

TABLE DES MATIERES

. AVANT PROPOS	00
. CHARGEMENT DU PROGRAMME	01
. MAJUSCULES/ Minuscules	01
. TRANSLATEUR	03
. INSTRUCTIONS COMMUNES BASIC // PRINTER ET BASIC EDIT	05
. CARACTERES SPECIAUX	21
. COMMANDES VICTOR LAMBDA	23
. INSTRUCTIONS EXCLUSIVES BASIC EDIT	25
. INSTRUCTIONS EDITEUR DE TEXTE	26
. CARACTERES SPECIAUX SOUS EDITEUR	30
. MESSAGES D'ERREUR BASIC	31
. MESSAGES D'ERREUR EDITEUR	33

AVANT PROPOS

BASIC // PRINTER est destiné à la programmation sur VICTOR 1 16 K, équipé ou non de l'interface imprimante, et remplace le BASIC II.

BASIC-EDIT fonctionne avec VICTOR II 48 K et comporte un éditeur résident. De plus, il possède une nouvelle instruction LINE, une instruction PLOT étendue et une fonction USR beaucoup plus souple.

TRANSLATION

Le programme "TRANSLATEUR" permet l'utilisation de programme écrit en BASIC II sur des VICTOR 48 K bien que ceux-ci soient équipés d'un microprocesseur différent (Z80 au lieu de 8080 sur VICTOR 1). Il est également possible de lancer l'exécution d'un programme écrit en BASIC II sur un VICTOR chargé avec BASIC // PRINTER en chargeant également EZEDIT et en utilisant la fonction TRANSLATE. Cependant, quelques modifications devront être faites "à la main". Si vous utilisez USR, vous devrez modifier les lignes concernées pour vous adapter à la nouvelle syntaxe de cette instruction.

L'instruction TEXT a disparu ; Vous devrez la remplacer par LPRINT CHR\$(15). Vous référer à la page 3 pour l'emploi du translateur.

CARTES MEMOIRE

Sur le BASIC destiné aux VICTOR I et VICTOR I IMP (16K) l'interpréteur réside aux adresses 4CO0 - 4D21 et 6000 - 7FFF (huitième). Il utilise les adresses comprises entre 49AO et 4CO0 pour le buffer de clavier, la pile et diverses zones de travail. L'emplacement mémoire utilisateur commence donc en 4D22, la plus haute adresse disponible étant 5F9F.

L'interpréteur du BASIC-EDIT réside aux mêmes adresses (6000-7FFF) mais la zone de programmation est déplacée vers 8000 - FFOO laissant 32699 octets libres pour les programmes d'application. L'espace compris entre 4AO0 et 4FFF est occupé par des variables et des routines de l'interpréteur, alors que l'espace 5000 - 5F80 est utilisé par l'éditeur.

CHARGEMENT DU PROGRAMME

Insérez la cassette dans la platine cassette. Appuyez sur le bouton "RESET" (nouveau départ) puis sur la lettre "L". Rembobinez si besoin la cassette en appuyant sur la touche "REWIND". Appuyez sur la touche "READ" (lecture) pour charger le programme. Branchez ensuite les contrôleurs à main.

Ce BASIC niveau 2 vous permet de gouverner une imprimante du type parallèle. Bien que spécialement conçu pour fonctionner avec l'imprimante SEIKOSHA, il peut fonctionner avec n'importe quelle imprimante au standard "Parallèle CENTRONICs".

MAJUSCULES/ Minuscules

Un autre gros avantage de l'imprimante et de BASIC P.P. réside dans la possibilité d'imprimer en majuscules et en minuscules.

Le mode de fonctionnement est le même que sur une machine à écrire ordinaire. Maintenez "SHIFT" en tapant pour obtenir des majuscules et relâchez-le pour obtenir des minuscules. Vous pouvez aussi bloquer le clavier en majuscules par la touche "LOCK".

Les minuscules n'apparaissent pas sur l'écran mais elles s'impriment sur l'imprimante.

NB

Quand vous comparez deux chaînes de caractères, pensez que "A" et "a" ne sont pas des chaînes de caractères égales. Bien sûr, la variable "A" et la variable "a" ne sont qu'une seule et même variable.

Vous pouvez vous reporter au manuel de l'imprimante pour les informations spécifiques.

VICTOR II HR

Le modèle VICTOR II HR bénéficie de la haute résolution d'écran. Toutes les instructions propres à la résolution standard sont utilisables mais les affichages sont modifiés comme suit :

- Affichage textes : 22 lignes de 37 caractères
- Affichage graphiques : 231 lignes de 241 points visibles pour un réglage optimum du téléviseur :
 $x = [\emptyset, 240]$, $y = [\emptyset, 23\emptyset]$

Le paramètre minimum de l'instruction WINDOW est 11. L'annulation de la fenêtre s'effectue par WINDOW 230

NOTA :

VICTOR II HR permet d'utiliser la presque totalité des programmes du VICTOR I. Ceux-là étant en résolution standard, il est nécessaire de choisir, après un RESET, l'option R qui permet au VICTOR II HR d'émuler un VICTOR I. Les programmes en langage machine par VICTOR I et qui utilisent l'imprimante, ne sont pas directement compatibles avec VICTOR II.

EMPLOI DU TRANSLATEUR

Destiné à convertir des programmes écrits en BASIC II ou BASIC // PRINTER pour leur emploi sur des VICTOR II 48 K, le programme "TRANSLATEUR" accomplit les fonctions suivantes :

- Traduction des codes propres aux mots-clés,
- Déplacement du programme-utilisateur depuis l'adresse 4D22 (début de la zone utile en 16K) vers 8001 (début de la même zone en 48 K.).

