

# MAIN DIFFERENCES BETWEEN VERSION D AND VERSION E

The differences between version D and version E (see VER\$ in *Other JPC Rom features*) result from some bug fixes, modifications and improvements. The version E mainly adds two large parts, namely the editor and the graphic. The main differences are listed below:

## Bug fixes

The addition of the pseudo keyword `obsolete` lead to some problems when the keywords of JPC Rom were analysed with a Basic program.

The structured programming keywords were affected by a problem when they were followed by a comment, like in:  
`10 LEAVE !`

This was also the case for `WHILE`, `IF` and `ELSE`. Error reported by Joseph A. Dickerson from the USA.

The `SELECT` structure didn't work well in the following case, for instance:

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

The `BEEP` statement was never executed. Error reported by Laurent Istria and Jean Horville.

## New keywords and new features

The editor in assembly language from Pierre David and Janick Taillandier was added to JPC Rom. It provides the following keywords: `XEDIT`, `TEDIT`, `GENPOS`, `GENLEN`, `GENRPLC$` and `FILEPOS`. The *Assembler Tabs* mode now works with `XEDIT` and `TEDIT` and no more with `EDTEXT`.

The graphic keywords (`BOX`, `CSIZE`, `DRAW`, `FRAME`, `GDUMP`, `GEND`, `GINIT`, `IDRAW`, `IMOVE`, `LABEL`, `LDIR`, `LINETYPE`, `LORG`, `MOVE`, `PENDOWN`, `PENUP`, `PEN`, `PLOTTER`, `TICLEN`, `XAXIS` and `YAXIS`) from Pierre David have been added.

## Modifications and improvements

The `FIND` keyword, despite several successive corrections, still didn't give satisfactory results. It was entirely rewritten, furthermore it now takes advantage of the generic string feature from `XEDIT` and of two additional parameters. By Pierre David and Janick Taillandier.

The address parameter in the keywords `OPCODE$`, `NEXTOP$`, `PEEK$`, `POKE` and `SYSEDIT` can now be specified in decimal, to simplify the calculation and avoid useless conversions.

The keyword `LEAVE` accepts now an optional parameter to specify the number of nested loops to exit from.

## Note

All these improvements and corrections have been realized preserving program compatibility with previous versions of JPC Rom. So your programs written with previous versions are totally compatible with the new JPC Rom.

## DIFFERENCES 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.

**BOX**

BOX trace un rectangle.

- |   |  |
|---|--|
| <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    |

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.

**Paramètres d'entrée**

Elément	Description	Restrictions
$x1, y1, x2, y2$	Expression numériques arrondies a 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

# CSIZE

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

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>■ Ordre</li> <li>○ Fonction</li> <li>○ Opérateur</li> </ul> | <ul style="list-style-type: none"> <li>■ Exécution au clavier</li> <li>○ Mode CALC</li> <li>■ IF...THEN...ELSE</li> <li>■ Opération d'unité</li> </ul> |
|--|--|

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

Elément	Description	Restrictions
hauteur rapport	Expression numérique arrondie à un entier. Expression numérique Défaut : 0.6	-32767 à +32767 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.

2

**CSIZE** (suite)

**Mots-clés associés**

LABEL

**Auteur**

Pierre David

# 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  $(x + r \cos t, y + r \sin t)$ .

## Paramètres d'entrée

Elément	Description	Restrictions
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

# FILEPOS

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

- |  |  |
|--|--|
| <input type="radio"/> Ordre                  | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="radio"/> Mode CALC                          |
| <input type="radio"/> 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	Restrictions
fichier canal	Expression alphanumérique. Expression numérique arrondie à un entier	Le fichier doit être en mémoire vive. 1 à 255, et le fichier doit être en mémoire vive.
motif début	Expression alphanumérique. Expression numérique arrondie à un entier. Défaut : 0	Expression générique valide. 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 (suite)

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

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 et Janick Taillandier.



