEXT des modules *Editeur de Textes*, *Forth / Assembleur* ou *Translator Pac*.

Ces conventions comprennent des caractères spéciaux qui permettent des opérations plus évoluées. Ces caractères sont ., @, &, ^ et \$. Ils prennent leur signification particulière lorsqu'ils sont précédés par le caractère \. Pour retourner à la signification normale des caractères, il faut un deuxième \. Deux caractères \ consécutifs sont considérés comme un simple caractère \, et non comme deux bascules.

Caractère Signification

```
-----
.      Caractère quelconque
@      Nombre indéterminé de caractères quelconques
&      Le texte de motif1 quand employé dans motif2
^      Début de ligne (doit être en début de motif1)
$      Fin de ligne (doit être en fin de motif1)
```

Le paramètre *début* spécifie à partir de quel caractère de *chaîne* la recherche et la substitution doivent commencer. Par défaut, *motif1* est cherché dans toute la chaîne *chaîne*.

Si la chaîne *joker* est présente et non nulle, elle représente un caractère quelconque de *motif1*. Dans ce cas, les caractères spéciaux de l'éditeur de texte HP sont neutralisés.

# REPLACE\$ (suite)

La fonction GENRPLC\$ est similaire à la fonction REPLACE\$. La fonction REPLACE\$ est, elle-même, le regroupement de deux fonctions plus anciennes, REPLACE\$ et RPLC\$. Cette fonction REPLACE\$ n'est laissée dans JPC Rom que pour des raisons de compatibilité ascendante, et vous êtes plutôt encouragés à utiliser GENRPLC\$. Le tableau ci-dessous résume les différences entre ces trois fonctions :

Fonction	Ancien nom	Caractéristiques
REPLACE\$	REPLACE\$	Remplacement simple avec caractère joker
REPLACE\$	RPLC\$	Chaînes génériques compatibles avec EDTEXT
GENRPLC\$	GENRPLC\$	Chaînes génériques compatibles avec XEDIT

Note : l'ancienne fonction REPLACE\$ correspond au cas où l'actuelle fonction REPLACE\$ est appelée avec un quatrième paramètre alphanumérique représentant le joker. Dans les autres cas d'appel, c'est l'ancienne fonction RPLC\$ qui est exécutée.

## Références

*JPC 23* (page 34) première version par Michel Martinet.

*JPC 35* (page 28) première version de RPLC\$ par Jean-Jacques Moreau.

Manuel de l'éditeur de textes.

REPLACE\$ inclut les fonctionnalités de l'ancienne fonction RPLC\$.

## Mots-clés associés

GENRPLC\$, MAP, MAP\$

## Auteurs

Michel Martinet, Jean-Jacques Moreau

ROMAN valide le jeu de caractères étendu *Roman*.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
ROMAN ON ROMAN OFF	

## Exemple

```
10 SUB ML
20 ROMAN ON
30 END SUB
```

Après une réinitialisation de la mémoire (Memory Lost), valide le jeu de caractères étendu Roman.

## Opération

### Le jeu de caractères Roman

Les imprimantes Hewlett-Packard ou les interfaces vidéo admettent généralement la possibilité d'utiliser des caractères accentués. Pour cela, il faut utiliser le jeu de caractères baptisé *Roman*.

Pour imprimer le caractère "é" sur une imprimante ThinkJet ou LaserJet par exemple, il suffit de faire :

```
PRINT CHR$(201)
```

La table suivante résume le jeu de caractère étendu *Roman*.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	□	p					—	â	À	Á	Þ
1		!	1	A	Q	a	q				À		ê	î	Ã	þ
2		"	2	B	R	b	r				Â		ô	ø	ã	
3		#	3	C	S	c	s				È	°	û	Æ	Ð	
4		\$	4	D	T	d	t				Ê	Ç	á	â	ð	
5		%	5	E	U	e	u				Ë	ç	é	í	Í	
6		&	6	F	V	f	v				Ï	Ñ	ó	ø	ì	-
7		□	7	G	W	g	w				Ï	ñ	ú	æ	Ó	¼
8		(	8	H	X	h	x				□	i	à	Ä	Ò	½
9		)	9	I	Y	i	y				□	ı	è	ì	Õ	a
A		*	:	J	Z	j	z				^	ı	ò	Ö	õ	°
B		+	;	K	[	k	{				□	£	ù	Ü	Š	«
C		,	<	L	\	l					~	¥	ä	É	š	■
D		-	=	M	]	m	}				Ù	š	ë	ï	Ú	»
E		.	>	N	^	n	~				Û	f	ö	ß	ÿ	±
F		/	?	O	_	o					£	¢	ü	ô	ÿ	

La première moitié de la table est l'alphabet ASCII standard (voir la fonction ASC\$). La deuxième est appelée *Extension romane*.

Dans les deux moitiés, on remarque que les deux premières colonnes ne sont pas utilisées. Ce sont les caractères de contrôle.

### Le jeu de caractères du HP-71

Le HP-71 ne connaît que la première moitié de la table ci-dessus. Il représente les caractères de la deuxième moitié par des caractères de la première, à moins que vous n'ayez défini ces caractères (selon vos désirs) à l'aide de l'ordre CHARSET standard.

## ROMAN (suite)

L'exemple d'impression ci-dessus fonctionne que vous ayez fait ROMAN ON ou non. Dans ce cas, c'est l'imprimante qui reconnaît le caractère. En revanche, votre HP-71 affichera un caractère sans aucun rapport avec "é" si vous n'avez pas fait ROMAN ON.

L'utilisation de ROMAN ON permet d'avoir les caractères accentués à l'écran du HP-71.

Si vous aviez défini des caractères à l'aide de CHARSET, ils seront réactivés lorsque vous ferez ROMAN OFF.

### Au clavier

Dans les programmes que vous rentrez, vous pouvez utiliser des caractères accentués sous la forme CHR\$(...). Cependant, il est plus facile et plus lisible de vous créer des définitions de touches. Par exemple, pour faire le journal, nous utilisons les définitions suivantes :

```
DEF KEY 'fW', CHR$(197);    é
DEF KEY 'fE', CHR$(193);    ê
DEF KEY 'fR', CHR$(201);    è
DEF KEY 'fY', CHR$(203);    ù
DEF KEY 'fU', CHR$(195);    û
DEF KEY 'fI', CHR$(209);    î
DEF KEY 'fO', CHR$(194);    ô
DEF KEY 'f/', CHR$(92);     \
DEF KEY 'fA', CHR$(192);    â
DEF KEY 'fS', CHR$(200);    à
DEF KEY 'fD', CHR$(205);    ë
DEF KEY 'fJ', CHR$(207);    ü
DEF KEY 'fK', CHR$(221);    ï
DEF KEY 'f*', CHR$(124);    |
DEF KEY 'fC', CHR$(181);    ç
```

Notez le point virgule après les définitions et l'absence de guillemets autour de CHR\$(...). Cela a pour effet d'introduire le caractère et non la chaîne CHR\$(...). Il nous suffit donc d'appuyer sur la touche [f][W] pour avoir un "é" à l'affichage.

### Pour résumer

Les imprimantes HP utilisent le jeu de caractères *Roman*.

ROMAN ON remplace le jeu de caractères (de codes supérieurs à 128) par l'extension *Romane*.

Les définitions de touches permettent de faciliter l'introduction au clavier des caractères *Roman*.

ROMAN OFF enlève l'extension *Romane* et active le précédent jeu de caractères.

### Références

*JPC 35* (page 8) première version de CHARLEX par Pierre David.

*A paraître* : première version de ROMAN par Pierre David et Janick Taillandier.

Consultez en outre le manuel de votre imprimante.

### Mots-clés associés

ASC\$, CHARSET\$, CHARSET, CHR\$, DEF KEY, NUM, MAP, MAP\$

### Auteurs

Pierre David, Janick Taillandier

RREC\$ (Read REcord) lit un enregistrement sur le support de mémoire de masse indiqué.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input type="checkbox"/> Opération d'unité

RREC\$ ( *adresse* , *spécificateur de périphérique* )

## Exemples

DIM A\$[256]  
A\$=RREC\$(2, " :TAPE" )

Lit le premier enregistrement de la partie répertoire du support.

A\$=RREC\$(2, ".VOLLBL :2" )

Lit le deuxième enregistrement du support repéré par le volume label VOLLBL sur la deuxième boucle HP-IL.

## Paramètres d'entrée

Élément	Description	Restriction
adresse	Expression numérique arrondie à un entier, ou expression alphanumérique contenant des nombres hexadécimaux représentant une adresse sur le support.	Valeur comprise entre 0 et la taille maximum du support.
spécificateur de périphérique	Expression numérique ou alphanumérique. Voir les définitions standard HP-IL.	Ne peut pas être une chaîne non comprise entre guillemets.

## Opération

L'unité de base des transferts entre HP-71 et unité de mémoire de masse est le *secteur*, ou bloc de 256 octets.  
RREC\$ lit les 256 octets d'un enregistrement du support.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur HPIL ERR:Abort ed. Il peut être nécessaire d'exécuter RESTORE IO pour réactiver le système HP-IL.

## Références

JPC 45 (page 15) première version par Michel Martinet.  
Manuel d'utilisation du lecteur de cassettes numériques HP-82161A.  
Manuel du module HP-IL pour le HP-71, chapitre 3 et annexe D.

## Mots-clés associés

ENTER, WREC

## Auteur

Michel Martinet

# SELECT ... CASE ... END SELECT

La construction SELECT ... CASE ... END SELECT donne une exécution conditionnée de segments de programmes.

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> Ordre<br><input type="checkbox"/> Fonction<br><input type="checkbox"/> Opérateur | <input checked="" type="checkbox"/> Exécution au clavier<br><input type="checkbox"/> Mode CALC<br><input checked="" type="checkbox"/> IF...THEN...ELSE<br><input checked="" type="checkbox"/> Opération d'unité |
|--|---|

```
SELECT expression
  CASE clauses de choix
    segment de programme
  CASE clauses de choix
    segment de programme
  :
  [ CASE ELSE
    segment de programme ]
END SELECT
```

## Exemples

```
10 SELECT E+2
20 CASE <0
30   DISP "Positif"
40 CASE =0
50   DISP "Nul"
60 CASE ELSE
70   DISP "Autre" @ BEEP
80 END SELECT
```

Renvoie un message approprié en fonction de l'expression E+2.

```
10 SELECT E$
20 CASE "A" TO "Z"
30   DISP "Majuscule"
40 CASE ":", ";", ",", ".", " "
50   DISP "Ponctuation"
60 END SELECT
```

Il n'y a pas de CASE ELSE. Le premier choix est de type "intervalle", le deuxième est de type "énumération".

## Paramètres d'entrée

Élément	Description	Restriction
expression	Expression numérique ou alphanumérique.	Aucune.
clauses de choix	voir CASE	Le type des expressions doit être compatible avec celle du SELECT.
segment de programme	Tout nombre de lignes contiguës.	Aucune.

## Opération

SELECT ... END SELECT est similaire à la construction IF ... THEN ... ELSE ... END IF, mais permet de définir **plusieurs** segments de programme. Un seul d'entre eux sera exécuté. Chaque segment de programme commence avec CASE ou CASE ELSE, et finit quand l'ordre suivant est CASE, CASE ELSE ou END SELECT.

L'ordre SELECT spécifie une expression dont la valeur est comparée à *chaque clause de choix*. Quand une occurrence est trouvée, le segment de programme correspondant est exécuté. Les segments restants sont sautés, et l'exécution continue immédiatement après le END SELECT.

Toutes les *clauses de choix* doivent être du même type (soit numérique, soit alphanumérique), et doivent correspondre au type de l'expression de SELECT.

Le CASE ELSE optionnel définit un segment de programme à exécuter lorsque toutes les *clauses de choix* ont échoué.

Les erreurs rencontrées lors de l'évaluation des *clauses de choix* seront reportées au niveau du SELECT.

# SELECT ... CASE ... END SELECT (suite)

Les segments de programme peuvent eux-mêmes contenir des structures de type boucle ou choix. L'emboîtement doit être correct, sinon l'erreur JPC ERR:Structure Mismatch est renvoyée.

## Références

*JPC 52* (page 10) première version par Pierre David et Janick Taillandier.  
Basic 5.0 pour HP 9000 séries 200/300.

## Mots-clés associés

IF ... THEN ... ELSE ... END IF

## Auteurs

Pierre David, Janick Taillandier

# SHRINK

SHRINK minimise en mémoire la taille d'un fichier texte provenant d'une unité de mémoire de masse, et récupère la place inutilisée.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

SHRINK <i>fichier</i>
-----------------------

## Exemple

```
10 COPY :TAPE TO A$  
20 SHRINK A$  
30 PURGE A$&" :TAPE"  
40 COPY A$ TO :TAPE
```

Copie le fichier dont le nom est dans la variable A\$, le réduit, l'efface du support (très important), puis le recopie sur le support.

## Paramètre d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemet.	Nom de fichier avec un spécificateur de périphérique (interne) optionnel.

## Opération

### Le problème :

Lorsque des fichiers texte sont stockés sur support magnétique, leur taille est arrondie au secteur supérieur (soit au multiple de 256 octets supérieur).

Si vous rappelez un fichier texte depuis une unité de stockage de masse, puis que vous y ajoutez ne serait-ce qu'une lettre par l'intermédiaire de votre traitement de texte favori, sa taille augmentera. La prochaine copie sur le support magnétique vous montrera que la simple insertion d'une lettre prend 256 octets de plus, soit 255 inutiles. En répétant le processus, on arrive très vite à des tailles gigantesques pour des fichiers normaux plein d'espaces vides.

Un autre cas de fichier texte contenant de l'espace inutilisé est la création par l'ordre CREATE en donnant une taille initiale du fichier, ou en faisant un PRINT # alors que le pointeur de fichier est au milieu du fichier texte.

Cet espace inutilisé correspond à la zone comprise entre le marqueur de fin de fichier et la fin physique du fichier.