MODE OPERATOIRE

Si vous avez un modèle Haute Résolution, passez d'abord en résolution standard (touche R).

- 1/ Chargez le programme "TRANSLATEUR" par la séquence RESET-L. (voir manuel utilisation).
- 2/ Après le chargement le menu suivant apparaît :
 - BASIC //
 - BASIC II
- 3/ Placez l'ancien programme dans le magnétocassette et chargez-le en appuyant sur la touche READ de la platine puis tapez 1 ou 2 selon le langage utilisé pour l'écrire.
- 4/ Le translateur traduit et translate le programme et vous avertit dès que la translation est achevée par le message "TRANSLATION TERMINEE".
- 5/ Placez une cassette vierge (correctement positionnée) dans le magnétocassette. Enfoncez simultanément les touches READ et WRITE puis tapez 1 ou 2 selon le nombre de copies souhaitées.

Si vous avez plusieurs programmes à translater, reprenez les opérations depuis le point 3 pour chaque programme à modifier.

NOTES

Vous pouvez utiliser la barre d'espacement pour commander le moteur du magnétocassette. Pour rembobiner, par exemple, enfoncez la touche REWIND, puis appuyez sur la barre d'espacement. Dès que le rembobinage est terminé appuyez de nouveau sur la barre d'espacement pour arrêter le moteur du magnétocassette.

- Si votre programme à modifier contient des instructions USR, vous devez les adapter manuellement à la nouvelle syntaxe de cette instruction. De même, vous devrez vérifier que les adresses pokées ne sont pas à modifier.
- Si votre programme original était écrit en Basic II, toutes les chaînes de caractères sont en majuscules. Vous devrez en tenir compte lors des demandes d'intervention par INPUT.

ATTENTION : Un programme Basic contenant des modifications de l'interpréteur risque de modifier "Translateur" dont les adresses sont comprises entre 6000 et 6270 (Hexadécimal)

INSTRUCTIONS COMMUNES

BASIC//PRINTER ET BASIC - EDIT

ABS

Renvoie l'argument en valeur positive (valeur absolue)

Exemple : Résultat :

1Ø PRINT ABS (6)	6
2Ø PRINT ABS (-6)	6
3Ø PRINT ABS (2* -3)	6

AND

Opérateur booléen liant plusieurs opérateurs ensembles, dont tous les termes doivent être vrais. Réalise une condition ET sur des valeurs binaires.

Exemple : Résultat:

1Ø A = 4 ; B = 1Ø	
2Ø IF A = 4 AND B = 1Ø THEN PRINT	
"BIEN"	BIEN
3Ø PRINT 5Ø AND 26	18

ASC

Renvoie la valeur décimale dans le code ASCII d'une chaîne de caractères. (Voir CHR\$)

Exemple : Résultat :

1Ø PRINT ASC ("W")	87
--------------------	----

ATN

Renvoie l'arctangente de l'argument en radians.

Exemple : Résultat :

1Ø PRINT ATN (.3Ø5669)	.29665
--------------------------	--------

CHR\$

Renvoie la chaîne de caractères indiquée dans l'argument par son code ASCII (Voir ASC)

Exemple :

1Ø PRINT CHR\$ (87)

Résultat :

W

CLEAR

Remet les variables à zéro. Utilisé avec un argument (entier) permet d'allouer aux variables caténiques une place mémoire spécifiée par l'argument. Par défaut, cette allocation est de 5Ø octets. (Voir FRE)

Exemple :

1Ø A = 6 ; B = 8

2Ø PRINT A ; B

3Ø CLEAR

4Ø PRINT A ; B

Résultat :

6 8

Ø Ø

CLOAD*

Transfère les données stockées sur une cassette dans une matrice déterminée.

Exemple :

1Ø DIM A (9)

2Ø CLOAD* (A)

Résultat :

Lit sur une cassette et charge en mémoire centrale les 10 éléments de A.

CLS

Efface l'écran.

Exemple :

1Ø CLS

Résultat :

Efface l'écran.

COLOR

Sélectionne une palette de 4 couleurs. Le premier chiffre donne la couleur de l'arrière plan. Les autres couleurs sont sélectionnées d'après leur position dans l'instruction selon le code suivant : Ø = Noir, 1 = Rouge, 2 = Vert, 3 = Jaune, 4 = Bleu, 5 = Magenta, 6 = Cyan, 7 = Blanc. (Voir OUTPUT et PLOT).

Exemple :

1Ø COLOR 4,7,3,1

Résultat :

Arrière plan : Bleu (code couleur 4)
1ère couleur : Blanc (code couleur 7)
2ème couleur : Jaune (code couleur 3)
3ème couleur : Rouge (code couleur 1)

COS

Renvoie le cosinus de l'argument en radians.

Exemple :

1Ø PRINT COS (.29665)

Résultat :

.956321

CSAVE*

Mémorise sur cassette les données d'une matrice déterminée. Appuyez, simultanément, sur les touches READ et WRITE du magnétophone avant exécution.

Exemple :

1Ø CSAVE* A

Résultat :

Les données de la matrice A sont mémorisées sur cassette après que l'utilisateur ait tapé l'instruction RUN.

DATA

Stocke en mémoire centrale des données lisibles par l'instruction READ. (Voir READ)

Exemple :

1Ø DATA VICTOR, 1
2Ø READ A\$, N
3Ø PRINT A\$; N

Résultat :

VICTOR 1

DEF (Non utilisable en mode direct)

Définit une fonction utilisateur.

Exemple :

10 DEF FNT (H) = H* 1.186
20 H = 100 : PRINT FNT (H)

Résultat :

118.60

DIM

Alloue des espaces mémoire pour le dimensionnement de tableaux pouvant aller jusqu'à 5 dimensions. La première position a l'indice zéro.

