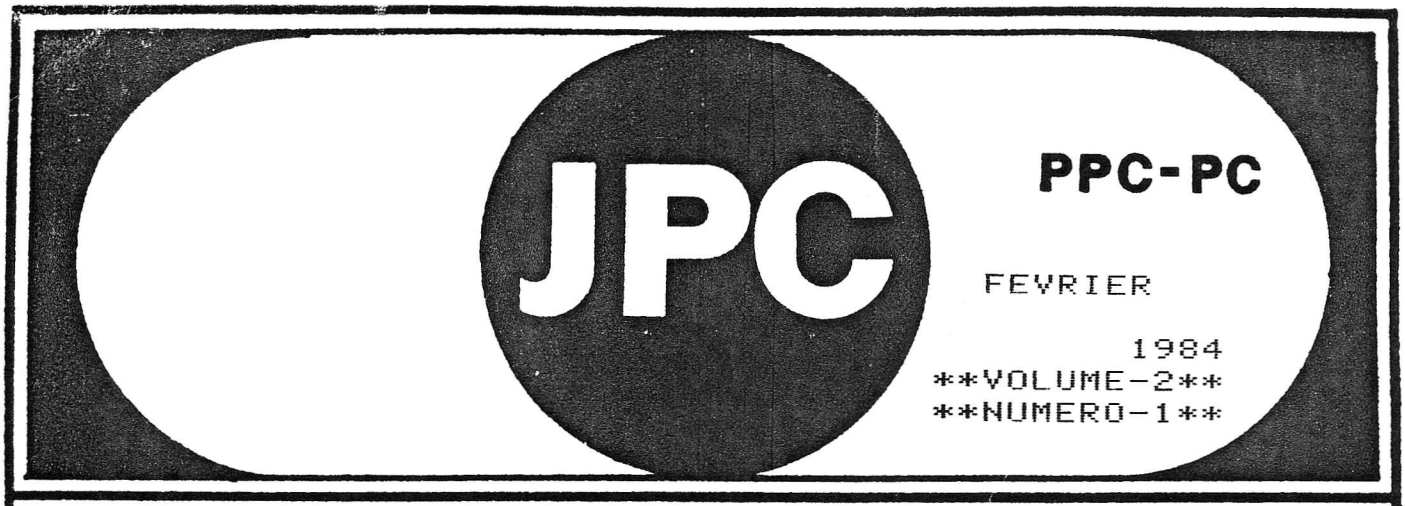


Edité par PPC-PC

56,rue J.J.Rousseau,75001 PARIS

Chers AMIS

une nouvelle de l'Angleterre, m'annonce qu'ils organisent une réunion européenne le premier week end de mai. veuillez m'écrire et me signaler qui sera interressé par ce voyage . Vous aurez trés prochainement des nouvelles de cette réunion mais ils ont besoin de savoir le nombre de français qui iront pour la fin du mois de mars; Je leur répondrai le 25 mars trés éxactement.

PHILIPPE

Sont en vente au club:

des cartes magnetiques au prix des 300FF les 100 cartes
des EPROMS au prix de 55FF pièce , 14EPROM vierge
le VASM (listing des ROMS 0, 1, 2 en mnémoniques HP) au prix de 150FF (340 pages)

Un monde inconnu : HEWLETT-PACKARD
Habitants : des HEWLETTISTES-PACKARDIENS.

$(x+y)^2$

$x+y = \sqrt{x^2+2-yx}$

LES PREMIERS PAS DE BEBE

Le REGistre e

Plaçons par exemple, le POINTeur sur la ligne 6Ø d'un PRGM situé en MEM (mode RUN GTO "nom du PRGM" GTO .Ø6Ø) RCL e HEQ "NH" (JPC N6P6). Les 3 DIG de Droite seront: Ø3C qui est le N° de la ligne du PRGM (HEX Ø3C = DEC 6Ø). Vous pouvez vérifier avec toute autre valeur.

Taper le PRGM suivant: LBL "LP" CLA 8 XTOA RCL M STO e STOP BEEP et l'XEQécuter. Vous constatez, en mode PRGM, que le POINT. s'est bien placé sur la ligne Ø8 mais sans l'XEQécuter. Si vous aviez des ASN sur des touches SHIFTées, elles sont perdues !