### La solution :

L'ordre SHRINK compresse les fichiers texte en mémoire de manière à ce qu'il n'y ait plus d'espace inutilisé.

Avant de recopier les fichiers vers le support, il faut prendre soin de les effacer, sinon l'ancienne taille subsiste, et le fichier pourrait encore grossir !

## Références

JPC 35 (page 35) première version par Jean-Jacques Moreau.

## Mots-clés associés

COPY, EDTEXT, PURGE

## Auteur

Jean-Jacques Moreau

SLEEP place le HP-71 en état de sommeil léger.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

SLEEP

## Exemple

10 IF NOT KEYDOWN THEN SLEEP

Place le HP-71 en sommeil léger si aucune touche n'est pressée.

## Opération

Dans certaines conditions, il faut pouvoir obtenir des réponses rapides à des événements espacés. Ceci arrive, par exemple, dans des applications de mesure ou de contrôle de processus.

Placer le HP-71 en sommeil profond par OFF ou BYE nécessiterait trop de temps pour remettre en marche la machinerie. D'autre part, le laisser allumé pose des problèmes d'autonomie.

SLEEP place le HP-71 en sommeil léger, c'est à dire le place dans un état de veille. L'affichage est maintenu allumé, et les réactions aux interruptions de l'HP-IL ou du clavier sont plus rapides.

N'importe quelle touche suffit à rallumer le HP-71. De manière plus générale, n'importe quelle interruption suffit à rallumer le HP-71.

## Références

JPC 30 (page 50) première version par Jean-François Garnier.

JPC 37 (page 33) deuxième version par Jean-François Garnier.

## Mots-clés associés

BYE, KEYWAIT\$, OFF, ON TIMER, ON INTR

## Auteur

Jean-François Garnier

# SPACE\$

SPACE\$ renvoie une chaîne composée du nombre spécifié d'espaces.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

SPACE\$ ( <i>nombre de répétitions</i> )
SPACE\$ ( <i>chaîne</i> , <i>nombre de répétitions</i> )
SPACE\$ ( <i>caractère</i> , <i>nombre de répétitions</i> )

## Exemples

A\$=SPACE\$(5)

Place une chaîne composée de 5 espaces dans la variable alphanumérique A\$.

10 DIM X\$[50]

20 X\$=SPACE\$("ESSAI",10)

Répète 10 fois la chaîne "ESSAI" et place le résultat dans la variable X\$.

DISP SPACE\$(65,20)

Affiche le caractère de code 65 (la lettre "A" ici) 20 fois.

## Paramètres d'entrée

Élément	Description	Restriction
nombre de répétitions	Expression numérique arrondie à un entier.	-1048575 à 1048575.
chaîne	Expression alphanumérique.	Aucune
caractère	Expression numérique arrondie à un entier.	0 à 255.

## Opération

SPACE\$ renvoie une chaîne composée du nombre d'espaces spécifié. Si ce nombre est négatif ou nul, la chaîne vide est renvoyée.

SPACE\$ est utile pour toutes les applications de formatage de textes, pour positionner une marge, aligner des colonnes, centrer du texte, et plus encore. Il peut aussi servir à initialiser les tableaux utilisés avec FINPUT.

SPACE\$ permet aussi de répéter une chaîne de caractères quelconque. Ce sont les deuxième et troisième formes. Dans le troisième cas, la chaîne est composée du caractère de code ASCII spécifié.

Ainsi, SPACE\$("ESSAI ",3) renvoie la chaîne ESSAI ESSAI ESSAI. De même, SPACE\$(65,2) renvoie AA, 65 représentant le code ASCII de la lettre A.

## Références

JPC 21 (page 34) première version du programme de formatage en basic par Pierre David.

JPC 26 (page 50) deuxième version du programme en Basic avec les fonctions en assembleur par Pierre David et Michel Martinet.

## Mots-clés associés

CENTER\$, FORMAT\$

## Auteurs

Pierre David, Michel Martinet

SRQ (Service ReQuest) envoie un message sur la boucle HP-IL pour tester si un périphérique demande service.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input checked="" type="checkbox"/> Mode CALC            |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input type="checkbox"/> Opération d'unité               |

SRQ  
SRQ ( *numéro de boucle* )

## Exemple

```
N=SRQ(2) @ IF N THEN GOSUB 'INTERPT'
```

Si un périphérique demande service sur la deuxième boucle HP-IL, exécuter le sous programme spécifié.

## Paramètre d'entrée

Élément	Description	Restriction
numéro de boucle	Expression numérique arrondie à un entier. Défaut : 1	1 à 3.

## Opération

### La demande de service :

La demande de service est un dispositif permettant à des périphériques de signaler un état ou de demander un service au contrôleur HP-IL.

Par exemple, l'imprimante ThinkJet signale qu'elle n'a plus de papier en émettant des demandes de service.

La demande de service est un état dans lequel rentre le périphérique.

Si le contrôleur (le HP-71) envoie des octets de données (messages DAB, envoyés par SEND DATA) ou des messages d'identification (IDY), le périphérique met à un bit spécial dans le message, signifiant la demande de service.

Après la réception de la demande de service, le HP-71 devra interroger l'état des périphériques, par la fonction SPOLL, pour déterminer lequel demande le service.

SRQ fournit une méthode rapide pour déterminer si un périphérique émet une demande de service. A la différence de l'interrogation parallèle, il n'aide pas à l'identification du périphérique qui demande le service. D'un autre côté, SRQ ne requiert aucune routine d'initialisation.

### La fonction :

SRQ envoie un message IDY 00 sur la boucle spécifiée par le paramètre optionnel, puis renvoie l'état du bit de service. La réponse est donc :

- 1 si un ou plusieurs périphériques demandent service, ou
- 0 si aucun périphérique ne demande service.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur HPIL ERR:Abort ed. Il peut être nécessaire d'exécuter RESTORE IO pour réactiver le système HP-IL.

## Références

JPC 30 (page 50) première version par Jean-François Garnier.

JPC 37 (page 33) deuxième version par Jean-François Garnier.

*The HP-IL System : An Introductory Guide to the Hewlett-Packard Interface Loop*, par Gerry Kane, Steve Harper, et David Ushijima aux éditions Mc Graw-Hill.

*The HP-IL Interface Specification*, référence HP-82166-90017. D'abord difficile, mais référence la plus précise.

# SRQ (suite)

## Mots-clés associés

PARPOLL, SEND

## Auteur

Jean-François Garnier

STACK initialise la pile de commandes au nombre d'entrées spécifié.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

  

STACK *nombre de niveaux*

## Exemple

STACK 15

Initialise la pile de commandes à 15 niveaux.

## Paramètre d'entrée

Élément	Description	Restriction
nombre de niveaux	Expression numérique arrondie à un entier.	1 à 16.

## Opération

### La pile de commandes :

La pile de commandes est un outil bien pratique pour épargner des efforts de frappe. Cependant, cette pile est limitée à 5 niveaux sur un HP-71 standard.

### L'ordre STACK :

L'ordre STACK initialise la pile de commandes au nombre de niveaux spécifié. La pile de commandes est effacée.

Ce nombre de niveaux est réglable entre 1 et 16. Une profondeur de 1 est la moins gourmande en mémoire, mais la moins utile pour aider à la frappe.

La profondeur de 16 niveaux n'est pas conseillée. En effet, lors d'un INPUT ou LINPUT, une pression sur la touche [↓] monte dans la pile. Ce comportement est étrange, voire perturbant.

Le nombre de niveaux idéal est 15. C'est typiquement le genre de commandes à mettre dans le programme ML lancé automatiquement après l'initialisation complète de la machine.

## Références

*User's Library Solutions - Utilities*, sous programme SETCMDST (page 3).

*JPC 25* (page 57) première version par Michel Martinet.

*JPC 60* (page 24) deuxième version par Henri Kudelski.

## Mots-clés associés

INIT 3

## Auteurs

Henri Kudelski, Michel Martinet

# STARTUP\$

STARTUP\$ renvoie la chaîne de commandes spécifiée par STARTUP.

- |  |  |
|--|--|
| <input type="checkbox"/> Ordre               | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur           | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|  | <input checked="" type="checkbox"/> Opération d'unité    |

STARTUP\$

## Exemple

A\$=STARTUP\$

Place dans la variable A\$ la chaîne de commandes exécutée à chaque mise sous tension du HP-71.

## Opération

STARTUP\$ renvoie la chaîne de commandes devant être exécutée à la mise sous tension du HP-71. La longueur de cette chaîne ne peut excéder 95 caractères.

STARTUP\$ renvoie une chaîne nulle si aucune commande n'avait été spécifiée par STARTUP.

## Référence

*JPC 25* (page 43) première version par Jean-Jacques Moreau.

*JPC 31* (page 29) deuxième version par Jean-Jacques Moreau.

## Mots-clés associés

ENDUP\$, STARTUP

## Auteur

Jean-Jacques Moreau

SYSEDIT (SYStem EDITor) place le HP-71 dans un mode d'édition / désassemblage interactif de la mémoire.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

SYSEDIT <i>adresse hexadécimale</i> SYSEDIT <i>adresse décimale</i>
--

## Exemples

SYSEDIT "0BD38"	Examine le contenu de la mémoire en commençant à l'adresse 0BD38, c'est à dire au début de la routine POP1S.
SYSEDIT ADDR\$( "EXEMPLE" )	Examine la mémoire en commençant au début de l'en-tête du fichier EXEMPLE.
SYSEDIT HTD(ADDR\$( "X" ) )+37	Commence l'examen de la mémoire après l'en-tête du fichier X.

## Paramètres d'entrée

Élément	Description	Restriction
adresse hexadécimale	Expression alphanumérique contenant des chiffres hexadécimaux.	Jusqu'à 5 chiffres en majuscules ou en minuscules.
adresse décimale	Expression numérique arrondie à un nombre entier.	0 à 1048575.

## Opération

SYSEDIT place le HP-71 dans un mode d'édition / désassemblage de la mémoire. Ce mode permet de visualiser le contenu de la mémoire sous forme hexadécimale, sous forme de caractères, ou sous forme d'instructions ou de macro-opérations assembleur. L'affichage est toujours organisé comme suit :

00000 : 2034EE100060F481

La première partie est l'adresse.

La deuxième partie, séparée de la première par le caractère : représente le contenu de la mémoire à partir de cette adresse.

Le HP-71 réagit alors à certaines pressions de touches.

**[ATTN]** ou **[f][OFF]** - Sortie de l'éditeur

Sort de SYSEDIT.

**[+]**, **[-]**, **[\*]** ou **[/]** - Déplace la fenêtre d'affichage

Opérations sur l'adresse courante. Respectivement : +1, -1, +16 et -16. Dans tous les cas, le mode hexadécimal est remis par défaut.

**[A][1]** à **[A][8]** - Visualisation en ASCII

Affiche le contenu de la mémoire sous forme de NIBASC. Le nombre qui suit le [A] est le nombre de caractères désiré. Les caractères non affichables (ceux dont le code n'est pas compris entre 32 et 126) sont représentés par un point.

**[N][1]** à **[N][9]** et **[N][.][0]** à **[N][.][6]** - Visualisation en hexadécimal

Affiche le contenu de la mémoire sous forme de NIBHEX. Le nombre qui suit le [N] est le nombre de chiffres hexadécimaux désiré. Si celui-ci est supérieur à 9, il faut mettre un point avant le chiffre des unités. Par exemple, [N] [8] affiche 8 chiffres, alors que [N] [.] [6] en affiche 16.

**[C][1]** à **[C][6]** - Constante décimale

Affiche le contenu de la mémoire sous forme de CON. Le nombre qui suit le [C] est le nombre de quartets formant la constante. Si ce nombre est inférieur ou égal à 4, la constante est décimale, sinon elle est hexadécimale (repérée par un symbole #).

**[C][H][1]** à **[C][H][6]** - Constante hexadécimale

Cette option est la même que précédemment, mais force l'affichage de la constante en mode hexadécimale.

**[R][1]** à **[R][5]** - Adresse relative

# SYSEDIT (suite)

Affiche le contenu de la mémoire sous forme d'une adresse relative (macro-opération REL). L'adresse calculée est toujours affichée en hexadécimal.

**[H]** - Mode hexadécimal

Active le mode hexadécimal : affiche le contenu de la mémoire sous forme de 16 chiffres hexadécimaux. C'est le mode par défaut lorsqu'on rentre sous SYSEDIT.

**[D]** - Mode désassemblage

Active le mode désassemblage : affiche le contenu de la mémoire sous forme de mnémoniques assembleur. Voir la fonction `OPCODE$` pour plus de détails sur le format de la sortie.

**[L]** - Visualise l'instruction courante comme LCASC

Si le mode désassemblage est actif, et si l'instruction désassemblée est `LC(n)`, avec  $n$  multiple de 2, affiche l'instruction courante sous forme de LCASC. Les caractères non affichables (ceux dont le code n'est pas compris entre 32 et 126) sont représentés par un point.

**[F]** - Sauvegarde dans un fichier

Demande un nom de fichier. Chaque pression ultérieure sur `[ENDLINE]` stockera le contenu de l'affichage à la fin du fichier. Le fichier est créé s'il n'existe pas. Sinon, le pointeur est placé à la fin. Pour supprimer la sortie dans un fichier, effacer l'affichage avec `[f]` `[-LINE]` et valider.

**[=]** - Déplacement direct

Si l'affichage montre une adresse, positionne l'adresse courante à cette adresse.

**[I]** - Déplacement et empilement

Si l'affichage montre une adresse, sauvegarde l'adresse courante dans une pile, et positionne l'adresse courante à cette adresse. La pile peut contenir un maximum de 7 adresses. Ceci vous permet d'aller désassembler à une adresse, et de reprendre à l'ancienne adresse.

**[J]** - Retour

Si la pile d'adresses n'est pas vide, dépile la dernière adresse et positionne l'adresse courante.

**[ENDLINE]** - Validation

Si la sauvegarde dans un fichier est active, sauve l'affichage dans le fichier. Saute l'objet courant. Incrémente l'adresse du nombre de quartets occupé par l'objet (opcode, caractères, adresse relative, constante). Si le mode désassemblage est actif, affiche le mnémonique de la prochaine instruction, sinon passe en mode hexadécimal.