Exemple :

10 DIM A (6), B\$ (6,4)

Résultat :

Dimensionne un tableau A de 7 éléments numériques et un tableau B\$ de 7x5 éléments caténiques.

END (Non utilisable en mode direct)

Marque la fin d'un programme. L'utilisation est optionnelle si la dernière ligne du programme porte le plus grand numéro.

Exemple :

100 END

Résultat :

Arrête définitivement le programme à la ligne 100

EXP

Renvoie la valeur de e à la puissance de l'argument.
(e = 2.71828 base de la notation népérienne).

Exemple :

10 PRINT EXP (2.83321)

Résultat :

16.999

FIRE

Lit le bouton poussoir du contrôleur à main. FIRE (0) lit le bouton du contrôleur de gauche, FIRE (1) celui de droite.

L'instruction provoque un zéro si le bouton est pressé, un 1 s'il ne l'est pas.

Exemple :

```
10 IF FIRE () = 0 THEN GOTO 100
20 GOTO 10
100 PRINT "BONJOUR"
```

Résultat :

Affiche BONJOUR quand le bouton du contrôleur de gauche est pressé.

FOR... STEP

Initialise une boucle entre deux bornes. STEP (optionnel) fixe le pas de l'itération. Si STEP est omis le pas, par défaut, est de 1 (Voir NEXT).

Exemple :

```
10 FOR I = 2 TO 10 STEP 2
20 PRINT I ;
30 NEXT I
```

Résultat :

2 4 6 8 10

FREE

Suivi d'un argument numérique, renvoie la place mémoire disponible. Avec un argument caténique indique la place allouée aux variables chaîne de caractères. (50 par défaut). (Voir CLEAR)

Exemple :

```
10 PRINT FRE ()
10 PRINT FRE ("A")
```

Résultat :

4698 (avec 16K)
50

GOSUB

Exécute un sous-programme commençant au numéro de ligne indiquée. (Voir RETURN).

Exemple :

```
10 GOSUB 100
20 PRINT "FIN" : END
100 PRINT "BONJOUR" : RETURN
```

Résultat :

BONJOUR
FIN

GOTO

Fait sauter le programme au numéro de ligne indiquée.

Exemple :

```
10 PRINT "BONJOUR"  
20 GOTO 10
```

Résultat :

```
BONJOUR  
BONJOUR etc....
```

IF... GOTO

Etablit un test. Si la condition est vérifiée l'exécution du programme saute au numéro de ligne spécifié.

Exemple :

```
10 A = 1  
20 A = A + 1  
30 IF A = 5 GOTO 100  
40 PRINT "A < 5" ; GOTO 20  
100 PRINT "A = 5" ; END
```

Résultat :

```
A < 5  
A < 5  
A < 5  
A = 5
```

IF... THEN

Etablit un test. Si la condition est vérifiée l'instruction après THEN est exécutée.

Exemple :

```
10 A = 1  
20 A = A + 1:PRINT A  
30 IF A = 5 THEN PRINT "A=5" : END  
40 GOTO 20
```

Résultat :

```
2  
3  
4  
5  
A = 5
```

INPUT (Non utilisable en mode direct)

Permet d'entrer des données en mémoire centrale par le clavier.

Exemple :

```
10 INPUT "NOM,AGE" ; N$, A  
20 PRINT "NOM : "; N$  
30 PRINT "AGE : "; A
```

Résultat :

```
NOM, AGE? VICTOR, 2  
NOM : VICTOR  
AGE : 2
```

INSTR\$

Attend l'entrée au clavier du nombre de caractères spécifiés par l'argument. Affiche la chaîne de caractères une fois ce nombre atteint sans frappe de la touche RETURN.

Exemple :

```
1Ø PRINT "CODE POSTAL ?"  
2Ø CP$ = INSTR$ (5)  
3Ø PRINT CP$
```

Résultat :

```
CODE POSTAL ? 12345  
12345
```

INT

Renvoie le plus grand nombre entier de l'argument.

Exemple :

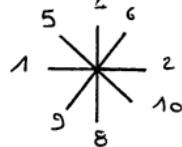
```
1Ø PRINT INT (3.999)  
2Ø PRINT INT (-4.0001)
```

Résultat :

```
3  
- 5
```

JOY

JOY (0) lit la position du manche à balais du contrôleur à main de gauche ; JOY (1) de celui de droite. L'instruction renvoie les codes selon ce graphique.



Exemple :

```
1Ø J = JOY (0)  
2Ø IF J = 1 THEN PRINT "1"  
3Ø IF J = 4 THEN PRINT "4"  
4Ø GOTO 1Ø
```

Résultat :

```
1  
4
```

LEFT\$

Isole dans une chaîne un caractère ou une sous-chaîne de caractères en partant de la gauche. L'argument donne la longueur de la sous-chaîne (1 par défaut)

Exemple :

```
1Ø A$ = "BONJOUR"  
2Ø B$ = LEFT$(A$, 3)  
3Ø PRINT B$
```

Résultat :

```
BON
```

LEN

Renvoie la longueur de l'argument.

<u>Exemple</u> :	<u>Résultat</u> :
1Ø A\$ = "BONJOUR"	7
2Ø PRINT LEN (A\$)	7
3Ø PRINT LEN ("BONJOUR")	

LET

Ordre d'affectation. Cette instruction optionnelle, sur la plupart des Basic, n'existe pas sur Basic//Printer et Basic-Edit.

<u>Exemple</u> :	<u>Résultat</u> :
1Ø A = 2	2
2Ø PRINT A	

LPRINT (Utilisable que pour sortie sur imprimante).

Permet l'écriture sur imprimante. Le ; supprime les intervalles. La virgule en met 8. Peut s'abréger en LP. .