# FIND

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

- |             |                        |
|-------------|------------------------|
| ■ Ordre     | ■ Exécution au clavier |
| ○ Fonction  | ○ Mode CALC            |
| ○ Opérateur | ○ IF...THEN...ELSE     |
|             | ■ 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	Restrictions
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.	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.

**FIND** (suite)

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 et Janick Taillandier

# 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**

GDUMP (Graphic DUMP)

- |             |                        |
|-------------|------------------------|
| ■ Ordre     | ■ Exécution au clavier |
| ○ Fonction  | ○ Mode CALC            |
| ○ Opérateur | ■ IF...THEN...ELSE     |
|             | ○ Opération d'unité    |

GDUMP

**Exemple**

```
IF UPRC$(R$)="O" THEN 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     | ■ Exécution au clavier |
| ○ Fonction  | ○ Mode CALC            |
| ○ Opérateur | ■ IF...THEN...ELSE     |
|             | ■ Opération d'unité    |

GEND

## Exemple

```
100 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 déclenchera l'erreur :  
GRPH ERR:Graph not initialized

## Mots-clés associés

GINIT, PLOTTER IS

## Auteur

Pierre David

# GENLEN

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

- |  |  |
|--|--|
| <input type="radio"/> Ordre                  | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="radio"/> Mode CALC                          |
| <input type="radio"/> 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

Elément	Description	Restrictions
chaîne motif début	Expression alphanumérique. Expression alphanumérique. Expression numérique arrondie à un entier. Défaut : 0	Aucune. Expression générique valide. 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.

**GENLEN** (suite)

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.

**Mots-clés associés**

XEDIT, FILEPOS, FIND, GENPOS, GENRPLC, LEN

**Auteurs**

Pierre David et Janick Taillandier.

# GENPOS

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

- |  |  |
|--|--|
| <input type="radio"/> Ordre                  | <input checked="" type="checkbox"/> Exécution au clavier |
| <input checked="" type="checkbox"/> Fonction | <input type="radio"/> Mode CALC                          |
| <input type="radio"/> 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

Elément	Description	Restrictions
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



## GENPOS (suite)

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

## Auteurs

Pierre David et Janick Taillandier.

## GENRPLC\$

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

- |                                   |   |
|-----------------------------------|---|
| <input type="radio"/> Ordre       | <input type="checkbox"/> Exécution au clavier |
| <input type="checkbox"/> Fonction | <input type="radio"/> Mode CALC               |
| <input type="radio"/> Opérateur   | <input type="checkbox"/> IF...THEN...ELSE     |
|                                   | <input 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 AS 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

Elément	Description	Restrictions
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 :

## GENRPLC\$ (suite)

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
[^ ]	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, REPLACE\$

GINIT (Graphic INITIALisation) initialise une session graphique.

- |             |                        |
|-------------|------------------------|
| ■ Ordre     | ■ Exécution au clavier |
| ○ Fonction  | ○ Mode CALC            |
| ○ Opérateur | ■ IF...THEN...ELSE     |
|             | ■ Opération d'unité    |

GINIT "RASTER", *dimension*

## Exemples

GINIT "RASTER", 100

dimensionne une zone de traçage de 100 unités graphiques de hauteur.

GINIT "RASTER", H

## Paramètre d'entrée

Élément	Description	Restrictions
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

2

**GINIT** (suite)

**Mots-clés associés**

GEND, PLOTTER IS

**Auteur**

Pierre David

# 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.

## Paramètres d'entrée

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

## Opération

DRAW 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

## IMOVE

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 *incrément x, incrément y*

### 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.

### Paramètres d'entrée

Élément	Description	Restrictions
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

# LABEL

LABEL trace des caractères alphanumériques à partir de la position courante de la plume.

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>■ Ordre</li> <li>○ Fonction</li> <li>○ Opérateur</li> </ul> | <ul style="list-style-type: none"> <li>■ Exécution au clavier</li> <li>○ Mode CALC</li> <li>■ IF...THEN...ELSE</li> <li>■ Opération d'unité</li> </ul> |
|--|--|

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	Restrictions
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.