**[Z]** - Edition du pointeur

Modifie interactivement la valeur de l'adresse. Pour valider la nouvelle adresse, faire `[ENDLINE]`. Pour arrêter et ne rien modifier, appuyer sur `[ATTN]`.

**[f][Z]** ou **[M]** - Edition de la mémoire

Modifie interactivement le contenu de la mémoire (affiché en hexadécimal). Pour valider, faire `[ENDLINE]`. Pour arrêter et ne rien modifier, appuyer sur `[ATTN]`.

## Références

*JPC 22* (page 31) première version en Basic de désassembleur par Michel Martinet.

Non publié : première version de SYSEDIT en Basic par Pierre David.

Non publié : SYSEDIT par Pierre David et Janick Taillandier.

*Manuel du module Forth / Assembler* : pages 55 et suivantes. Bonne introduction aux instructions du microprocesseur du HP-71.

*Internal Design Specification* Volume I (Chapitre 16). Description détaillée et exhaustive du jeu d'instructions.

*Internal Design Specification* Volume III. La manière dont le HP-71 utilise ses instructions...

## Mots-clés associés

`ADDR$`, `OPCODE$`, `PEEK$`, `POKE`

## Auteurs

Pierre David, Janick Taillandier

TEDIT (Text EDITor) est un éditeur de textes extrêmement rapide, compatible avec l'éditeur de textes Hewlett-Packard.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input checked="" type="checkbox"/> Opération d'unité
--	---

TEDIT <i>fichier</i> TEDIT <i>fichier</i> , <i>chaîne de commandes</i>
---

## Exemples

TEDIT ESSAI

Crée le fichier ESSAI et rentre sous l'éditeur de textes en mode commandes.

TEDIT ESSAI, "1#R/TOTO/TATA/;E"

Appelle l'éditeur de textes sur le fichier ESSAI, substitue toutes les occurrences de la chaîne TOTO par la chaîne TATA et sort de l'éditeur.

TEDIT ESSAI:TAPE

Copie le fichier ESSAI en mémoire et rentre sous l'éditeur de textes en mode commandes.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Spécificateur de périphérique en option.
chaîne de commandes	Expression alphanumérique. Défaut : Aucune commande n'est exécutée.	Aucune.

## Opération

TEDIT n'est inclus dans JPC Rom que pour des raisons de compatibilité, et son utilisation n'est conseillée que dans ce cas. TEDIT est un éditeur de textes compatible avec l'éditeur de textes EDTEXT inclus dans les modules Forth / Assembleur et Text Editor de Hewlett-Packard. Il est cependant conseillé d'utiliser XEDIT, plus agréable et plus puissant.

Le lecteur désirant utiliser TEDIT pour des raisons de compatibilité avec EDTEXT est invité à se reporter au manuel de ce dernier pour référence, puis au paragraphe **différences entre TEDIT et EDTEXT**.

Le lecteur désirant utiliser XEDIT et connaissant déjà EDTEXT ou TEDIT est invité à se reporter au paragraphe **différences entre TEDIT et XEDIT**, puis au manuel de XEDIT pour la référence.

### Différences entre TEDIT et EDTEXT

Ce paragraphe suppose que vous connaissez déjà le fonctionnement de EDTEXT. Si ce n'est pas le cas, nous vous conseillons d'apprendre directement le fonctionnement de XEDIT.

La principale différence entre TEDIT et EDTEXT est la rapidité. TEDIT est entièrement en assembleur et optimisé spécialement pour les grands fichiers, EDTEXT est pour une grande partie en Basic, et est donc très lent.

Les autres différences sont résumées ci-dessous.

#### Commandes ajoutées

TEDIT dispose de commandes supplémentaires :

- H (*Help*) : affiche une aide,
- J (*Join*) : réunit plusieurs lignes consécutives en une seule,
- Q (*Quit*) : équivalent à E (*Exit*), et
- X (*eXchange file*) : édite un autre fichier sans sortir de TEDIT.

Pour plus de détails sur ces commandes, voir le manuel de XEDIT.

#### Commandes différentes

Les commandes C (*Copy*) et M (*Move*) de EDTEXT n'autorisent pas la copie ou le déplacement d'un bloc de texte incluant la ligne courante, sauf si cette ligne est la première ou la dernière du bloc. Les commandes C et M de TEDIT ne sont sujettes à aucune restriction.

#### Fonctionnalités supplémentaires

# TEDIT (suite)

TEDIT diffère de EDTEXT par quelques détails de comportement, qui sont le plus souvent des ajouts motivés par une longue utilisation de EDTEXT :

- La touche [g][CMDS] active la pile de commandes, et permet d'utiliser les touches de curseur verticales ([↓], [↑], [g][↓] et [g][↑]), que ce soit en mode *commandes* ou en mode *saisie de texte*. EDTEXT ne le permet pas, ou ne le permet qu'au prix d'une manipulation hasardeuse.
- La touche [f][CONT] affiche le nom du fichier, la ligne courante et le nombre total de lignes du fichier. EDTEXT n'affiche que le nom du fichier.
- En mode commandes, les touches de curseur verticales ([↓], [↑], [g][↓] et [g][↑]) n'affichent pas la nouvelle ligne. EDTEXT affiche la nouvelle ligne. TEDIT permet ainsi de se déplacer plus vite dans le fichier, l'affichage d'une ligne étant maintenant demandé explicitement par l'utilisateur en appuyant sur [ENDLINE]
- La touche [RUN] est désactivée. Avec EDTEXT, elle agit comme la touche [ATTN], ce qui provoque souvent la perte d'une ligne de texte en cours de frappe.
- La touche [f][USER] est maintenant active, ce qui permet de valider ou non le mode *user*. EDTEXT, à cause de sa programmation, interdit de passer en dehors du mode *user* (ou ne le permet qu'au prix d'une manipulation hasardeuse).
- La sortie de TEDIT ne provoque pas l'affichage du message Done, car la sortie est instantanée.
- La touche [f][OFF] (ou une inactivité de plus de 10 minutes) sort de TEDIT. Avec EDTEXT, cela ne provoque que la perte de la ligne courante avec retour au mode *commandes*, sans possibilité ultérieure d'extinction.

## Différences entre TEDIT et XEDIT

Les éditeurs TEDIT et XEDIT sont très proches l'un de l'autre. L'objectif de TEDIT est de rester compatible avec EDTEXT, alors que l'objectif de XEDIT est d'offrir un environnement d'édition proche de celui de l'éditeur ed sous le système Unix (Unix est une marque déposée par AT&T). Ces deux objectifs ont conduit à des divergences entre TEDIT et XEDIT lorsqu'ils étaient incompatibles.

### Différences dans les chaînes génériques

Le principe des chaînes génériques sous EDTEXT est d'offrir deux modes : un mode *normal* et un mode *générique*. Par exemple, la chaîne A\@1\\$ signifie :

- chercher un caractère A en mode normal, puis
- passer en mode générique, puis
- chercher un caractère quelconque de caractères quelconques (caractère spécial @), puis
- chercher un caractère 1 (le caractère 1 n'est pas spécial), puis
- repasser en mode normal, et enfin
- chercher un caractère \$ (le caractère \$ est spécial, mais nous ne sommes plus en mode générique).

Sous XEDIT, on est en permanence en mode *générique*. Si on veut utiliser un caractère spécial comme un caractère normal (exemple du \$ ci-dessous), il faut le faire précéder par le caractère \. On passe donc en mode *normal* pour un et un seul caractère.

Hormis cette différence liée au mode, le caractère @ n'a pas de signification spéciale sous XEDIT (il est remplacé par la séquence . . \*). La table ci-dessous représente les caractères spéciaux utilisés lors des recherches avec XEDIT :

Caractère	Signification
-----	-----
\	annule la signification du caractère suivant
^	début de ligne
\$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^ ]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

L'exemple ci-dessus est donc A . . \*\\$ avec XEDIT.

### Différences dans le format des paramètres

Les paramètres des commandes de TEDIT sont de simples numéros de lignes. Avec XEDIT, ce sont de véritables expressions contenant :

- des numéros de lignes classiques,
- le symbole . pour désigner la ligne courante,

- le symbole \$ pour désigner la dernière ligne du fichier,
- des chaînes génériques.

Par exemple, pour détruire toutes les lignes entre la suivante et celle juste avant la prochaine occurrence d'une majuscule en début de ligne, on fera sous XEDIT :

```
.+1 , /^[A-Z]/-1 D
```

Les espaces ont été mis ici pour plus de clarté, ils ne sont bien évidemment pas nécessaires lorsque vous tapez la ligne. On comprend aisément, au vu de cet exemple, la différence de puissance entre TEDIT et XEDIT.

## Références

*A paraître* : Première version de TEDIT par Pierre David et Janick Taillandier.

*Manuel du module Editeur de Textes* par Hewlett-Packard.

*Manuel du module Forth / Assembleur* par Hewlett-Packard.

## Mots-clés associés

XEDIT

## Auteurs

Pierre David, Janick Taillandier



TOKEN renvoie le code interne (Lex ID et token) de la fonction spécifiée.

<input type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input checked="" type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

TOKEN ( <i>mot-clef</i> )
TOKEN ( <i>mot-clef</i> , <i>numéro d'ordre</i> )

## Exemple

A=TOKEN("TOKEN", 1)

Place 225073 dans la variable A, soit le token de la fonction TOKEN. L'Id est 225, et le token est 73.

## Paramètres d'entrée

Élément	Description	Restriction
mot-clef	Expression alphanumérique.	Le mot-clef doit exister.
numéro d'ordre	Expression numérique arrondie à un entier. Défaut : 1	Le mot-clef doit exister.

## Opération

TOKEN renvoie le code interne de la fonction spécifiée. Le code interne est le numéro affiché avec XWORD ou XFN quand le Lex auquel appartient la fonction n'est pas présent.

TOKEN renvoie le code interne sous la forme *iiittt*, formé par l'identificateur du Lex (Id) et du code proprement dit (token). Par exemple, TOKEN a comme token 73 dans Lex d'identificateur 225. Le résultat est donc  $225*1000+73$ , soit 225073.

S'il existe plusieurs mots-clef de même nom, le paramètre *numéro d'ordre* permet de les distinguer. Par défaut, *numéro d'ordre* vaut 1.

Si le mot-clef n'existe pas, ou s'il n'y a pas assez d'occurrences du mot-clef, l'erreur `ERR:Invalid Arg` est renvoyée.

Les mots-clefs de plus de 8 caractères nécessitent un traitement spécial par le système. Ainsi, des mots comme UNDERLINE ou RANDOMIZE sont reconnus comme UNDERLIN ou RANDOMIZ. Ce sont ces fonctions qui reconnaissent le E final. TOKEN ne peut connaître ces caractères supplémentaires. Pour les trouver, cette fonction reconnaît UNDERLIN et ignore le E final. Ainsi, TOKEN("RANDOMIZE") comme TOKEN("RANDOMIZ----") ignorent les caractères supplémentaires et renvoient le même numéro.

TOKEN identifie le plus long mot-clef dont le nom est une sous-chaîne du paramètre fourni. Par exemple, TOKEN("MEMORY") renvoie 1056, le numéro de la fonction MEM.

TOKEN ne reconnaît pas seulement les fonctions et mots-clefs, mais tous les éléments syntaxiques du Basic du HP-71. Ainsi, TOKEN(", ") est légal, ainsi que TOKEN("3"), etc.

TOKEN fournit un moyen rapide et pratique pour identifier d'éventuels conflits entre mots-clefs.

## Référence

JPC 31 (page 22) première version par Jean-Jacques Moreau.

Manuel du module Forth / Assembler (page 64).

Internal Design Specification, Volume I

## Mots-clés associés

ENTRY\$

## Auteur

Jean-Jacques Moreau

# UNDERLINE

UNDERLINE active ou désactive le mode « caractères soulignés » sur l'imprimante.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input type="checkbox"/> Opération d'unité
--	--

UNDERLINE ON UNDERLINE OFF
-------------------------------

## Exemples

UNDERLINE ON @ PRINT "HP - 71"

Imprime la chaîne "HP-71".

UNDERLINE OFF @ PRINT "HP - 71"

Imprime la chaîne "HP-71" sans rien souligner.

## Opération

UNDERLINE ON active le mode souligné sur le périphérique désigné par PRINTER IS.

UNDERLINE OFF repasse en mode normal. Cet ordre dépend du périphérique utilisé, et est destiné aux imprimantes HP répondant au *Printer Control Language*.

Note : UNDERLINE est destiné aux imprimantes qui reconnaissent le langage *PCL* de Hewlett-Packard, telles que la ThinkJet ou la LaserJet. La séquence d'échappement envoyée peut ne pas donner le résultat escompté sur d'autres imprimantes.

### Codes envoyés à l'imprimante :

UNDERLINE ON : ESC & d D

UNDERLINE OFF : ESC & d @

## Références

*JPC 26* (page 39) première version par Pierre David.

*JPC 40* (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

## Mots-clés associés

BELL, BOLD, MODE, PAGELEN, PCR, PFF, PLF, PRINT, PRINTER IS, WRAP

## Auteur

Pierre David

VARSWAP permute deux variables ou éléments de tableau.

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> Ordre<br><input type="checkbox"/> Fonction<br><input type="checkbox"/> Opérateur | <input checked="" type="checkbox"/> Exécution au clavier<br><input type="checkbox"/> Mode CALC<br><input checked="" type="checkbox"/> IF...THEN...ELSE<br><input checked="" type="checkbox"/> Opération d'unité |
|--|---|

VARSWAP *variable1* , *variable2*

## Exemples

VARSWAP A, B

Echange le contenu des deux variables numériques A et B.

VARSWAP A\$, B\$

Echange le contenu des deux variables alphanumériques A\$ et B\$.