<u>Exemple</u> :	<u>Résultat</u> :
1Ø LPRINT "VICTOR";"LAMBDA"	VICTOR LAMBDA
2Ø LPRINT "VICTOR";SPC (2); "LAMBDA"	VICTOR LAMBDA
3Ø LPRINT "VICTOR";TAB (8); "LAMBDA"	VICTOR LAMBDA
4Ø LPRINT "VICTOR","LAMBDA"	VICTOR LAMBDA

LOG

Donne le logarithme de base e (Voir EXP)

<u>Exemple</u> :	<u>Résultat</u> :
1Ø PRINT LOG (17)	2.83321

NEXT

Butée finale d'une boucle initialisée par l'instruction FOR (Voir FOR).

NEW Annule tout programme BASIC se trouvant en mémoire. Initialise tous les pointeurs internes de l'interpréteur propres à la répartition de la taille mémoire. C'est la première commande à donner après chargement de l'interpréteur.

MIDS

Isole dans une chaîne de caractères, un caractère ou une sous-chaîne de caractères entre deux butées en partant de la gauche vers la droite.

Exemple :

```
10 A$ = "BONJOUR"  
20 B$ = MIDS (A$, 4, 2)  
30 PRINT B$
```

Résultat :

JO

NOT

Opérateur booléen donnant le complément binaire de l'argument.

Exemple :

```
10 INPUT "NOMBRE A" ; A  
20 IF NOT (A) > 10 THEN PRINT "A < 10"  
30 PRINT NOT (-1)
```

Résultat :

NOMBRE A ? 5
A < 10
Ø

ON GOSUB

Instruction du branchement multiple vers des sous-programmes.

Exemple :

```
10 INPUT "A (ENTRE 1 ET 3) =" ; A  
20 ON A GOSUB 100, 200, 300  
30 PRINT "FIN" : END  
100 PRINT "LIGNE 100" : RETURN  
200 PRINT "LIGNE 200" : RETURN  
300 PRINT "LIGNE 300" : RETURN
```

Résultat :

A (ENTRE 1 ET 3) = ? 2
Ligne 200
FIN

ON GOTO

Similaire à ON GOSUB mais continue en séquence après le saut de ligne.

Exemple :

```
10 INPUT "A (ENTRE 1 ET 3) =" ; A  
20 ON A GOTO 100, 200, 300  
100 PRINT "LIGNE 100" : END  
200 PRINT "LIGNE 200" : END  
300 PRJNT "LIGNE 300" : END
```

Résultat :

A (ENTRE 1 ET 3) = ? 2
LIGNE 200

OR

Opérateur booléen associant plusieurs conditions.

Exemple :

1Ø A = 4 : B = 1Ø

2Ø IFA=1 OR B = 1Ø THEN PRINT "BIEN"

3Ø PRINT 5Ø OR 26

Résultats :

BIEN

58

OUTPUT

Affiche une variable mise en argument à une position donnée sur l'écran par les coordonnées x,y (117 x 77) et d'une couleur définie par l'ordre de la dernière instruction COLOR (Voir COLOR)

Exemple :

1Ø OUTPUT "BONJOUR", 5Ø, 3Ø, 2

Résultat :

BONJOUR affiché aux coordonnées x = 5Ø, y = 3Ø
Couleur : vert

PEEK

Permet de lire le contenu décimal d'une adresse choisie.
(Voir POKE)

Exemple :

1Ø POKE 24560 (Ø) : B = 1

2Ø A = PEEK (24560)

3Ø IF A < B GOTO 1Ø

4Ø PRINT "TOP"

Résultat :

Met environ 5 secondes
Pour afficher TOP

PLOT

Permet de tracer un point ou un rectangle sur l'écran. Cette instruction nécessite 5 valeurs mises en argument : coordonnées des x, coordonnées des y, Couleur, longueur, hauteur. La longueur et la hauteur, par défaut, prennent la valeur 1 et la couleur est celle définie par l'ordre de la dernière instruction COLOR rencontrée. (Voir COLOR)

Exemple :

1Ø PLOT 55,38,2
2Ø PLOT 2Ø,3Ø,1,25,25

Résultat :

Dessine un point à la position 55,38 de la couleur en position 2.

Trace un carré de 25 x 25 à partir de l'angle en bas à gauche situé aux coordonnées 2Ø,3Ø de la couleur en position 1

POKE

Permet de stocker une valeur décimale comprise entre 0 et 255 à une adresse choisie. Pour sélectionner les adresses comprises entre 32768 et 65536 il faut poker un nombre négatif de la différence entre l'adresse désirée et 64536

Exemple :

POKE 2ØØØØ,2Ø
POKE - 1ØØØ,31

Résultat :

Met 2Ø à l'adresse 2ØØØØ
Met 31 à l'adresse 6Ø44Ø

POINT

Indique la couleur d'un point aux coordonnées x,y spécifiées par rapport à la dernière instruction COLOR.

Exemple :

1Ø COLOR 4,7,3,1
2Ø PLOT 6Ø,4Ø,2
3Ø PRINT POINT (6Ø,4Ø)

Résultat :

2

POS

Renvoie la position où le dernier caractère vient d'être imprimé. L'argument n'a aucune influence.

Exemple :

1Ø PRINT "BIEN" ; POS (Ø)

Résultat :

BIEN 4

POT

Renvoie la valeur numériqueline sur le potentiomètre commandé par le bouton rotatif du contrôleur à main. POT (Ø) lit le contrôleur de gauche, POT (1) celuide droite. Permet d'utiliser les connexions des contrôleurs à main comme convertisseur analogique / numérique .