2

**LABEL** (suite)

**Mots-clés associés**

CSIZE, LDIR, LONG, ENDLINE

**Auteur**

Pierre David

**LDIR**

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).

LDIR ASIN(.5)

L'angle entre la direction des labels et l'axe des  $x$  est de 30 degrés.

**Paramètre d'entrée**

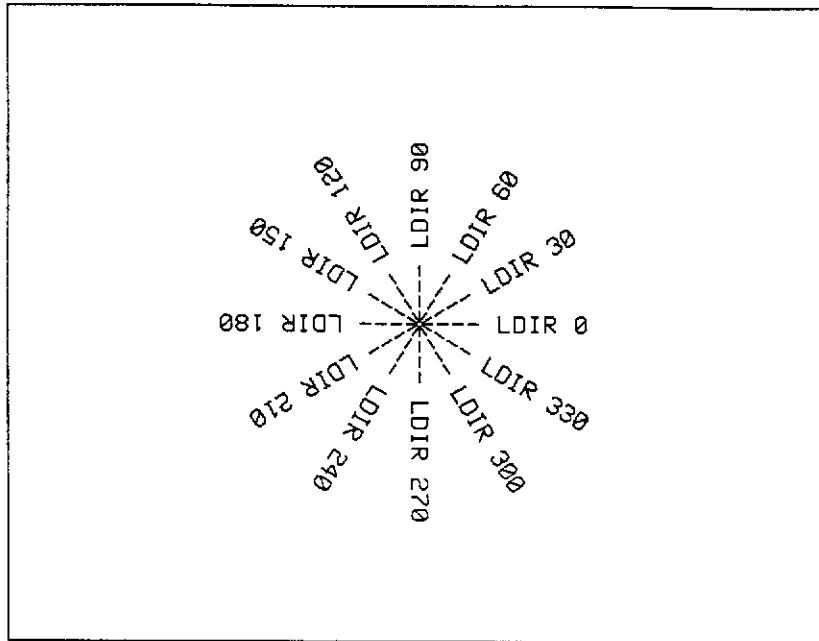
Elément	Description	Restrictions
angle	Expression numérique, interprétée suivant le mode trigonométrique en vigueur.	0 à 360 degrés, ou 0 à $2 \cdot \text{PI}$

**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.

**LDIR (suite)****Mots-clés associés**

DEGREES, RADIANS, LABEL

**Auteur**

Pierre David

# LEAVE

LEAVE sort proprement d'une boucle structurée de type WHILE, REPEAT ou LOOP.

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>■ Ordre</li> <li>○ Fonction</li> <li>○ Opérateur</li> </ul> | <ul style="list-style-type: none"> <li>■ Exécution au clavier</li> <li>○ Mode CALC</li> <li>■ IF...THEN...ELSE</li> <li>■ Opération d'unité</li> </ul> |
|--|--|

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	Restrictions
profondeur	Expression numérique arrondie à un entier Défaut : 1	1 au nombre de boucles 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.

2

**LEAVE** (suite)

### **Mots-clés associés**

EXIT, LOOP ... END LOOP, REPEAT ... UNTIL, WHILE ... END WHILE

### **Auteurs**

Pierre David et Janick Taillandier

# LINETYPE

LINETYPE sélectionne le type et la longueur du motif pour le tracé de lignes, axes et rectangles.

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>■ Ordre</li> <li>○ Fonction</li> <li>○ Opérateur</li> </ul> | <ul style="list-style-type: none"> <li>■ Exécution au clavier</li> <li>○ Mode CALC</li> <li>■ IF...THEN...ELSE</li> <li>■ Opération d'unité</li> </ul> |
|--|--|

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

Elément	Description	Restrictions
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.