## Paramètre d'entrée

Élément	Description	Restriction
variable1, variable2	Noms de variables numériques ou alphanumériques.	Les variables doivent être compatibles. Les tableaux entiers ne sont pas admis.

## Opération

VARSWAP permute les contenus des deux variables, pourvu qu'elles soient de même type (chaîne ou numérique) et de même dimension.

### Attention !

VARSWAP ne fonctionne pas lorsque la variable *variable2* n'existe pas et doit donc être créée par VARSWAP.

Ceci peut provoquer une réinitialisation complète de la mémoire.

## Références

*JPC 31* (page 50) première version par Jean-Jacques Moreau.

*A paraître* : deuxième version par Pierre David et Janick Taillandier.

L'ancienne appellation de VARSWAP était SWAP.

## Mots-clés associés

LET

## Auteurs

Pierre David, Jean-Jacques Moreau, Janick Taillandier

# WHILE ... END WHILE

WHILE ... END WHILE définit une boucle qui est exécutée tant que l'expression booléenne dans le WHILE est vraie.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

```
WHILE expression logique
  segment de programme
END WHILE
```

## Exemple

```
10 WHILE I+2<=N
20   CALL AA(I,N)
30   I=I+1
40 END WHILE
```

Répète l'appel au sous programme AA tant que la condition est vraie.

## Paramètres d'entrée

Élément	Description	Restriction
expression logique	Expression numérique évaluée comme vraie si non nulle.	Aucune.
segment de programme	Tout nombre de lignes de programme contiguës.	Aucune.

## Opération

La construction WHILE ... END WHILE autorise la répétition conditionnelle d'un segment de programme. Si la condition est vraie, le segment compris entre le WHILE et le END WHILE est exécuté, et un branchement est fait sur le WHILE.

Le segment de programme est répété jusqu'à ce que le test soit faux. Quand cela arrive, le segment de programme est sauté, et l'exécution reprend à la ligne suivant le END WHILE.

Le segment de programme peut ne jamais être exécuté, si l'expression est évaluée comme fausse la première fois.

Tout comme pour les boucles de type LOOP ... END LOOP ou REPEAT ... UNTIL, l'ordre LEAVE permet de sortir prématurément d'une boucle WHILE ... END WHILE.

Le segment de programme peut lui-même contenir des structures de type boucle ou choix. L'emboîtement doit être correct, sinon l'erreur JPC ERR:Structure Mismatch est renvoyée.

## Références

JPC 31 (page 38) première version par Janick Taillandier.

JPC 52 (page 10) deuxième version par Pierre David et Janick Taillandier.

Basic 5.0 pour HP 9000 séries 200/300.

## Mots-clés associés

LEAVE, LOOP ... END LOOP, REPEAT ... UNTIL

## Auteurs

Pierre David, Janick Taillandier

WRAP active ou désactive le mode «saut à la ligne» sur l'imprimante.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input type="checkbox"/> Opération d'unité
WRAP ON WRAP OFF	

## Exemple

WRAP ON @ PBLIST

Liste le programme courant sur l'imprimante, en passant à la ligne quand le chariot de l'imprimante arrive en fin de course.

## Opération

Le mode «saut à la ligne» est utilisé quand le HP-71 doit imprimer des lignes plus longues que la longueur admissible par l'imprimante. Celle-ci passe automatiquement à la ligne suivante quand le chariot arrive en bout de course.

Le HP-71 dispose déjà de cette possibilité avec l'ordre PWIDTH. Cependant, WRAP est géré par l'imprimante, donc décharge le HP-71 de cette gestion. D'autre part, le HP-71 compte les éventuelles séquences d'échappement envoyées avec PRINT dans ses calculs, ce qui induit des erreurs. L'utilisation de WRAP donne des résultats plus exacts, sans faire intervenir le HP-71.

### Codes envoyés à l'imprimante :

WRAP ON : ESC & s 0 C

WRAP OFF : ESC & s 1 C

## Références

JPC 26 (page 39) première version par Pierre David.

JPC 40 (page 16) deuxième version par Pierre David.

Consultez aussi le manuel de votre imprimante.

## Mots-clés associés

BOLD, ENDLINE, ESC\$, PRINT, PRINTER IS, UNDERLINE

## Auteur

Pierre David

# WREC

WREC (Write RECORD) écrit une chaîne de 256 caractères sur le secteur spécifié du support de mémoire de masse indiqué.

<input checked="" type="checkbox"/> Ordre <input type="checkbox"/> Fonction <input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> Exécution au clavier <input type="checkbox"/> Mode CALC <input checked="" type="checkbox"/> IF...THEN...ELSE <input type="checkbox"/> Opération d'unité
--	--

WREC <i>secteur</i> , <i>adresse</i> , <i>spécificateur de périphérique</i>
---

## Exemple

WREC A\$,1, :TAPE

Écrit la chaîne A\$ (256 caractères) dans l'enregistrement numéro 1 du support indiqué par :TAPE.

## Paramètres d'entrée

Élément	Description	Restriction
secteur	Expression alphanumérique.	La longueur doit valoir exactement 256 caractères.
adresse	Expression numérique arrondie à un entier, ou expression alphanumérique contenant des nombres hexadécimaux, représentant un numéro de secteur sur le support.	Valeur comprise entre 0 et la taille maximum du support.
spécificateur de périphérique	Expression numérique ou alphanumérique. Voir les définitions standard HPIL.	Ne peut pas être une chaîne non comprise entre guillemets.

## Opération

WREC écrit une chaîne de 256 octets représentant un secteur, ou enregistrement sur le support indiqué par *spécificateur de périphérique*.

L'adresse est un numéro de secteur en décimal, ou une chaîne représentant un numéro hexadécimal de secteur.

### Interruption par [ATTN] :

Cette fonction est interruptible par deux appuis successifs sur la touche [ATTN]. Le HP-71 affiche alors le message d'erreur HPIL ERR: Abort ed. Il peut être nécessaire d'exécuter RESTORE IO pour réactiver le système HP-IL.

## Références

JPC 45 (page 15) première version par Michel Martinet.

Manuel d'utilisation du lecteur de cassettes numériques HP-82161A.

Manuel du module HP-IL pour HP-71, chapitre 3 et Annexe D.

## Mots-clés associés

RREC\$, OUTPUT

## Auteur

Michel Martinet

XAXIS trace un axe horizontal avec des marques de graduation optionnelles, à l'ordonnée  $y$  spécifiée.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

XAXIS *ordonnée y*  
 XAXIS *ordonnée y, espacement*  
 XAXIS *ordonnée y, espacement, xmin, xmax*

## Exemples

XAXIS 10

Trace un axe à l'ordonnée 10.

XAXIS (Y2-Y1)/2, -5

Trace un axe et des marques de graduation espacées de 5 unités graphiques, en commençant par la droite.

## Paramètres d'entrée

Élément	Description	Restriction
ordonnée $y$	Expression numérique arrondie à un entier.	-32767 à +32767
espacement	Expression numérique arrondie à un entier. Défaut : 0, pas de marques.	-32767 à +32767
$xmin$	Expression numérique arrondie à un entier. Défaut : 0	-32767 à +32767
$xmax$	Expression numérique arrondie à un entier. Défaut : 639	-32767 à +32767

## Opération

XAXIS trace un axe horizontal à l'ordonnée  $y$ , de  $xmin$  à  $xmax$ .

Si le paramètre *espacement* est présent et non nul, des marques sont tracées toutes les *espacement* unités graphiques, en commençant par  $xmin$  si *espacement* est positif, par  $xmax$  si *espacement* est négatif.

Les longueurs des marques de graduation sont définies par TICLEN. Après exécution de GINIT, elles valent 4.

## Mots-clés associés

LINETYPE, TICLEN, YAXIS

## Auteur

Pierre David

# XEDIT

XEDIT (eXtended EDITor) est un éditeur de textes étendu.

<input checked="" type="checkbox"/> Ordre	<input checked="" type="checkbox"/> Exécution au clavier
<input type="checkbox"/> Fonction	<input type="checkbox"/> Mode CALC
<input type="checkbox"/> Opérateur	<input checked="" type="checkbox"/> IF...THEN...ELSE
	<input checked="" type="checkbox"/> Opération d'unité

XEDIT <i>fichier</i>
XEDIT <i>fichier</i> , <i>chaîne de commandes</i>

## Exemples

XEDIT ESSAI

Crée le fichier ESSAI et rentre sous l'éditeur de textes en mode commandes.

XEDIT ESSAI, "1, \$R/TOTO/TATA/;E"

Appelle l'éditeur de textes sur le fichier ESSAI, substitue toutes les occurrences de la chaîne TOTO par la chaîne TATA et sort de l'éditeur.

XEDIT ESSAI:TAPE

Copie le fichier ESSAI en mémoire et rentre sous l'éditeur de textes en mode commandes.

## Paramètres d'entrée

Élément	Description	Restriction
fichier	Expression alphanumérique ou chaîne sans guillemets.	Spécificateur de périphérique en option.
chaîne de commandes	Expression alphanumérique. Défaut : Aucune commande n'est exécutée.	Aucune.

## Opération

### Qu'est-ce qu'un éditeur de textes ?

Un *éditeur de textes* est un programme permettant de rentrer des caractères dans un fichier TEXT du HP-71. Hewlett-Packard fournit un éditeur de textes dans les modules *Forth/Assembleur* d'une part, et *Text Editor* d'autre part.

Le module *Text Editor* comprend en outre un *formatteur de textes*, c'est à dire un programme pour aligner des paragraphes, centrer des lignes, etc. De tels programmes ont été publiés dans *JPC*. Un formatteur de textes opère sur des fichiers qu'il a donc fallu introduire dans le HP-71 à l'aide d'un éditeur de textes.

EDTEXT, l'éditeur de textes présent dans les deux modules de Hewlett-Packard, est rédigé en majeure partie en Basic, et souffre d'une lenteur caractérisée.

*JPC Rom* contient un éditeur de textes entièrement en assembleur, extrêmement rapide et dont les performances ne se dégradent pas, même avec de très grands fichiers.

### Entrée sous l'éditeur

Lorsque vous exécutez l'ordre XEDIT suivi d'un nom de fichier, le fichier est soit :

- créé s'il n'existait pas,
- copié en mémoire s'il réside sur un support magnétique,
- ou simplement cherché en mémoire,

puis il est analysé pour vérifier s'il est bien du type TEXT, enfin l'éditeur passe en mode *commandes* (voir chapitre suivant).

Notez que l'indicateur 1 disparaît de l'affichage (le drapeau 1 conserve toujours son état en mémoire, seul l'affichage est affecté).

L'éditeur détermine en outre si votre fichier est modifiable, c'est à dire s'il n'est ni en Rom ou Eprom, pas sécurisé par SECURE, et pas déjà ouvert par ASSIGN #. Si votre fichier n'est pas modifiable, vous pouvez le visualiser, mais toute tentative de modification déclenchera l'erreur correspondant au type de protection.

Vous sortez de l'éditeur par la commande E ou Q. Si vous étiez en mode *saisie*, vous sortez en appuyant sur la touche [ATTN], puis en tapant une de ces deux commandes. Si vous ne savez pas en quel mode vous êtes, appuyez sur [ATTN], puis tapez la commande E ou Q.

Si vous aviez spécifié une chaîne de commandes, celle-ci aurait été exécutée automatiquement, sans passer en mode

*commandes*. Lorsque vous utilisez une chaîne de commandes, faites attention toutefois à la terminer par la commande E ou Q pour sortir de l'éditeur, faute de quoi l'éditeur attendra une nouvelle commande.

## Les modes d'opération

L'éditeur possède deux modes, le mode *commandes* et le mode *saisie*.

### Le mode commandes

Lorsque vous rentrez sous l'éditeur, vous êtes placés en mode *commandes*. L'éditeur vous le signale en affichant un message tel que :

Line *nnn*, Cmd: ou bien

Eof, Cmd:

et en plaçant le curseur à droite des caractères.

L'éditeur attend alors que vous tapiez une commande selon le format décrit dans le chapitre suivant.

Si vous laissez votre doigt appuyé sur la touche :

- [ENDLINE], la ligne est affichée, sans changer la ligne courante,
- [f][CONT], l'éditeur affiche le nom du fichier courant, le numéro de la ligne courante et le nombre de lignes dans le fichier.

Vous pouvez, en outre, utiliser les touches de curseur verticales pour passer à la ligne suivante (touche [↓]), revenir à la ligne précédente (touche [↑]), aller directement à la dernière (touche [g][↓]) ou à la première (touche [g][↑]) ligne du fichier.

Vous pouvez aussi choisir d'utiliser la pile de commandes en appuyant sur [g][CMDS], de la même manière qu'en utilisation normale.

Toutes les touches d'édition (touches de curseur horizontales, [f][BACK], [f][-CHAR], etc.) sont utilisables pour faciliter l'introduction de vos commandes.

### Le mode saisie

Vous pouvez choisir de passer en mode *saisie* (saisie de texte avec les commandes T (*text*) ou I (*insert*)). Tout ce que vous introduisez est alors rentré dans votre fichier, jusqu'à ce que vous appuyiez sur la touche [ATTN] qui vous replace en mode commandes.

La touche [f][CONT] est là aussi active, ce qui vous permet de connaître à tout moment le nom du fichier en cours d'édition, le numéro de la ligne courante et le nombre de lignes dans le fichier.

La touche [f][VIEW] est aussi active (en mode USER seulement) et vous donne la position du curseur dans la ligne.

La pile de commandes est accessible à tout moment, toujours en appuyant sur [g][CMDS].

### Inactivité de plus de dix minutes

L'éditeur détecte les périodes d'inactivité de plus de dix minutes et :

- si vous étiez en mode *saisie*, repasse en mode *commandes*,
- si vous étiez en mode *commandes*, sort de l'éditeur,

Ceci assure l'extinction automatique de votre HP-71B au bout de 20 ou 30 minutes suivant le mode dans lequel vous étiez.