Exemple :

```
10 L = POT (Ø)
20 PRINT L : GOTO 10
```

Résultat :

Affiche une valeur numérique comprise entre les deux valeurs extrêmes du potentiomètre utilisé.

PRINT

Permet l'écriture sur l'écran. Le ; supprime les intervalles. La virgule en met 8. Peut s'abréger par ?

Exemple :

```
10 PRINT "VICTOR";"LAMBDA"
20 PRINT "VICTOR"; SPC (2); "LAMBDA"
30 PRINT "VICTOR"; TAB (8); "LAMBDA"
40 PRINT "VICTOR", "LAMBDA"
```

Résultat :

```
VICTORLAMBDA
VICTOR LAMBDA
VICTOR LAMBDA
VICTOR LAMBDA
```

LOG

Donne le logarithme de base e (Voir EXP).

Exemple :

```
10 PRINT LOG (17)
```

Résultat :

2.83321

READ

Attribue des valeurs aux variables stockées en DATA.
(Voir DATA)

Exemple :

```
10 READ A$,A,B$,B
20 PRINT B$,A,A$,B
100 DATA LAMBDA,1,VICTOR,2
```

Résultat :

VICTOR 1 LAMBDA 2

REM

Remarque ou commentaire documentant un programme. Est sans effet sur son déroulement.

Exemple :

```
10 REM PROGRAMME 1
```

Résultat :

RESTORE

Renvoie la pointeur du fichier interne (DATA) au premier élément. Peut être suivi d'un numéro de ligne.

Exemple :

```
1Ø READ A$,B$ : PRINT A$;B$  
2Ø RESTORE 11Ø  
3Ø READ A$,B$ : PRINT A$;B$  
1ØØ DATA PIERRE, JEAN  
11Ø DATA RENE,ROBERT
```

Résultat :

```
PIERREJEAN  
RENEROBERT
```

RETURN

Termine un sous-programme et renvoie à l'instruction qui suit immédiatement le GOSUB. (Voir GOSUB)

REWIND

Rembobine la bande magnétique de la cassette. Nécessite d'avoir enfoncé la touche REWIND du magnétophone. Après le rembobinage, appuyez sur n'importe quelle touche du clavier pour reprendre la main. S'utilise le plus souvent en mode direct.

RIGHT\$

Isole dans une chaîne de caractères, un caractère ou une sous-chaîne de caractères en partant de la droite. L'argument donne la longueur de la sous-chaîne. (1 par défaut)

Exemple :

```
1Ø A$ = "BONJOUR"  
2Ø B$ = RIGHTS (A$,4)  
3Ø PRINT B$
```

Résultat :

```
JOUR
```

RND

Initialise une liste de nombres pseudo-aléatoires compris entre Ø et 1.

Argument $> \emptyset$ = Renvoie un nombre de la liste

Argument = \emptyset = Renvoie toujours le même nombre (Utile pour la mise au point)

Argument $< \emptyset$ = Renvoie toujours le premier nombre de la liste

Exemple :

```
1Ø A = RND (1) * 1ØØ
2Ø A = INT (A) : ? A
3Ø GOTO 1Ø
```

Résultat :

Génère une série de nombres entiers compris entre Ø et 1ØØ

SGN

Renvoie le signe de l'argument.

Exemple :

```
1Ø PRINT SGN (-4837)
2Ø PRINT SGN (8 -8)
3Ø PRINT SGN (4837)
```

Résultat :

- 1
Ø
1

SOUND

Génère des sons divers dépendant de la valeur de l'argument. La première valeur doit être comprise entre Ø et 7, la seconde entre 1 et 32767. SOUND 7,4096 arrête toute production de son.

Exemple :

```
SOUND Ø,24844
SOUND 3,3Ø
SOUND 5.938
```

Résultat :

Sirène
Alarme
Ambulance

SPC

Génère des espaces dont le nombre est fixé par l'argument.

Exemple :

```
1Ø PRINT "BONJOUR" ; SPC (4); "MONSIEUR" BONJOUR....MONSIEUR
```

Résultat :

SQR

Renvoie la racine carrée de l'argument.

Exemple :

```
1Ø PRINT SQR (25)
```

Résultat :

5

STOP

Provoque l'arrêt de l'exécution du programme, la commande CONT le fait redémarrer, à moins qu'une modification soit apportée au programme après l'interruption.

Exemple :

```
10 PRINT "BONJOUR"  
20 STOP  
30 PRINT "MONSIEUR"
```

Résultat :

```
BONJOUR  
BREAK IN 20  
OK  
CONT  
MONSIEUR
```

STR\$

Convertit une valeur numérique en son images chaîne de caractères (selon le code ASCII). (Voir VAL)

Exemple :

```
10 A = 10  
20 B = A + 2 : PRINT B  
30 A = STR$ (A)  
40 B = A + 2 : PRINT B
```

Résultat :

```
12  
? TM ERROR IN 40
```

TAB

L'argument indique la position qui sera attribuée au premier caractère suivant le TAB.

Exemple :

```
10 FOR I = 1 TO 3  
20 PRINT TAB (I); "BONJOUR"  
30 NEXT I
```

Résultat :

```
BONJOUR  
BONJOUR  
BONJOUR
```

TAN

Renvoie la tangente de l'argument exprimé en radians.

Exemple :

```
10 PRINT TAN (.29665)
```

Résultat :

```
.305669
```

TONE

Instruction générant des sons musicaux. Le premier argument (positif et entier) est inversement proportionnel à la fréquence (plus il est petit plus le son est aigu). Le deuxième argument (positif et entier) indique la durée de la note.