—————	LINETYPE 1
.....	LINETYPE 2
.....	LINETYPE 3
-----	LINETYPE 4
-----	LINETYPE 5
=====	LINETYPE 6
=====	LINETYPE 7
=====	LINETYPE 8
—————	LINETYPE 1,48
.....	LINETYPE 2,48
.....	LINETYPE 3,48
-----	LINETYPE 4,48
-----	LINETYPE 5,48
=====	LINETYPE 6,48
=====	LINETYPE 7,48
=====	LINETYPE 8,48

2

## **LINETYPE** (suite)

### **Mots-clés associés**

DRAW, IDRAW, BOX, FRAME, XAXIS, YAXIS

### **Auteur**

Pierre David

**LORG**

LORG (Label ORiGin) définit la position des labels par rapport 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    |

LORG *position*

**Exemples**

LORG 5

Les labels seront centrés par rapport à la position courante de la plume.

LORG X

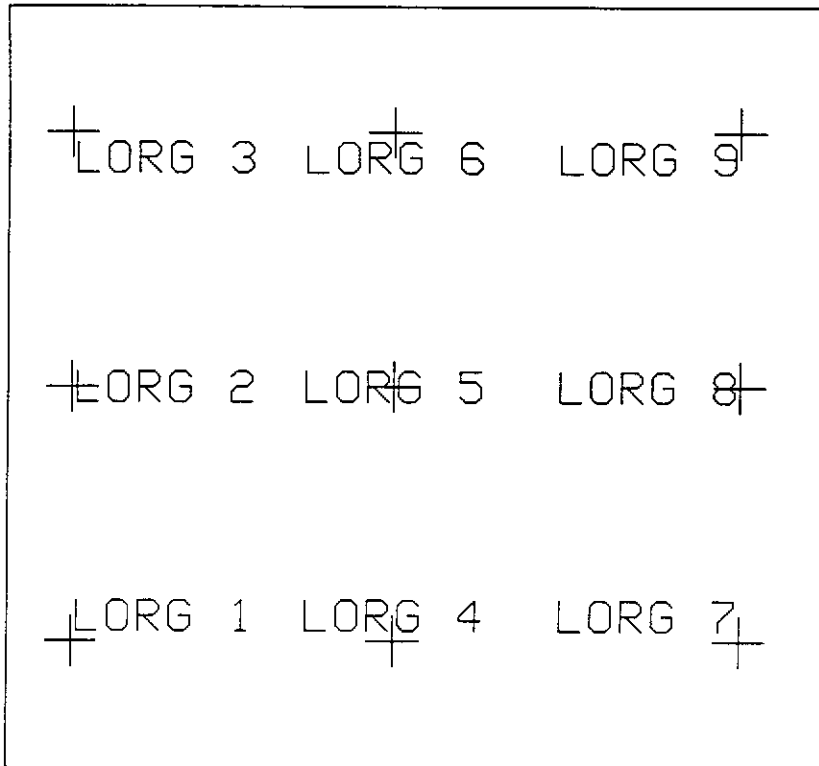
**Paramètre d'entrée**

Elément	Description	Restrictions
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).



**LORG** (suite)

La fonction LORG utilise la valeur absolue de la partie entière de l'argument. Si le nombre est supérieur à 9, la fonction utilise cette valeur modulo 10. Si le paramètre est nul, LORG utilise la valeur 1 définie par défaut lors de l'exécution de GINIT.

Note : Attention à l'introduction de LORG avec un argument constant. L'espace avant le nombre est impératif, sinon le HP71 comprendra LORG5 comme :

DISP L OR G5

ce qui n'est évidemment pas le but recherché !

**Mots-clés associés**

LABEL, MOVE

**Auteur**

Pierre David

**MOVE**

MOVE lève la plume et la déplace jusqu'au point spécifié par les coordonnées.

- |   |  |
|---|--|
| <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    |

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)

**Paramètres d'entrée**

Élément	Description	Restrictions
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

**PEN**

PEN sélectionne une 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
--	---

PEN <i>numéro de plume</i>
----------------------------

**Exemples**

PEN 2

Sélectionne la plume 2.

PEN X

Sélectionne la plume numéro X.