## Syntaxe des commandes

Les commandes que vous introduisez ont la syntaxe suivante :

[ *début* [[ , ] *fin* ] ] [ ? ] [ *commande* ] [ *paramètre* ]

où les éléments entre crochets sont optionnels. Certaines commandes peuvent restreindre leur syntaxe : par exemple, la commande T n'admet pas de *fin*, pas de point d'interrogation, pas de *paramètre*.

Les éléments *début* et *fin* sont des numéros de ligne spécifiant un intervalle bornes comprises (si les deux sont présents), une simple ligne (si *début* est seul) ou la ligne courante (si aucun n'est présent). Pour plus de précision, voir les valeurs par défaut pour chacune des commandes.

Ces éléments sont en fait de véritables expressions pouvant faire jouer :

- la ligne courante (symbole .),
- la dernière ligne du fichier (symbole \$),
- une constante numérique, ou
- la prochaine occurrence d'un certain motif (chaîne encadrée par /).

Par exemple, l'expression .+5, /exemple/-3 signifie : de la ligne courante plus cinq lignes jusqu'à la prochaine occur-

# XEDIT (suite)

rence de la chaîne *exemple* moins trois lignes.

Le point d'interrogation (utilisé par les commandes S, R et J) spécifie qu'une commande doit être exécutée en mode *confirmation*, c'est à dire qu'une question vous est posée, pour la commande R par exemple, à chaque remplacement.

La *commande* est une simple lettre. Il n'y a pas de distinction entre majuscule et minuscule. Le cas spécial où la commande est vide implique le positionnement sur une ligne. Par exemple, pour vous positionner sur la ligne 523, vous taperez simplement 523, puis [ENDLINE]. Vous pouvez aussi vous rendre directement à la prochaine occurrence de la chaîne *exemple* en tapant /*exemple* (notez que le / final peut être omis ici), puis [ENDLINE].

Le *paramètre* est enfin un paramètre optionnel de la commande. Par exemple, il permet à la commande H (Help) de savoir si vous désirez l'aide sur toutes les commandes, ou sur une commande particulière.

Vous pouvez placer plusieurs commandes sur la même ligne, séparées par le caractère ; (point-virgule). Elles seront alors exécutées en séquence, sauf si l'une d'entre-elles s'interrompt à cause d'une erreur. Cette facilité de regrouper des commandes est très intéressante lorsque vous appelez XEDIT avec une chaîne de commandes.

## Chaînes génériques

Le terme *chaîne générique* est utilisé pour définir les chaînes avec ou sans caractère spécial utilisées dans les recherches ou les remplacements avec XEDIT et les fonctions associées (FILEPOS, FIND, GENLEN, GENPOS et GENRPLC\$).

Ces chaînes génériques sont compatibles avec les chaînes génériques sous Unix (Unix est une marque déposée par AT&T), et plus particulièrement avec l'éditeur ed d'Unix.

### Chaînes génériques de recherche

Les chaînes génériques sont le plus souvent utilisées lors des recherches de chaînes dans le fichier.

Pour chercher une suite de caractères standards, il n'y a aucun problème. Par contre, les choses se compliquent dès que vous avez des besoins un peu plus compliqués, comme par exemple chercher un chiffre quelconque, une majuscule en début de ligne, un a et un b sur la même ligne, etc. Il faut alors que vous insériez dans votre chaîne de recherche des caractères spéciaux pour représenter ces cas.

La table ci-dessous décrit les caractères spéciaux utilisés dans les chaînes de recherche :

Caractère	Signification
-----	-----
\	annule la signification du caractère suivant
^	début de ligne
\$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^ ]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

- Le caractère \ (back-slash) est le caractère spécial des caractères spéciaux ! Si vous voulez chercher un point, par exemple, cela est impossible puisque c'est un caractère spécial et XEDIT le prendra comme tel. Il faut donc annuler la signification spéciale du point, pour ne plus le considérer que comme un caractère standard. Pour chercher un point, la chaîne générique sera donc : "\." (les guillemets ne font pas partie de la chaîne, et sont là simplement pour la lisibilité).

- Le caractère ^ signifie, lorsqu'il est placé au début d'une chaîne de recherche, *début de ligne*. Ceci vous permet, par exemple, de chercher toto en début de ligne : vous utiliserez la chaîne générique : "^toto".

- Le caractère \$ signifie, lorsqu'il est placé à la fin d'une chaîne de recherche, *fin de ligne*. Ceci vous permet, par exemple, de chercher tata en fin de ligne : vous utiliserez la chaîne générique : "tata\$".

- Le caractère . signifie *n'importe quel caractère*. Pour chercher un x en deuxième position sur une ligne, vous utilisez la chaîne générique : ".^x".

- Les caractères [ et ] servent à chercher un caractère dans un ensemble. Un ensemble ne peut trouver qu'un et un seul caractère. Il existe plusieurs types d'ensembles : les ensembles par énumération et les intervalles.

Un ensemble par énumération est, par exemple, la chaîne générique "[0123456789]" : elle représente un (et un seul) chiffre. L'ensemble est défini comme l'ensemble de tous les caractères entre les crochets.

Un ensemble de type intervalle est, par exemple, la chaîne générique "[0-9]", qui représente la même chose que ci-dessus, mais de manière plus condensée. Un tel ensemble est défini par tous les caractères de code ASCII compris entre les deux bornes.

La séparation entre ces deux types d'ensembles n'est pas si rigide, et les deux types peuvent cohabiter dans un seul ensemble. Par exemple, pour identifier un caractère qui peut être une lettre (minuscule ou majuscule) ou un chiffre impair, on utilisera la chaîne générique suivante : "[A-Za-z13579]". Il y a là deux intervalles et une énumération dans le même ensemble.

- Lorsqu'un caractère ^ est le premier caractère dans un ensemble, cela signifie que cet ensemble ne trouvera pas un

caractère appartenant à cet ensemble, mais au contraire un caractère *n'appartenant pas* à cet ensemble. Par exemple, pour trouver un caractère qui ne soit pas une lettre, on utilisera la chaîne générique : "[^A-Za-z]".

- Le caractère \*, quant à lui, ne représente pas un caractère, mais la répétition 0 ou n fois du *motif précédent*. Par exemple, la combinaison "[A-Z]\*" représente 0 ou n occurrences d'une majuscule. Le caractère \* est indissociable du motif précédent. D'autre part, remarquez que la répétition est faite 0 ou n fois. Ceci signifie, par exemple, que "X\*" sera trouvé, même dans une chaîne ne contenant pas de X. Si vous désirez trouver au moins un X, il faudra utiliser "X\*", ce qui signifiera : trouver le caractère X suivi du caractère X 0 ou n fois, donc trouver le caractère X 1 ou n fois.

Vous pouvez combiner ces caractères spéciaux entre eux ou avec des caractères normaux comme bon vous semble, pour arriver au résultat.

Voici quelques exemples, pour vous familiariser avec ces chaînes génériques :

- une référence à TEDIT ou XEDIT :

[TX]EDIT

- une ligne vide :

^\$

- une ligne se terminant par des espaces :

\*\$ note : la chaîne commence par deux espaces

- une phrase ne commençant pas par une majuscule :

\. \*[^A-Z] note : ne traite pas les phrase en début de ligne

- un nombre négatif :

-[0-9][0-9]\* note : au moins un chiffre

- un nombre hexadécimal en assembleur HP-71 :

#[0-9A-F][0-9A-F]\*

- une affectation à un tableau numérique en basic HP-71 :

[^\$](^[^])\*=

## Chaînes génériques de remplacement

Lorsque vous désirez effectuer un remplacement, vous spécifiez une chaîne de recherche, délimitant le ou les motifs que vous voulez remplacer, puis une chaîne de remplacement qui spécifie par quoi vous allez remplacer votre motif. La chaîne de recherche est une de celles vues plus haut.

Ceci permet, par exemple, de remplacer toutes les suites d'espaces par un seul (c'est à dire remplacer toutes les occurrences de " \*" (deux espaces) par " " (un seul espace)), et ainsi de simuler le comportement de la fonction REDUCE\$ de JPC Rom).

Mais la chaîne de remplacement peut également contenir des caractères spéciaux, résumés dans la table ci-dessous :

Caractère	Signification
\	annule la signification du caractère suivant
&	substitue l'occurrence trouvée

- Le caractère \ a la même signification que précédemment,

- Le caractère & (*et commercial*) signifie : substituer par l'occurrence trouvée.

Ainsi, si on veut mettre un signe - devant tous les nombres, on remplacera "[0-9][0-9]\*" par "-&".

Si on veut remplacer toutes les occurrences d'une majuscule par la même répétée deux fois, on remplacera "[A-Z]" par "&&".

## Affectation du back-slash

Comme vous avez pu le constater, le caractère \ (*back-slash*) a une certaine importance sous XEDIT. Nous vous conseillons de l'affecter définitivement à une touche. L'éditeur de textes EDTEXT de Hewlett-Packard l'affectait à la touche [f][/] ce qui nous semble un bon choix. Vous pouvez donc faire DEF KEY "f/", CHR\$(92) ; pour l'affecter à cette touche.

## Limitations des chaînes génériques

Les chaînes génériques de recherche sont sujettes à des limitations du fait de la ressource interne utilisée (les buffers du système sont limités à 2 Ko) : il ne peut y avoir plus de 14 ensembles dans un chaîne générique. Sachez en outre qu'un ensemble occupe 128 octets.

Une autre limitation vient de l'algorithme utilisé pour traiter les \*. Le cas pathologique est la recherche, par exemple, du

# XEDIT (suite)

motif . \*xx (où xx n'existe pas dans votre fichier), qui peut devenir nettement plus longue que prévu.

## Traduction des messages et commandes

Vous pouvez utiliser un traducteur de messages pour traduire les messages de XEDIT ainsi que ses commandes.

## Utilisation de la mémoire

XEDIT, pour ses besoins propres, nécessite de la mémoire disponible dans votre HP-71. Cette mémoire est libérée dès que vous sortez de l'éditeur.

XEDIT, à l'initialisation, cherche à créer une table d'accès rapide aux lignes de votre fichier. Pour cela, l'éditeur peut consommer jusqu'à 2 Ko si votre fichier est très gros. Très exactement, il occupe 3,5 octets + 5 octets toutes les 200 lignes. Si votre fichier fait plus de 80000 lignes (environ), XEDIT s'arrêtera à 2 Ko. Si vous ne disposez pas de la place suffisante en mémoire, XEDIT prendra ce qui est disponible.

A chaque fois que vous faites une recherche, XEDIT alloue de la mémoire pour stocker la chaîne générique de recherche. La place occupée est 4 octets + 0,5 octets pour les caractères ^, \$, . et \*, 1,5 octets pour les caractères normaux et 128,5 octets pour les ensembles. S'il n'y a pas assez de mémoire, XEDIT affiche l'erreur No Room for Pattern.

Lors d'une commande C, M ou D dans un fichier externe, 256 octets au moins sont requis pour mener à bien l'opération dans des conditions de rapidité convenables. Plus de mémoire disponible accélère notablement le transfert.

## Note à propos des grands fichiers

XEDIT crée une table d'accès rapide aux lignes de votre fichier, ce qui lui évite de parcourir tout le fichier depuis le début à chaque fois que vous faites la moindre opération. Cette création est faite lorsque vous rentrez sous XEDIT.

Toutefois, il peut arriver qu'au cours d'une séance d'édition, votre fichier s'amenuise ou grandisse. Lorsque XEDIT le jugera bon, il décidera de recalculer cette table d'accès rapide, ce qui peut se traduire par un ralentissement d'autant plus important que votre fichier est gros. Ceci dit, cette opération est rarement perceptible, le recalcul dépassant rarement l'ordre de la seconde.

## Commande nulle

La commande nulle a comme seul effet de déplacer la ligne courante.

### Syntaxe

*ligne*

### Valeurs par défaut

*aucune*

### Opération

La commande nulle a comme seul effet de déplacer la ligne courante. En profitant de l'utilisation des chaînes génériques dans les numéros de lignes, cela permet de faire une recherche très simplement. Par exemple, pour se déplacer à la prochaine occurrence de la chaîne toto, faire /toto, puis [ENDLINE].

## Commandes Exit et Quit

Les commandes E (*Exit*) et Q (*Quit*) sortent de XEDIT.

### Syntaxes

E

Q

### Valeurs par défaut

*Aucune*

### Opération

Les commandes E et Q sont deux synonymes servant à sortir de l'éditeur. Aucun message n'est affiché, le curseur Basic revient à l'affichage, et l'état initial du drapeau 1 est restauré.

### Exemples

E                    sort de l'éditeur

Q                    sort de l'éditeur

## Commande Help

La commande H (*Help*) affiche la syntaxe d'une ou de plusieurs commandes.

### Syntaxe

H [ *commande* ]

## Valeurs par défaut

*commande* = toutes les commandes

## Opération

Si vous spécifiez une commande derrière H, XEDIT affiche la syntaxe de la commande spécifiée. Toute pression sur une touche ramène l'affichage en mode *commandes*.

Si vous ne spécifiez aucune commande, XEDIT affiche la syntaxe de toutes les commandes par ordre alphabétique. Vous passez d'une commande à l'autre par pression sur les touches de curseur verticales ([↓], [↑], [g][↓] et [g][↑]). Toute autre touche ramène l'affichage en mode *commandes*.

## Exemples

H                    affiche la syntaxe de toutes les commandes, une par une  
H H                  affiche la syntaxe de la commande H

## Commandes Text et Insert

Les commandes T (*Text*) et I (*Insert*) rentrent en mode *saisie*.

### Syntaxes

[ *ligne* ] T

[ *ligne* ] I

### Valeurs par défaut

*ligne* = ligne courante

### Opération

La commande T est la commande d'introduction de texte. Lorsque vous entrez en mode *saisie* avec la commande T, la ligne courante est affichée, et vous pouvez la modifier. Lorsque vous appuyez sur la touche [ENDLINE], cette ligne est introduite dans le fichier.

La commande I est la commande d'insertion de lignes de texte. Lorsque vous entrez en mode *saisie* avec la commande I, l'indicateur 1 s'allume pour signaler que vous êtes en mode insertion, et la ligne courante est affichée. Lorsque vous tapez du texte et que vous appuyez sur la touche [ENDLINE], le texte sera inséré dans le fichier juste avant la ligne que vous avez vue.