Exemple :

1Ø TONE 168, 1ØØ
2Ø TONE 148, 1ØØ
3Ø TONE 131, 1ØØ
4Ø TONE 124, 1ØØ
5Ø TONE 11Ø, 1ØØ

Résultat :

DO
RE
MI
FA
SOL

USR

Permet d'appeler des sous-programmes écrits en langage machine. Poker l'adresse du sous-programme en 30499 et 30500, puis réaliser l'appel par USR. La partie haute de l'adresse va en 30500 et la partie basse en 30499.

Exemple :

1Ø POKE 30499, Ø : POKE 30500, 8Ø
2Ø USR

Résultat :

Exécute un sous-programme
situé à l'adresse 5ØØØ
(hexadécimal) -5Ø Héxa = 8Ø dé-
cimal

VAL

Convertit une chaîne de caractères en sa valeur numérique. Renvoie Ø si le premier caractère de la chaîne n'est pas un chiffre, un + ou un -.

Exemple :

1Ø A\$ = "1Ø" : B\$ = "A1Ø"
2Ø A = VAL (A\$) : B = VAL (B\$)
3Ø A = A + 2 : B = B + 2
4Ø PRINT A,B

Résultat :

12 2

WINDOW

Amenage une fenêtre fixe en haut de l'écran dont la hauteur est égale à 77 moins l'argument.

Le minimum étant 11.

Exemple :

1Ø WINDOW 38.5
2Ø PRINT "BONJOUR"
3Ø GOTO 2Ø

Résultat :

Fait défiler BONJOUR sur
la moitié inférieure de
l'écran

CARACTERES SPECIAUX

; Sépare plusieurs instructions sur la même ligne de programme.

_ Equivalent abrégé de PRINT.

RETURN (Touche rouge)

Touche de validation. Permet la prise en compte de l'instruction oude la commande par l'ordinateur.

BACK SPACE

Retour en arrière et effacement d'un caractère.

CARACTERES DE CONTROL

S'obtiennent en appuyant simultanément sur la touche CONTROL et la lettre précisée.

CONTROL - C

Interrompt le programme ou le listage en cours.

CONTROL - O

Supprime l'affichage pendant l'exécution d'un programme ou d'unlistage jusqu'à :

- Une instruction INPUT
- Une deuxième CONTROL-O
- La rencontre d'une erreur
- Retour en prode COMMANDE (Message OK)

CONTROL - Q

Continue l'exécution après une interruption par CONTROL - S

CONTROL - S

Permet l'interruption momentanée d'une exécution.

CONTROL - U

Empêche la prise en compte d'une instruction en cours par la mémoire.

COMMANDES VICTOR LAMBDA

CLOAD

Détruit le programme en cours, initialise toutes les variables et charge en mémoire centrale un programme stocké sur une cassette. Le programme est identifié par les cinq mémoires caractères du nom qui lui a été attribué. Si aucun nom n'est indiqué le premier programme rencontré est chargé.

Exemple :

CLOAD "TEST"

Résultat :

Chargé le programme nommé
TEST

CONT

Permet de reprendre l'exécution d'un programme interrompu par CONTROL - C ou STOP.

CSAVE

Sauvegarde un programme sur cassette. Lui attribue le nom spécifié par les cinq premiers caractères du nom entre guillemets.

ATTENTION ! Appuyez sur les touches WRITE et READ du magnétophone avant de valider cette commande.

Exemple :

CSAVE "TEST"

Résultat :

Sauve le programme nommé
TEST

EDIT (N'est utilisable qu'avec BASIC-EDIT)

Permet de passer du mode Basic au mode Editeur (voir instructions éditeur de texte). Le signe > OK vous indique que vous êtes sous éditeur. Le retour en mode Basic se fait par la commande "Basic" abrégée en B.

LIST

Liste le programme en mémoire. List suivi d'un numéro de ligne fait commencer le listage au numéro indiqué.

Exemple :

LIST

LIST 500

Résultat :

Liste le programme depuis le début

Liste le programme depuis la ligne 500

LLIST

Même effet que LIST mais provoque l'impression du programme sur l'imprimante.

NEW

Détruit le programme se trouvant en mémoire centrale et remet toutes les variables à zéro.

RUN

Lance l'exécution d'un programme. RUN suivi d'un numéro de ligne fait démarrer l'exécution à ce nouveau.

Exemple :

RUN

RUN 500

Résultat :

Lance l'exécution depuis le début

Lance l'exécution depuis la ligne 500

BASIC - EDIT

Instructions uniquement disponibles sur Basic-Edit ou à la syntaxe différente.

LARGE

Fait passer l'imprimante aux caractères double largeur.
Pour revenir à l'impression standard, tapez LPRINT CHR\$ (15).

Exemple :

```
1Ø LPRINT "BONJOUR"  
2Ø LARGE  
3Ø LPRINT "BONJOUR"
```

Résultat :

BONJOUR
BONJOUR

LINE

Permet de tracer la meilleure approximation d'une ligne droite entre deux points.

Exemple :

```
1Ø LINE 1Ø,15,1ØØ,6Ø,2
```

Résultat :

Trace une ligne de points
situé aux coordonnées x = 1Ø,
y = 15 vers le point situé
x = 1ØØ, y = 6Ø de la couleur
de position 2 (Voir COLOR)

USR

Permet d'appeler des sous-programmes écrits en langage machine. Ils sont appelés directement par USR, adresse en décimal et se terminent obligatoirement par POPH - RET (c 9H - 201D)(E1H-225D)

Exemple :

```
1Ø USR 2Ø48Ø
```

Résultat :

Exécute un programme dont le début est situé à l'adresse
Hexa 50ØØ

INSTRUCTIONS
EDITEUR DE TEXTE

Toutes les commandes de l'éditeur de texte peuvent être abrégées à condition que l'abréviation permet l'identification de la commande sans ambiguïté. Utilisez un point (.) à la fin de toutes commandes abrégées : LI. (LISTE) ; SUB. (SUBSTITUTE), LOC. (LOCATE) etc...