**Paramètre d'entrée**

Élément	Description	Restrictions
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

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

## PLOTTER IS

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	Restrictions
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

## TEDIT

TEDIT (Text EDITor) est un éditeur de textes extrêmement rapide, compatible avec l'éditeur de textes Hewlett-Packard.

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>■ Ordre</li> <li>○ Fonction</li> <li>○ Opérateur</li> </ul> | <ul style="list-style-type: none"> <li>■ Exécution au clavier</li> <li>○ Mode CALC</li> <li>■ IF...THEN...ELSE</li> <li>■ Opération d'unité</li> </ul> |
|--|--|

TEDIT *fichier*  
TEDIT *fichier* , *chaîne de commandes*

### 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	Restrictions
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**.

## TEDIT (suite)

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 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.



## TEDIT (suite)

- 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
<code>\</code>	annule la signification du caractère suivant
<code>^</code>	début de ligne
<code>\$</code>	fin de ligne
<code>.</code>	caractère quelconque
<code>[ ]</code>	ensemble de caractères
<code>[^ ]</code>	complémentaire de l'ensemble
<code>*</code>	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.

## TEDIT (suite)

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 et Janick Taillandier.

**TICLEN**

TICLEN (TICK LENgth) définit la longueur des marques de graduation.

- |   |  |
|---|--|
| <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    |

**TICLEN** *longueur des marques*

**Exemple**

TICLEN 10

définit des marques de graduation de 10 unités de long.

**Paramètre d'entrée**

Élément	Description	Restrictions
longueur des marques	Expression numérique arrondie à un nombre entier.	-32767 à +32767

**Opération**

La longueur des marques de graduation s'exprime en unités graphiques.

Les marques sont tracées de part et d'autre de l'axe. L'argument de TICLEN spécifie la longueur totale des marques.

**Mots-clés associés**

XAXIS, YAXIS

**Auteur**

Pierre David

## XAXIS

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

Elément	Description	Restrictions
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
$x_{min}$	Expression numérique arrondie à un entier. Défaut : 0	-32767 à +32767
$x_{max}$	Expression numérique arrondie à un entier. Défaut : 639	-32767 à +32767

### Opération

XAXIS trace un axe horizontal à l'ordonnée  $y$ , de  $x_{min}$  à  $x_{max}$ .

Si le paramètre *espacement* est présent et non nul, des marques sont tracées toutes les *espacement* unités graphiques, en commençant par  $x_{min}$  si *espacement* est positif, par  $x_{max}$  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

# 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 *fichier*  
 XEDIT *fichier* , *chaîne de commandes*

## 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	Restrictions
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

## XEDIT (suite)

*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.

## XEDIT (suite)

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 occurrence 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.

## XEDIT (suite)

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\$".



## XEDIT (suite)

- 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 `"XX*"`, 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.

## XEDIT (suite)

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 une 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 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.

**XEDIT (suite)**

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.

**XEDIT** (suite)**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 ([↑], [↓], [↵] [↓] et [↵] [↑]). 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 I s'allume pour signaler que vous êtes en mode insertion, et la ligne courante est affichée. Lorsque vous taperez du texte et que vous appuierez 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 <i>saisie</i> sur la ligne courante
\$I	rentre en mode <i>saisie</i> avant la dernière ligne
/toto/T	rentre en mode <i>saisie</i> sur la prochaine occurrence de <b>toto</b>

## 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
.+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.

**XEDIT (suite)**

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 [O] 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.

## XEDIT (suite)

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
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 chaînes 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



**XEDIT** (suite)

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 et Janick Taillandier.

## YAXIS

YAXIS trace un axe vertical avec des marques de graduation optionnelles, à l'abscisse  $x$  spécifiée.

- |             |                        |
|-------------|------------------------|
| ■ Ordre     | ■ Exécution au clavier |
| ○ Fonction  | ○ Mode CALC            |
| ○ Opérateur | ■ IF...THEN...ELSE     |
|             | ■ 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

Elément	Description	Restrictions
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

## 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 et Michel Martinet.

## 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

## 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  
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