Lorsque vous êtes en mode *saisie*, vous pouvez vous déplacer avec les touches de curseur horizontales sur la même ligne, ou sur une autre ligne avec les touches verticales.

Pour sortir du mode *saisie*, appuyez sur la touche [ATTN].

### Exemples

T                    rentre en mode *saisie* sur la ligne courante  
\$I                  rentre en mode *saisie* avant la dernière ligne  
/toto/T            rentre en mode *saisie* sur la prochaine occurrence de toto

## Commandes List et Print

Les commandes L (*List*) et P (*Print*) listent les lignes sur l'appareil de visualisation ou d'impression.

### Syntaxes

[ *ligne début* [ *ligne fin* ] ] L [ *nombre de lignes* ] [ N ]

[ *ligne début* [ *ligne fin* ] ] P [ *nombre de lignes* ] [ N ]

### Valeurs par défaut

*ligne début* = ligne courante

*ligne fin* = dernière ligne

### Opération

Les commandes L et P sont très similaires (autant que LIST et PLIST en Basic).

Après le listage, la ligne courante devient la dernière ligne listée plus une. Le *nombre de lignes* est optionnel (et est incompatible avec *ligne fin*). Il indique le nombre de lignes à lister. Le caractère N optionnel provoque l'apposition du numéro de ligne devant chaque ligne listée.

### Exemples

3, 5L              liste les lignes 3 à 5 et la ligne 6 devient la ligne courante  
LN                 liste toutes les lignes avec leur numéro jusqu'à la fin

# XEDIT (suite)

.3,\$-5LN liste avec numéros de lignes  
5L10N liste 10 lignes à partir de la ligne 5

## Commande Delete

La commande D (*Delete*) détruit des lignes dans le fichier.

### Syntaxe

[ *ligne début* [ *ligne fin* ] ] D [ *fichier* [ + ] ]

### Valeurs par défaut

*ligne début* = ligne courante

*ligne fin* = *ligne début*

### Opération

La commande D détruit une ou plusieurs lignes. Si plus d'une ligne doit être détruite, confirmation est demandée avec le message `Ok to Delete? Y/N:`. Toute pression sur une touche autre que [Y] annule l'opération.

Il est possible de spécifier que les lignes détruites doivent être placées dans le fichier *fichier*. Si celui-ci n'existe pas, il est créé. S'il existe, une erreur est générée. Si vous voulez ajouter les lignes détruites à un fichier existant, faites suivre son nom par +.

### Exemples

D détruit la ligne courante

\$-5,\$D détruit les 6 dernières lignes du fichier

D EXEMPLE1 détruit la ligne courante et la met dans le fichier EXEMPLE1

D EXEMPLE2 + détruit la ligne courante et l'ajoute au fichier EXEMPLE2

## Commandes Copy et Move

Les commandes C (*Copy*) et M (*Move*) copient ou déplacent une ou plusieurs lignes du fichier.

### Syntaxes

[ *ligne début* [ *ligne fin* ] ] C [ *fichier* ]

[ *ligne début* [ *ligne fin* ] ] M [ *fichier* ]

### Valeurs par défaut (pas de *fichier* spécifié)

*ligne début* = ligne courante

*ligne fin* = *ligne début*

### Valeurs par défaut (*fichier* spécifié)

*ligne début* = première ligne du fichier spécifié

*ligne fin* = dernière ligne du fichier spécifié

### Opération

La commande C copie une plusieurs lignes et les insère avant la ligne courante. La commande M est similaire à la commande C, mais détruit les lignes originales.

Si aucun fichier n'est spécifié, les lignes insérées viennent du fichier édité. Si un fichier est spécifié et qu'il existe, ces lignes viendront de ce fichier.

Il faut un minimum de 256 octets disponibles pour déplacer des lignes de texte. De manière générale, plus grande est la mémoire disponible et plus rapide est le déplacement.

Attention à l'utilisation des expressions (., \$, recherches) dans les paramètres de ces commandes lorsqu'elles sont utilisées avec un fichier externe : les numéros de lignes font références au fichier externe, et les expressions sont évaluées avec le fichier courant. Pour cette raison, n'utilisez que des nombres pour vos numéros de lignes.

### Exemples

C duplique la ligne courante

5,7C duplique les lignes 5 à 7 et les insère avant la ligne courante

5,\$-1M déplace les lignes 5 à l'avant-dernière ligne du fichier

5,7C EXEMPLE1 copie les lignes 5 à 7 du fichier EXEMPLE1 devant la ligne courante

20M EXEMPLE2 copie les lignes 20 jusqu'à la fin de EXEMPLE2 et les insère dans le fichier courant

## Commande Join

La commande J (*Join*) réunit deux (ou plus) lignes du fichier en une seule.

## Syntaxe

[ *ligne début* [ *ligne fin* ] ] [ ? ] J [ *nombre de lignes* ]

## Valeurs par défaut

*ligne début* = ligne courante

*ligne fin* = *ligne début* + 1

## Opération

La commande J réunit les lignes spécifiées en une seule. Attention toutefois : le HP-71 ne permet pas l'édition de lignes supérieures à 96 caractères !

Un espace est ajouté avant chaque ligne jointe.

Si vous spécifiez un *nombre de lignes*, vous ne pouvez pas spécifier de *ligne fin*.

Vous pouvez contrôler finement la réunion des lignes en utilisant le caractère ? avant J. Chaque tentative de réunion provoquera l'affichage du numéro de ligne, d'un :, du numéro de colonne auquel s'effectue la réunion, d'un \, de la ligne au point précis de la réunion, d'un / et d'un point d'interrogation indiquant que XEDIT attend une réponse.

Vous pouvez alors répondre par :

- [Y] pour accepter la réunion et passer éventuellement à la suivante,
- [N] ou [Q] pour refuser la réunion et revenir en mode *commandes*.

Pour faire l'opération inverse de la commande J, c'est à dire séparer une ligne en deux, vous pouvez copier la ligne courante par C, puis éditer la ligne courante avec T, détruire du point de césure jusqu'à la fin, puis descendre d'une ligne et détruire jusqu'au point de césure.

## Exemples

J                   réunit la ligne courante et la suivante en une seule  
5, 7J               réunit les lignes 5 à 7 en une seule ligne 5  
J5                   réunit 5 lignes à partir de la ligne courante  
./toto/?J réunit les lignes spécifiées en demandant confirmation

## Commande Search

La commande S (*Search*) cherche un motif dans le fichier.

## Syntaxe

[ *ligne début* [ *ligne fin* ] ] [ ? ] S/ *motif* [ / ]

## Valeurs par défaut

*ligne début* = ligne courante + 1

*ligne fin* = dernière ligne

## Opération

La commande S cherche dans le fichier le motif spécifié à l'aide d'une chaîne générique (voir le paragraphe *chaînes génériques de recherche*). Si le motif est trouvé, la ligne trouvée devient la ligne courante.

Le caractère délimiteur de fin est optionnel.

Si vous désirez chercher une chaîne contenant un caractère /, spécifiez un autre caractère délimiteur après le S. Tout caractère non blanc peut convenir.

Vous pouvez contrôler finement la recherche en utilisant le caractère ? avant S. Chaque occurrence trouvée provoquera l'affichage du numéro de ligne, d'un :, du numéro de colonne trouvé, d'un \, de la partie de la ligne contenant l'occurrence, d'un / et d'un point d'interrogation indiquant que XEDIT attend une réponse.

Vous pouvez alors répondre par :

- [Y] pour accepter l'occurrence trouvée et revenir en mode commande avec cette ligne comme ligne courante,
- [N] pour refuser cette occurrence et continuer la recherche, en affichant un message d'erreur si toutes les occurrences ont été passées,
- [Q] pour quitter la recherche et revenir en mode *commandes*.

Note : pour effectuer une recherche simple, il est plus facile de taper directement /*motif* puis [ENDLINE] que d'utiliser la commande S, réservée pour les utilisations « lourdes ».

## Exemples

1S/toto           recherche la chaîne toto dans tout le fichier

# XEDIT (suite)

1, \$-10S/^\$      recherche une ligne vide avant la dixième avant la fin  
?S:1/4:          recherche 1/4 dans le fichier (caractère délimiteur = :)

## Commande Replace

La commande R (*Replace*) remplace une chaîne par une autre dans le fichier.

### Syntaxe

[ *ligne début* [ *ligne fin* ] ] [ ? ] R/ *motif* / *remplacement* [ / ]

### Valeurs par défaut

*ligne début* = ligne courante

*ligne fin* = *ligne début*

### Opération

La commande R cherche dans le fichier le motif spécifié à l'aide de chaîne génériques (voir le paragraphe *chaînes génériques de recherche*). Si le motif est trouvé, l'occurrence trouvée est remplacée par la chaîne de remplacement (voir le paragraphe *chaînes génériques de remplacement*).

Le caractère délimiteur de fin est optionnel.

Si vous désirez chercher ou remplacer une chaîne contenant un caractère /, spécifiez un autre caractère délimiteur après le R. Tout caractère non blanc peut convenir.

Vous pouvez contrôler finement le remplacement en utilisant le caractère ? avant R. Chaque occurrence trouvée provoquera l'affichage du numéro de ligne, d'un :, du numéro de colonne trouvé, d'un \, de la partie de la ligne contenant l'occurrence substituée par *remplacement*, d'un / et d'un point d'interrogation indiquant que XEDIT attend une réponse.

Vous pouvez alors répondre par :

- [Y] pour accepter le remplacement et passer éventuellement au suivant,
- [N] pour refuser ce remplacement et continuer la recherche,
- [Q] pour quitter la recherche et revenir en mode *commandes*.

### Exemples

R/toto/tata/      remplace toto par tata sur la ligne courante

.\$R/[A-Z]/&&    remplace les majuscules par deux majuscules jusqu'à la fin du fichier

Rx/x\\x          remplace tous les / par des \ sur la ligne courante

?R/a/b          remplace avec confirmation tous les a par des b

## Commande eXchange

La commande X (*eXchange file*) édite un autre fichier sans sortir de XEDIT.

### Syntaxe

X *fichier*

### Valeurs par défaut

*Aucune*

### Opération

La commande X édite un autre fichier sans sortir de XEDIT. Si le fichier nommé n'existe pas, il est créé. S'il existe, il est simplement cherché en mémoire.

### Exemples

X EXEMPLE      édite le fichier EXEMPLE sans sortir de XEDIT

## Références

*A paraître* : Première version de XEDIT par Pierre David et Janick Taillandier.

*Manuel du module Editeur de Textes* par Hewlett-Packard.

*Manuel du module Forth / Assembleur* par Hewlett-Packard.

*Unix User's Reference Manual* par AT&T.

## Mots-clés associés

FILEPOS, FIND, GENLEN, GENPOS, GENRPLC, TEDIT

**Auteurs**

Pierre David, Janick Taillandier

# YAXIS

YAXIS trace un axe vertical avec des marques de graduation optionnelles, à l'abscisse  $x$  spécifiée.

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Ordre | <input checked="" type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction         | <input type="checkbox"/> Mode CALC                       |
| <input type="checkbox"/> Opérateur        | <input checked="" type="checkbox"/> IF...THEN...ELSE     |
|   | <input checked="" type="checkbox"/> Opération d'unité    |

YAXIS *abscisse x*  
YAXIS *abscisse x, espacement*  
YAXIS *abscisse x, espacement, ymin, ymax*

## Exemples

YAXIS 20

Trace un axe à l'abscisse 20.

YAXIS  $(X2 - X1) / 2, -5$

Trace un axe et des marques de graduation espacées de 5 unités graphiques, en commençant par le bas.

## Paramètres d'entrée

Élément	Description	Restriction
abscisse $x$	Expression numérique arrondie à un entier.	-32767 à +32767
espacement	Expression numérique arrondie à un entier. Défaut : 0, pas de marques	-32767 à +32767
ymin	Expression numérique arrondie à un entier. Défaut : 0	-32767 à +32767
ymax	Expression numérique arrondie à un entier. Défaut : hauteur maximum définie par GINIT.	-32767 à +32767

## Opération

YAXIS trace un axe vertical à l'abscisse  $x$ , de  $ymin$  à  $ymax$ .

Si le paramètre *espacement* est présent et non nul, des marques sont tracées toutes les *espacement* unités graphiques, en commençant par  $ymin$  si *espacement* est positif, par  $ymax$  si *espacement* est négatif.

Les longueurs des marques de graduation sont définies par TICLEN. Après exécution de GINIT, elles valent 4.

## Mots-clés associés

LINETYPE, TICLEN, XAXIS

## Auteur

Pierre David

## Chapitre 7

# Les autres caractéristiques de JPC Rom

Les caractéristiques de JPC Rom ne sont pas seulement accessibles par l'intermédiaire des mots-clefs. D'autres sont disponibles, à travers ce qu'on appelle des *polls*.

Ces fonctionnalités ne sont pas appelables. Elles ne sont pas réglables. Elles existent quand JPC Rom est présent dans votre HP-71, sans avoir à faire de manipulation particulière.

Les pages ci-après décrivent l'ensemble de ces caractéristiques.

# La fonction VER\$

La fonction VER\$ renvoie une chaîne identifiant la version de JPC Rom présente dans le HP-71.

## Opération

La fonction VER\$ renvoie une chaîne indiquant la version de JPC Rom.

La version de JPC Rom décrite dans ce manuel est la version : JPC:E.

## Références

*Manuel de référence du HP-71B*

*JPC 23* (page 33) article d'initiation aux polls par Laurent Istria.

*Internal Design Specification* Volume I, chapitre 8.4 et page 17.14.

## Mots-clés associés

VER\$

## Auteurs

Pierre David, Janick Taillandier

# Tabulations assembleur

Les fichiers source pour l'assembleur sont plus lisibles quand les champs sont bien alignés.