ASN la FS 'STO M' à la touche -34 (145 ENTER↑ 117 ENTER↑ 34 CHS XEQ "XYZASN", par exemple) puis RCL e XEQ "NH". Nous obtenons ØØ4 comme premiers DIG en ALPHA et '4' indique que la touche -34 est ASN. Notre '41' ayant 35 touches, 36 BITs (9 DIG) ont été utilisés pour représenter (répertorier) la situation des touches (ASN = 1, non ASN = Ø).Vous trouverez ci-joint un tableau représentant les 36 premiers BITs du REG e (ou ⊢). Le 2ème rang de CHiffres indique les N°s de touches sous les N°s de BITs correspondants. C'est le BIT 24 qui est inutilisé et qui se trouvera donc toujours à Ø. A droite de ce tableau, j'ai représenté les N°s de BITs implantés sur le clavier pour faire apparaître la disposition adoptée. Je dois signaler que ce "papier" n'a pu être écrit que grâce à la lecture du Manuel de la PPC-ROM (PRGM "MK") qui est une véritable mine d'or (492 pages) et qui est écrit d'une façon très claire (heureusement pour mon 'anglais' minable !).

Si nous reprenons l'exemple précédent, le BIT Ø9 sera levé (à 1) et, sur 36 BITS, nous obtenons l'équivalent HEX ØØ4ØØØØØØ (9 DIG). Là, il faut avouer que la HP-16C est bien utile.....

Si nous prenons l'exemple mentionné dans la dernière rangée du tableau, les touches 11, 12, 13, 14 & 15 sont ASN; la VAL BIN correspondante sera donc: ØØØ1ØØØØØØØ1ØØØØØØØ1ØØØØØØØ1ØØØØØØØ1 ce qui donne en HEX: 1Ø1Ø1Ø1Ø1et vous pourrez vérifier.....

Les CODEs des F sont STO dans les REG d'ASN dont la 1ère ADD ABS est DEC 192. L'introduction de ces CODEs se fait de D à G et le dernier entré pousse le précédent vers la G, celui de G étant alors "poussé" dans la partie D du REG supérieur et ainsi de suite. Chaque REG peut contenir 2 ASN de la façon suivante : (ceux qui ne disposent pas d'un LECT d'EPROM et du jeu d'EPROM '-MLEPROM 1H' de Stéphane BARIZIEN, devront me faire confiance)

192 CHS XEQ "NRCL" XEQ "NH"

AFFichage : FØØØØØØØ91753B ALPHA

où 'F' est une constante indiquant qu'il s'agit de CODEs 'ASN', les 2 derniers DIG '3B' indiquent le CODE de la touche suivant tableau Fig. 5 du bouquin de JDD (équivalent de -34) et, HEX 91 = DEC 145 qui est le préfixe de STO M et HEX 75 = DEC 117 qui est le suffixe. A la prochaine ASN, les 6 DIG de D seront poussés de 8 'crans' vers la G et, le 2ème DIG étant inutile, sera toujours Ø.

Signalons que les ASN de LBL ALPHA globaux sont STO avec le LBL correspondant et c'est donc "gratuit"; il suffit de passer en mode USER .....

Lorsque nous effaçons une ASN, seul le REG "répertoire" (e ou ⊢) est affecté (le BIT adéquat est mis à zéro), les CODEs STO ne sont pas supprimés, ce qui occupe inutilement un demi REG par ASN. Nous verrons ultérieurement avec le PRGM "PK" (Pack Key) de la toujours aussi fameuse PPC-ROM, comment nous pouvons récupérer cette place perdue.

Vous voici, je l'espère, en possession des éléments indispensables qui pourront vous permettre de décortiquer par exemple, un PRGM tel que "MXX" de Tapani TARVANEN & Clifford STERN.....

Je dédie ce "papier" à JDD dont le bouquin m'a également été d'une grande utilité car écrit dans ma demi-langue maternelle.....

rS (T178+P2Ø)

P.S.: Un beau sujet de PRGM pour ceux qui ont compris (?) : "PKX".....