Certaines commandes ont plusieurs formes syntaxiques. Elles sont décrites à chaque fois. La forme la plus courte utilise des valeurs par défaut indiquées dans la description.

AUTO, AUTO n, AUTO n,m

Assure le numérotage automatique des lignes d'instruction. Pour arrêter le numérotage automatique tapez RETURN (touche rouge) après l'apparition du nouveau numéro de ligne. Le paramètre m donne le pas du numérotage. Par défaut, la première ligne porte le numéro 10 et le pas est de 10 en 10. La plus haute valeur disponible est 65529. notez que si vous aviez déjà un numéro de ligne attribué, la numérotation automatique de ce même numéro aura pour effet le remplacement de l'ancienne ligne.

Exemple :

AUTO 50,5

Résultat :

Numérotation obtenue
50,55,60,65,...

APPEND APPEND "NOM"

Annexe un programme provenant de la cassette à un programme déjà en mémoire. Les lignes annexées sont renumérotées automatiquement au pas de 10 et commencent à un numéro supérieur de 10 au dernier numéro de ligne du programme en mémoire. Les GOTO et GOSUB contenus dans la partie annexée sont recalculés. L'absence de nom après APPEND provoque l'annexion du premier programme rencontré sur la cassette. Si un nom est spécifié l'éditeur recherchera

ce programme pour l'annexer. Au cas où aucun programme du nom recherché n'existe sur la cassette, vous devrez annuler la commande APPEND par RESET puis R.

Pour utiliser APPEND, positionnez la cassette, appuyez sur la touche READ du magnétophone et tapez APPEND. Après le chargement, l'écran s'éteint quelques secondes, le temps du numérotage des lignes. Vous ne pouvez réaliser une annexion que si la programme résident occupe moins de la moitié de la mémoire totale disponible sur votre configuration, et que le programme annexé ne dépasse pas la place libre restante. Utilisez l'instruction PRINT FRE (Ø) pour connaître le nombre d'octets disponibles.

Exemple :

APPEND "PROG"

Résultat :

Recherche le programme nommé "PROG" et l'annexe au programme résident en renumérotant les lignes de "PROG"

BASIC

Retour du mode éditeur au mode Basic.

CLOAD CSAVE

Identiques aux commandes similaires sous Basic.

DELETE n - m

Détruit les lignes comprises entre les deux butées n et m incluses. L'annulation d'une fausse ligne s'obtient par la frappe de son numéro suivi de RETURN (touche rouge).

Exemple :

DELETE 5Ø - 12Ø

Résultat :

Détruit les lignes 5Ø à 12Ø

EXTRACT n.. EXTRACT n - m

Détruit toutes les lignes d'un programme sauf celles spécifiées par n ou comprises entre n et m inclus. Cette instruction

est particulièrement utile pour isoler un groupe de lignes et l'annexer à un autre programme.

Exemple :

EXTRACT 130

EXTRACT 50 - 170

Résultat :

Détruit toutes les lignes du programme sauf la ligne 130

Détruit toutes les lignes du programme sauf celles comprises entre 50 et 170 incluses

FRE

Affiche le nombre d'octets disponibles en mémoire, et le nombre déjà utilisés.

LAST

Affiche le numéro de la dernière ligne utilisée.

LOCATE "texte" LOCATE n - m "texte"

Affiche le numéro de ligne contenant le texte spécifié. L'adjonction des paramètres n et m restreint les recherches entre ces deux bornes. ATTENTION ! Les espaces utilisés dans les chaînes de caractères ou l'emploi de majuscules/minuscules sont significatifs. Respectez la concordance entre le texte spécifié et recherché. Les mots-clés utilisés par le Basic sont des mots réservés. Pour localiser un texte contenant un mot-clé, vous devez faire précéder le mot-clé par le signe  obtenu en appuyant sur la touche TAB. Les opérateurs mathématiques sont des mots-clés.

Exemple :

LOCATE "T.V.A."

LOCATE 20 - 100 "17.6"

Résultat :

Affiche tous les numéros de lignes contenant le texte "T.V.A."

Affiche tous les numéros de lignes compris entre 20 et 100 inclus et comportant le texte "17.6"

LOCATE " GOTO"

Affiche tous les numéros de lignes comportant l'instruction GOTO

RESEQUENCE RESEQUENCE n RESEQUENCE n - m

Change les numéros de lignes du programme. Le paramètre n indique le premier numéro de ligne à renommer, m indique le pas. Par défaut, n et m prennent la valeur 100 - Les GOTO, GOSUB, THEN, ON sont recalculés, cependant si plusieurs GOTO ou GOSUB se trouvent sur la même ligne, seul le premier est modifié. Il n'est pas possible de reséquencer un programme de plus de 15 K octets.

REWIND Identique à la commande similaire sans Basic.

SUBSTITUTE "nouveau texte" FOR "ancien texte"

SUBSTITUTE n "nouveau texte" FOR "ancien texte"

SUBSTITUTE n - m "nouveau texte" FOR "ancien texte"

Modifie le programme en remplaçant une chaîne de caractères par une autre. Le paramètre n entraîne la modification que de la ligne précisée. Les paramètres n et m marquent les butées de la zone à modifier. Les espaces, minuscules et majuscules sont significatifs, aussi respectez leur position. Si dans une même ligne il y a deux fois le texte à modifier, seul le premier rencontré le sera. Dans ce cas, vous devez donner un deuxième ordre semblable. De même que dans l'instruction LOCATE vous devez faire précéder les mots-clés du Basic à modifier du signe  obtenu par la touche TAB.