## Opération

### Les tabulations :

Bien que l'assembleur accepte les fichiers source en format *libre*, ceux-ci sont plus lisibles s'ils sont bien alignés.

Les tabulations sont accessibles par la touche [SPC] en mode éditeur de texte. Chaque appui sur la touche fait passer à la tabulation suivante, sauf après la dernière tabulation.

Les tabulations sont réglées aux colonnes 9, 16 et 25. Une étoile dans la ligne annule les tabulations de la ligne, l'étoile signifiant un commentaire.

### L'accès au mode tabulations :

La touche [SPC] déplace le curseur à la prochaine tabulation quand :

- le mode *tabulations* est actif,
- le HP-71 est en mode éditeur de textes (avec XEDIT ou TEDIT),
- quand il n'y a pas d'étoile (\*) dans la ligne.

Le mode *tabulation* est activé ou désactivé par l'appui sur la touche [CALC] en mode éditeur de textes.

## Références

*JPC 30* (page 42) la troisième version du Lex par Stéphane Barizien, Pierre David et Michel Martinet.

Manuel du module Forth / Assembleur, page 46.

## Mots-clés associés

XEDIT

## Auteurs

Stéphane Barizien, Pierre David, Michel Martinet

# Correction et Mode CALC

En mode CALC, l'accès à la touche de correction est difficile. JPC Rom donne accès à la touche [<] pour corriger l'introduction.

## Opération

### Le mode CALC :

Le mode CALC (Calcul) est très pratique pour effectuer des calculs. Toutefois, les corrections ne sont pas très aisées, puisqu'il faut appuyer sur deux touches ([g] et [<-]) pour revenir sur le dernier caractère introduit.

Avec JPC Rom dans votre HP-71, il suffit d'appuyer sur la touche [<-] pour revenir en arrière. Les corrections se trouvent grandement facilitées.

## Références

*JPC 27* (page 26) première version du Lex par Pierre David.

## Mots-clés associés

mode CALC

## Auteur

Pierre David

# Position du curseur

Sans interface vidéo, il est quelquefois difficile de savoir où le curseur se situe dans la ligne.

## Opération

En mode USER, une pression sur la touche [ f ] [VIEW] affiche la position du curseur. Cet affichage se maintient tant que la touche est maintenue enfoncée.

La position du curseur est un nombre variant entre les valeurs 1 et 96.

Cette caractéristique est accessible uniquement en mode USER, gardant ainsi la possibilité d'accéder à la fonction VIEW pour les touches assignées.

Cette caractéristique est accessible dans la plupart des modes du HP-71, même en mode Forth, ou sous INPUT, LINPUT, FINPUT ou encore KA et XEDIT.

## Références

*JPC 24* (page 33) première version par Pierre David.

## Mots-clés associés

DEF KEY, FORTH, FINPUT, INPUT, KA, LINPUT, XEDIT

## Auteur

Pierre David

# Accélération du curseur

La présence de JPC Rom dans votre HP-71 accélère la répétition du curseur.

## Opération

Sitôt JPC Rom dans votre HP-71, vous pouvez constater que la répétition automatique du curseur est accélérée notablement. Cette caractéristique est permanente. Elle est accessible en mode éditeur de Textes, en mode Forth et même sous INPUT, LINPUT, FINPUT et sous KA. En revanche, le curseur n'est pas accéléré durant un CAT.

## Références

*JPC 26* (page 29) première version par Jean-Jacques Moreau.

*A paraître* : deuxième version par Pierre David et Janick Taillandier.

## Mots-clés associés

EDTEXT, FORTH, FINPUT, INPUT, LINPUT

## Auteur

Jean-Jacques Moreau

# Initialisation au Memory-Lost

Après une initialisation complète du HP-71, JPC Rom cherche à exécuter un sous-programme nommé ML.

## Opération

Ceci est utile pour initialiser toutes sortes de choses après un Memory Lost. Par exemple, l'heure et la date, le délai, le contraste, etc. Voici un exemple :

```
100 SUB ML
110 DIM D$
120 FINPUT D$, "Heure : Hr:Mn:Sc", "8P2UP2UP2UP", A
130 SETTIME D$[1,2]&":"&D$[3,4]&":"&D$[5]
120 FINPUT D$, "Date : Jr/Mo/An", "7P2UP2UP2UP", A
130 SETDATE D$[5]&"/"&D$[3,4]&"/"&D$[1,2]
140 DELAY 0,0
150 USER ON
160 STACK 15
170 LC ON
180 WIDTH 80
190 PWIDTH INF
200 DMY
210 ROMAN ON
220 DEF KEY "#46", "RUN ";
230 END SUB
```

Bien sûr, ce programme doit se trouver en port indépendant ou en module Rom ou Eprom pour ne pas être perdu lors d'un Memory Lost.

Attention : le sous programme ne doit pas contenir d'ordre faisant appel à une configuration. De tels ordres sont, par exemple, LEX ON/OFF ou COPY de fichiers Lex en mémoire, etc.

## Références

*JPC 31* (page 24) première version par Jean-Jacques Moreau.

## Mots-clés associés

CALL, SUB

## Auteur

Jean-Jacques Moreau

# Reconnaissance des types de fichiers

JPC Rom reconnaît de nouveaux types de fichiers non standard et les affiche en clair lors d'un CAT ou DDIR.

## Opération

Lorsque JPC Rom est dans le HP-71, il affiche en clair le nom des fichiers non standard lors d'un ordre CAT, d'une fonction CAT\$ ou d'un ordre DDIR ou PDIR.

Ceci permet de distinguer et de reconnaître facilement des fichiers produits par d'autres appareils (par exemple HP-41 ou HP-75) lorsqu'ils sont sur un support de mémoire de masse.

La liste des types de fichiers reconnus est :

Fichiers produits par le HP-71

ADRS	Carnets d'adresses utilisés par KA (JPC Rom)
GRAPH	Images graphiques produites par le graphique (JPC Rom)
FORTH	Forthrams des modules Forth ou Translator
ROM	Images de ports indépendants (fichier Lex ROMCOPY)
OBJ	Fichiers objets (Module de développement, non commercialisé)
SYM	Table des symboles (Module de développement, non commercialisé)

Fichiers produits par le HP-41

41:XM	Sauvegarde de la mémoire étendue
41:WA	Sauvegarde du contenu total du HP-41 (fonction WRTA)
41:KE	Sauvegarde des assignations (fonction WRTK)
41:ST	Sauvegarde de l'état (fonction WRTS)
41:PR	Programme HP-41
41:ML	Programmes en micro-code pour MLDL

Fichiers produits par le HP-75

75:T	Fichier Texte
75:A	Fichier d'alarmes (Appt)
75:B	Programme Basic
75:L	Fichier Lex
75:W	Feuille de calcul VisiCalc
75:G	Fichier à usage général de la Rom I/O
75:R	Rom PMS

## Références

*A paraître* : première version par Jan Buitenhuis et Janick Taillandier.

## Mots-clés associés

CAT, CAT\$, DDIR, PDIR

## Auteurs

Jan Buitenhuis, Janick Taillandier

# Les messages du module HP-IL

Les messages d'erreur du module HP-IL ne sont pas toujours très explicites. JPC Rom les redéfinit de manière à avoir des diagnostics clairs et précis.

## Opération

Le module HP-IL fait exactement 16384 octets, c'est à dire qu'il ne reste même pas un seul quartet de disponible dans le module HP-IL. On comprend mieux maintenant pourquoi les messages renvoyés sont souvent répétitifs. Hewlett-Packard a dû ignorer des messages d'erreur pour pouvoir faire tenir la ROM HP-IL dans un module de 16 Kilo octets.

JPC Rom redéfinit ces messages de manière à les rendre clairs et précis.

Voici la liste des messages, en première position on trouve le message standard, puis en deuxième position le message de JPC Rom :

0 : *inexistent*

0 : HPIL

Message 0, il est utilisé par le système dans l'affichage des erreurs.

Exemple : HPIL ERR:Blank Medium

1 : ASSIGN IO Needed

1 : ASSIGN IO Needed

Tentative d'exécution de LIST IO sans ASSIGN IO préalable. Exécuter ASSIGN IO.

3 : Excess Chars

3 : Excess Chars

Le HP-71 a trouvé plus de caractères qu'il en attendait dans un ordre. Vérifier la syntaxe.

4 : Missing Parm

4 : Missing Parm

Un des paramètres requis pour l'ordre manque. Vérifier la syntaxe.

5 : Invalid Parm

5 : Invalid Parm

Un des paramètres utilisé pour l'ordre n'est pas valide. Vérifier les paramètres.

6 : Invalid Expr

6 : Invalid Expr

L'expression ne peut pas être évaluée à cause d'un type non valide (tel que l'utilisation d'une variable alpha à la place d'une valeur numérique). Vérifier l'expression.

7 : Syntax

7 : Syntax

Le HP-71 ne reconnaît pas l'ordre. Vérifier l'orthographe des mots-clés et la validité des paramètres.

16 : File Protect

16 : File Protect

Le fichier est sécurisé ou privé; vous ne pouvez pas effectuer cette opération. Si le fichier est sécurisé, exécuter l'ordre UNSECURE.

17 : End Of Medium

17 : End of Medium

Le fichier est trop grand pour l'espace disponible sur le support; le support est plein; condition d'erreur d'entraîneur. Vérifier le support; recréer le fichier; compacter le support; utiliser un autre support de mémoire de masse.

18 : Invalid Medium

18 : Disk Drive Error

Le moteur de l'unité de stockage de masse ne fonctionne pas. Vérifier que le support (disquette ou cassette) n'est pas

# Les messages du module HP-IL (suite)

bloqué.

19: Invalid Medium

19: Not LIF Format

Le support n'est pas initialisé avec un format correct. Exécuter l'ordre INITIALIZE.

20: No Medium

20: No Medium

Le HP-71 ne trouve pas de support dans l'unité de stockage de masse. Vérifier que la porte est fermée ; insérer un support.

21: *inexistent*

21: Low Battery

L'alimentation de l'unité de stockage de masse est défectueuse. Recharger ou changer les batteries.

22: File Not Found

22: File Not Found

Le fichier spécifié est absent ; le nom de fichier spécifié diffère de l'étiquette reçue de l'unité de stockage de masse. Vérifier le catalogue et le nom de fichier.

23: Invalid Medium

23: New Medium

Ouverture et fermeture de la porte de l'unité de stockage de masse pendant une opération de stockage ou pendant l'exécution d'un ordre de positionnement du support. Le support est mal positionné : recommencer l'opération ou l'ordre.

24: Invalid Medium

24: Blank Medium

Support neuf non initialisé. Utiliser l'ordre INITIALIZE.

25: Invalid Medium

25: Wrong dir # records

Le nombre d'enregistrements indiqué dans l'étiquette ne correspond pas au nombre réel d'enregistrements sur le support. Recommencer l'opération. Si vous obtenez la même erreur, initialisez le support (ordre INITIALIZE) puis recréez le fichier.

26: Invalid Medium

26: Checksum

Le processeur de l'unité de stockage a détecté une erreur de somme de contrôle. Recréer le fichier.

28: Size of File

28: Size of File

Fichier trop grand pour être stocké ou appelé d'une unité de stockage de masse. Ajouter un module mémoire au HP-71 ou utiliser un autre support.

29: *inexistent*

29: Write Protected

Erreur réservée au lecteur de disquettes. La disquette se trouvant dans l'unité de mémoire de masse est protégée contre l'écriture.

30: File Exists

30: File Exists

Le nom de fichier spécifié dans un ordre CREATE ou en destination d'un ordre COPY existe déjà. Supprimer l'ancien fichier ou changer le nom de l'ancien ou du nouveau fichier.

31: Directory Full

31: Directory Full

Le catalogue du support est plein. Supprimer les fichiers inutiles et compacter le catalogue ou le support.

## Les messages du module HP-IL (suite)

32: Device Not Found

32: Device Not Found

L'unité demandée n'est pas sur la boucle. Vérifier le spécificateur d'appareil; vérifier l'organisation du système; exécuter RESTORE IO.

34: Device Not Ready

34: Device Not Ready

Une unité n'a pas répondu (n'a pas envoyé de données ou n'a pas accepté le contrôle, par exemple). Vérifier le spécificateur d'unité; vérifier l'unité; exécuter RESTORE IO.

35: Loop Broken

35: Loop Broken

La boucle n'est pas fermée. Vérifier les connexions et que toutes les unités sont allumées.

36: Message Error

36: Too Many Frames

Le HP-71 a reçu trop de messages. Recommencer l'opération.

37: Message Error

37: Frames Lost

Message perdu à cause d'une retransmission trop lente. Recommencer l'opération.

38: Message Error

38: Frames Altered

Message modifié pendant une transmission. Recommencez l'opération.

39: Unexpected Message

39: Unexpected Message

Violation du protocole HP-IL (plusieurs émetteurs étaient actifs en même temps); l'émetteur indique une erreur de transmission. Recommencer l'opération.

40: Message Error

40: Too Many Frames

Le HP-71 a reçu trop de messages. Recommencer l'opération.

41: Invalid Mode

41: Invalid Mode

Tentative d'exécution d'un ordre de contrôleur lorsque le HP-71 opère en tant qu'unité. Vérifier le mode (contrôleur ou unité) requis par l'ordre.

42: Loop Broken

42: Message Altered

Le HP-71 a reçu un message partiel à cause d'une erreur de transmission. Recommencer l'opération.

43: Loop Broken

43: Loop Timeout

Un message n'a pas parcouru la boucle pendant le délai d'attente spécifié par STANDBY. Annulez les états récepteurs et recommencer l'opération.

44: System Error

44: Bad Addresses

Adresses d'unités probablement non valides (si l'indicateur -24 est armé). Effacer l'indicateur -24 ou affecter de nouvelles adresses (exécuter RESTORE IO).

Erreur interne concernant les canaux d'entrée/sortie. Exécuter RUN et recommencer l'opération; exécuter INIT:1; exécuter INIT:3. Si l'erreur se maintient, l'interface HP-IL est défectueuse et doit être réparée.