Exemple :

SUBSTITUTE "ABC" FOR "DEF"

SUBSTITUTE 50 "AB" FOR "CD"

SUBSTITUTE 50 - 100 "AB" FOR "CD"

120 PRNT "BONJOUR"

SUBSTITUTE 120 " PRINT" FOR "PRNT"

SUBSTITUTE 50 - 100 " GOSUB"
FOR " GOTO"

Résultat :

Remplace DEF par ABC dans tout le programme

Remplace CD par AB ou ligne 50

Remplace CD par AB dans les lignes comprises entre 50 et 100 incluses.

Remplace le PRNT erroné de la ligne 120 en PRINT. 4  n'est nécessaire que pour le mot-clé PRINT

Remplace GOTO par GOSUB dans les lignes 50 à 100 incluses

CARACTERES SPECIAUX SOUS EDITEUR

. (Point)

Utilisé pour abréger une commande.

CONTROL - C

Stoppe un listage et redonne la commande au programme.

CONTROL - S

Arrête temporairement le listage ou l'effet de LOCATE.

CHIFFRES DE 1 A 9

Utilisé après un CONTROL - S fait reprendre le listage d'un nombre de lignes égal au chiffre tapé.

Exemple :

CONTROL - S 3

Résultat :

Affiche trois lignes après le CONTROL - S

CONTROL - Q

Annule l'effet d'un CONTROL - S.

LISTE DES MOTS RESERVES

ABS, AND, ASC, ATN, CHR\$, CLEAR, CLOAD, CLS, COLOR, CONT, COS, CASVE, DATA, DEF, DIM, DUMMY, EDIT, END, EXP, FIRE, FN, FOR, FRE, GOSUB, GOTO, IF, INP, INPUT, INSTR\$, INT, JOY, LARGE, LEFT\$, LEN, LET, LINE, LIST, LLIST, LPRINT, LOG, MID\$, NEW, NEXT, NOT, ON, OR, OUTPUT, PEEK, PLOT, POKE, POS, POT, PRINT, READ, REM, RESTORE, RETURN, REWIND, RIGHTS\$, RND, RUN, SGN, SIN, SOUND, SPC, SQR, STEP, STOP, STR\$, TAB, TAN, THEN, TO, TONE, USR, VAL, WINDOW.

MESSAGES D'ERREUR

BASIC

Syntaxe du messsage d'erreur :

xx ERROR IN n

xx = Code erreur

n = Numéro de ligne

BS (Bad Subscript)

Argument incorrect (Trop grand, négatif etc)

CN (Can't Continue)

La commande CONT est sans effet.

DD (DIM)

Tableau à redimensionner.

FC (Illégal Fonction call)

Fonction interdite.

ID (Illégal Direct Mode)

Mode direct interdit.

LS (String Too Long)

Chaine de caractères trop longue.

MO (Missing Operand)

Opérateur manquant.

NF (Next without For)

Next sans For dans une boucle.

OD (Out of Data)

Erreur ou insuffisance de données dans un DATA.

OM (Out of Memory)

Dépassement de capacité mémoire.

OS (Out of String space)

Chaine trop longue pour le dimensionnement alloué.

OV (Overflow)

Dépassement capacité numérique.

RG (Return without Gosub)

Return sans Gosub correspondant.

SN (Syntax Error)

Erreur de Syntaxe.

ST (String Formule too complex)

Expression ou fonction de chaine trop longue ou trop complexe.

TM (Type Mismatch)

Confusion entre variable numérique et variable chaine de caractères.

UF (Undefined User Fonction)

Fonction Utilisateur non définie.

1/0 (Division by Zéro)

Tentative de division par zéro.

MESSAGES D'ERREUR

EDITEUR

CHAINE ABSENTE :

La chaine demandée n'existe pas.

NOMBRE TROP GRAND :

Appel à un numéro de ligne dépassant 65529/

CHAINE ERROREE :

Problème de mot-clé dans une chaine.

DEPASSEMENT DE CAPACITE :

Dépassement de capacité mémoire.

COMMANDE ERROREE :

Commande n'existant pas dans l'éditeur.

LIGNE ABSENTE :

Ligne n'existant pas dans le programme.

GAMME VICTOR lambda

	VICTOR I	VICTOR II	VICTOR II IMP HR
1. MICROPROCESSEUR	8080	Z 80	Z 80
1. HORLOGE	1,7 MHz	1,7 MHz	5,1 MHz
1. TEMPS MOYEN D'ACCES	250 ns	250 ns	125 ns
1. ROM	2 K	2 K	4 K
1. RAM TOTALE	16 K	48 K	64 K (48 + 16)
1. RAM SOUS BASIC	4,5 K	32 K	16 K
1. RAM SOUS ASSEMBLEUR	13 K	45 K	45 K
1. RESOLUTION	113 x 77	113 x 77	241 x 231
1. AFFICHAGE (lignes x caract.)	12 x 17	12 x 17	22 x 37
1. MAJUSCULES-MINUSCULES	non	non	oui
1. COULEUR (demi luminosité)	8 x 2	8 x 2	8 x 2
1. GENERATEUR DE SONS (8000)	oui	oui	oui
1. GENERATEUR DE NOTES (4 octaves)	oui	oui	oui
1. CLAVIER	53 touches alphanumériques	53 touches alphanumériques	53 touches alphanumériques
1. MAGNETOCASSETTE INTEGRE	oui	oui	oui
1. VITESSE DE TRANSFERT	1500 bauds	1500 bauds	1500 bauds
1. CONNEXION POUR CAM (convertisseur A/N)	2	2	2
1. SORTIE IMP PARALLELE CENTRONIC	oui EN OPTION	oui	oui