# Les messages du module HP-IL (suite)

45: Self-test failed

45: Self Test Failed

L'interface n'a pas passé son auto-test interne. Répéter l'auto-test en exécutant RESET HPIL. Si l'erreur persiste, L'interface est défectueuse et doit être réparée.

47: Device Type

47: Device Type

L'unité spécifiée n'est pas du type correct pour cet ordre. Vérifier le type l'unité adressée.

52: Aborted

52: Aborted

Vous avez appuyé deux fois sur [ATTN] pour interrompre le fonctionnement de la boucle. Exécuter RESTORE IO; si nécessaire, RESET HPIL puis RESTORE IO. Vérifier les connexions HP-IL; vérifier que les unités sont allumées.

53: Invalid Device Spec

53: Invalid Device Spec

Le spécificateur d'unité n'est pas valide pour cet ordre. Vérifier le spécificateur d'unité.

54: Data Type

54: Data Type

Le type de variable spécifiée est incorrect (numérique ou alphanumérique). Changer l'argument de façon que le type soit correct.

56: Invalid Arg

56: Invalid Arg

Argument en dehors de l'intervalle permis. Vérifier la valeur de l'argument.

L'étiquette (enregistrement de début ou longueur) reçue pendant une opération de stockage de masse est incorrect. Restocker le fichier.

57: No Loop

57: No Loop

L'interface n'est pas installée. Vérifier la configuration du système.

59: Insufficient Memory

59: Insufficient Memory

Pas assez de mémoire principale pour effectuer l'opération. Ajouter un module mémoire; supprimer des fichiers ou des affectations de touches; réallouez la mémoire vive interne.

60: RESTORE IO Needed

60: RESTORE IO Needed

Tentative d'exécution d'une opération d'entrées sorties après l'exécution de OFF IO. Exécuter RESTORE IO.

## Références

JPC 37 (page 27) première version par Michel Martinet.

Manuel du module HP-IL, Annexe E.

## Mots-clés associés

MSG\$, Tous les mots-clefs du module HP-IL

## Auteur

Michel Martinet

# Chapitre 8

## Les auteurs

Voici la liste des auteurs ayant contribué à JPC Rom. Chaque mot-clef est suivi de son token et des auteurs.

ADBUF\$	XFN	225001	Pierre David, Michel Martinet
ASC\$	XFN	225002	Pierre David, Michel Martinet
ATH\$	XFN	225003	Pierre David, Michel Martinet
HTA\$	XFN	225004	Pierre David, Michel Martinet
RED\$	XFN	225005	Pierre David, Michel Martinet
REPLACE\$	XFN	225006	Michel Martinet, Jean-Jacques Moreau
FILESIZE	XFN	225007	Henri Kudelski
ATTN	XWORD	225008	Pierre David, Michel Martinet
LEX OFF	XWORD	225009	Pierre David, Michel Martinet, Janick Taillandier
LEX ON	XWORD	225010	Pierre David, Michel Martinet, Janick Taillandier
FKEY	XWORD	225011	Jean-Pierre Bondu
CONTRAST	XWORD	225012	Laurent Istria, Jean-Jacques Moreau
INVERSE	XWORD	225013	Pierre David, Jean-Jacques Moreau, Janick Taillandier
PAINT	XFN	225015	Pierre David, Jean-Jacques Moreau, Janick Taillandier
ENDUP\$	XFN	225016	Jean-Jacques Moreau
ENDUP	XWORD	225017	Jean-Jacques Moreau
STARTUP\$	XFN	225018	Jean-Jacques Moreau
EXECUTE	XWORD	225019	Jean-Jacques Moreau
ARR	XFN	225020	Laurent Istria, Guy Toublanc
COMB	XFN	225021	Laurent Istria, Guy Toublanc
HMSADD	XFN	225022	Michel Martinet, Guy Toublanc
HMSUB	XFN	225023	Michel Martinet, Guy Toublanc
HMS	XFN	225024	Michel Martinet, Guy Toublanc
HR	XFN	225025	Michel Martinet, Guy Toublanc
EDIT	XWORD	225026	Jean-Pierre Bondu, Pierre David, Michel Martinet
STACK	XWORD	225027	Henri Kudelski, Michel Martinet
MARGIN	XWORD	225029	Pierre David, Michel Martinet, Janick Taillandier
MENU	XFN	225033	Jean-Jacques Dhénin
CENTER\$	XFN	225034	Pierre David, Michel Martinet
CESURE	XFN	225035	Pierre David, Michel Martinet
FORMAT\$	XFN	225036	Pierre David, Michel Martinet
REDUCE\$	XFN	225037	Pierre David, Michel Martinet
SPACE\$	XFN	225038	Pierre David, Michel Martinet, Janick Taillandier
BELL	XWORD	225039	Pierre David
BOLD	XWORD	225040	Pierre David
PCR	XWORD	225041	Pierre David
ESC\$	XFN	225042	Pierre David
PFF	XWORD	225043	Pierre David
PLF	XWORD	225044	Pierre David
MODE	XWORD	225045	Pierre David
PERF	XWORD	225046	Pierre David
PAGELEN	XWORD	225047	Pierre David

## Les auteurs

UNDERLINE	XWORD	225048	Pierre David
WRAP	XWORD	225049	Pierre David
DATESTR\$	XFN	225050	Pierre David, Janick Taillandier
DATEADD	XFN	225051	Pierre David, Laurent Istria, François Le Grand, Janick Taillandier
DDAYS	XFN	225052	Pierre David, Laurent Istria, Janick Taillandier
DMY	XWORD	225053	Pierre David, Laurent Istria, Janick Taillandier
DOW\$	XFN	225054	Pierre David, Laurent Istria, Janick Taillandier
DOW	XFN	225055	Pierre David, Laurent Istria, Janick Taillandier
MDY	XWORD	225056	Pierre David, Laurent Istria, Janick Taillandier
MAXD	XFN	225057	Michel Martinet
MAXM	XFN	225058	Michel Martinet
MEMD	XFN	225059	Michel Martinet
MEMM	XFN	225060	Michel Martinet
EXIT	XWORD	225061	Janick Taillandier
NLOOP	XFN	225062	Jean-François Garnier
PARPOLL	XFN	225063	Jean-François Garnier
SLEEP	XWORD	225064	Jean-François Garnier
SRQ	XFN	225065	Jean-François Garnier
END	XWORD	225066	Pierre David, Janick Taillandier
WHILE	XWORD	225067	Pierre David, Janick Taillandier
REPEAT	XWORD	225068	Pierre David, Janick Taillandier
UNTIL	XWORD	225069	Pierre David, Janick Taillandier
LEAVE	XWORD	225070	Pierre David, Janick Taillandier
VARSWAP	XWORD	225071	Pierre David, Jean-Jacques Moreau, Janick Taillandier
ENTRY\$	XFN	225072	Jean-Jacques Moreau
TOKEN	XFN	225073	Jean-Jacques Moreau
FIND	XWORD	225075	Pierre David, Jean-Jacques Moreau, Janick Taillandier
MAP\$	XFN	225076	Tapani Tarvainen
MAP	XWORD	225077	Tapani Tarvainen
GLINE	XWORD	225078	Pierre David
GPSET	XWORD	225079	Pierre David
SHRINK	XWORD	225081	Jean-Jacques Moreau
FPRIM	XFN	225082	Guy Toublanc
NPRIM	XFN	225083	Guy Toublanc
PGCD	XFN	225084	Guy Toublanc
PHI	XFN	225085	Guy Toublanc
PPCM	XFN	225086	Guy Toublanc
PRIM	XFN	225087	Olivier Arbey, Guy Toublanc
FRAC\$	XFN	225088	Guy Toublanc
POSI	XFN	225089	Jean-Pierre Bondu, Pierre David, Janick Taillandier
DBLIST	XWORD	225090	Jean-Pierre Bondu, Pierre David, Janick Taillandier
PBLIST	XWORD	225091	Jean-Pierre Bondu, Pierre David, Janick Taillandier
RENUMREM	XWORD	225092	Jean-Pierre Bondu
FINPUT	XWORD	225093	Pierre David, Janick Taillandier
RREC\$	XFN	225094	Michel Martinet
WREC	XWORD	225095	Michel Martinet
LOOP	XWORD	225096	Pierre David, Janick Taillandier
SELECT	XWORD	225097	Pierre David, Janick Taillandier
CASE	XWORD	225098	Pierre David, Janick Taillandier
IF	XWORD	225099	Pierre David, Janick Taillandier
ELSE	XWORD	225100	Pierre David, Janick Taillandier
KA	XWORD	225101	Pierre David
ADCREATE	XWORD	225102	Pierre David
ADDELETE	XWORD	225103	Pierre David
ADFIND	XFN	225104	Pierre David
ADGET	XWORD	225105	Pierre David
ADPUT	XWORD	225106	Pierre David
ADSIZE	XFN	225107	Pierre David
ROMAN	XWORD	225108	Pierre David, Janick Taillandier
DDIR	XWORD	225110	Jean-Jacques Dhénin
PDIR	XWORD	225111	Jean-Jacques Dhénin

FILEPOS	XFN	225112	Pierre David, Janick Taillandier
GENLEN	XFN	225113	Pierre David, Janick Taillandier
GENPOS	XFN	225114	Pierre David, Janick Taillandier
GENRPLC\$	XFN	225115	Pierre David, Janick Taillandier
TEDIT	XFN	225116	Pierre David, Janick Taillandier
XEDIT	XFN	225117	Pierre David, Janick Taillandier
BOX	XWORD	104001	Pierre David
CSIZE	XWORD	104002	Pierre David
DRAW	XWORD	104003	Pierre David
FRAME	XWORD	104004	Pierre David
GDUMP	XWORD	104005	Pierre David
GEND	XWORD	104006	Pierre David
GINIT	XWORD	104007	Pierre David
IDRAW	XWORD	104008	Pierre David
IMOVE	XWORD	104009	Pierre David
LABEL	XWORD	104010	Pierre David
LDIR	XWORD	104011	Pierre David
LINETYPE	XWORD	104012	Pierre David
LORG	XWORD	104013	Pierre David
MOVE	XWORD	104014	Pierre David
PENDOWN	XWORD	104015	Pierre David
PENUP	XWORD	104016	Pierre David
PEN	XWORD	104017	Pierre David
PLOTTER	XWORD	104018	Pierre David
TICLEN	XWORD	104019	Pierre David
XAXIS	XWORD	104020	Pierre David
YAXIS	XWORD	104021	Pierre David
KEYWAIT\$	XFN	82001	Hewlett-Packard
MERGE	XWORD	1057	Pierre David, Michel Martinet
PEEK\$	XFN	1063	Pierre David, Michel Martinet
POKE	XWORD	1064	Pierre David, Michel Martinet

**Les auteurs**

## Chapitre 9

# Messages de JPC Rom

225000 JPC  
225001 Driver Lex File  
225002 Not Found  
225003 Structure Mismatch  
225004 Invalid Prompt  
225005 Invalid Format  
225006 #Dims  
225007 Var Not Found  
225008 Sunday  
225009 Monday  
225010 Tuesday  
225011 Wednesday  
225012 Thursday  
225013 Friday  
225014 Saturday  
225015 Function Interrupted  
225016 Removed Keyword  
225017 Insufficient Memory  
225018 Invalid Cmd  
225019 Invalid Pattern  
225020 No Room for Pattern  
225021 Line # Cmd:  
225022 Eof, Cmd:  
225023 [Eof  
225024 Ok to Delete? Y/N:  
225025 Yes/No/Quit ?  
225026 YNQ  
225027 CDEHIJLMPQRSTX  
225028 Copy: [b[e]] C [<file>]  
225029 Delete: [b[e]] D [<file>[+]]  
225030 Exit: E  
225031 Help: H [<cmd>]  
225032 Insert: [l] I  
225033 Join: [b[e]] [?] J [n]  
225034 List: [b[e]] L [n][N]  
225035 Move: [b[e]] M [<file>]  
225036 Print: [b[e]] P [n][N]  
225037 Quit: Q  
225038 Replace: [b[e]] [?] R/str1/str2[/]  
225039 Search: [b[e]] [?] S/str[/]  
225040 Text: [l] T  
225041 Exchange File: X <file>  
225042 (c) 1986, 1987, 1988, 1989 PPC-Paris

## Messages de JPC Rom

104000 GRPH  
104001 Invalid Graph type  
104002 Graph not initialized  
104003 No Loop  
104004 Module missing  
104005 Severe error  
104006 Plotter not reachable  
104007 GRAPHILE Not found

# Chapitre 10

## Vous avez trouvé une erreur

Vous avez trouvé une erreur dans le fonctionnement de JPC Rom ou dans le manuel. Ne gardez pas ce secret pour vous ! Faites nous en part, ce qui nous permettra de faire la correction dans les versions ultérieures. Il se pourrait bien, d'autre part, que votre nom soit cité, comme vos prédécesseurs, dans les pages listant les différences au début de ce manuel.

Remplissez soigneusement cette page. Soyez précis. Essayez de reproduire le problème par un exemple aussi court que possible. Plus facilement le problème est mis en évidence, et plus vite nous pouvons le corriger.

Vérifiez en outre que le problème n'est pas déjà spécifié dans le manuel, par exemple sous forme de restriction d'utilisation.

Faites autant de copies de cette page que nécessaire, et n'hésitez pas à joindre des feuillets additionnels si vous n'avez pas assez de place. Envoyez le tout à :

PPC Paris  
B.P. 604  
75028 Paris Cedex 01  
France

Votre nom :

Votre numéro de téléphone (de préférence) ou votre adresse :

Version de votre HP-71B (résultat de VER\$) :

La configuration minimale (modules mémoire, HP-IL, fichiers Lex présents, etc.) avec laquelle le problème apparaît :

Version de JPC Rom (résultat de VER\$) :

Nom du mot-clef :

Description du problème (environnement, symptômes, que tentiez-vous de faire, quel est le comportement que vous estimez anormal, et toute autre information pouvant être utile) :

Instructions détaillées pour reproduire le problème :