35	27	19	11	3
34	26	18	1Ø	2
33	25	17	9	1
32		16	8	Ø
31	23	15	7	
3Ø	22	14	6	
29	21	13	5	
28	2Ø	12	4	

9 premiers DIGits des REGistres ⊢ & e (répertoire des ASN)

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
44	43	52	51	58	47	46	45	44	43	42	41	48	37	36	35	34	33	32	31	38	27	26	25	52	3	2	1	28	29	30	31	32	33	34	35
											X								X								X									X

En cas d'erreurs ou d'omissions, veuillez m'en faire part.

Le REGistre Q

En mode PRGM, taper LBL "TREBOR" puis en mode RUN: XEQ "TREBOR" RCL Q CLA STO M et passer en mode ALPHA.....

Le REG Q STO donc le nom de la dernière F XEQécutée mais à l'envers. Je laisse le soin à un collègue dévoué, de traiter de l'utilisation du Q-LOADER (CODEs DEC 27 & Ø).

Ce REG est également utilisé comme "brouillon" à maintes occasions, ce qui interdit pratiquement de l'utiliser en PS.

Je ne pense qu'il soit nécessaire d'en dire plus sur ce REG dont le nom à lui seul, est tout un programme..... *rS (T178+P2Ø)*

Q-LOADER

Les CODEs DEC pour ASN le QL sont 27 & Ø. Dans la foulée, ASN STO Q (145 & 121) et RCL M (144 & 117). Ceux qui désirent "rester Français" peuvent utiliser à la place du QL, la FS 'bêta' dont les CODEs DEC sont 1 à 16; les résultats sont les mm.

En mode RUN, introduire en ALPHA "TREBOR" (vous connaissez déjà !) et toujours en mode RUN: RCL M STO Q puis passer en mode PRGM QL vous devez obtenir 'E' ('Ø' avec 'bêta') et SST aura introduit une CHaîne qui est l'inverse de 'TREBOR'. Vous avez donc compris que cette méthode permet de créer une CH de CAR en mode PRGM. La F XTOA permet l'introduction des CAR non accessibles directement au clavier et il faut introduire les CODEs en allant de D à G, c'est-à-dire, en sens inverse. Le 'E' (ou le 'Ø') est en prime et, si vous n'en avez pas l'utilité, il suffira de l'effacer. Voici je l'espère, dévoilé le "secret" du Q-LOADER qui n'est réservé, bien entendu, qu'aux "sportifs".....

Des collègues 'sportifs' m'ont demandé à qoi pouvaient servir les FS dont les CODES DEC pour ASN sont compris entre 16 & 25 et qui donnent, avec 1 comme préfixe, Ø à 9 en mode PRGM. A ma connaissance, elles ne servent qu'à introduire ces CHiffres, en association avec E, par exemple. Ce sont en fait, toutes des QL mais dont le CH peut être "exploité". Pour obtenir E12, par exemple, taper USER EEX (XR 44,27=DEC 27,27ASN/43) et ensuite: '1' & '2' effacer les 3 "merdes" suivantes et vous obtiendrez très rapidement ce que nous désirions..... OK ?

Afin que je ne risque pas de perdre mon temps inutilement, dites-moi si cela vous plait !

MERCI de votre attention,rS (T178+P2Ø)

Bondy, le 30.10.83

Le REGistre c

Nous allons décortiquer ce fameux REG réputé dangereux afin de pouvoir le dompter.

Ceux qui ne disposent pas de la PPC-ROM devront rentrer les PRGM "NH"+"HN" (JPC N6P6) et "C?" (JPC N5P2).

ASN la FS RCL c à la touche C (13) :
144 ENTER↑ 125 ENTER↑ 13 XEQ "1K" ou "MKX" ou ....

SIZE ØØØ SCI 9 RCL c XEQ "NH"
AFFichage : $2ØBØØ1692ØØ1CF_{ALPHA}$

ne vous affolez pas si vous n'optenez pas exactement ces DIC ! c'est normal !

Si vous n'avez pas utilisé la F ΣREG,le 1er REG STAT doit être le REG 11 mais avec SIZE ØØØ, toute tentative de Σ+, Σ-, ou CLΣ donnera NONEXISTENT. Les 3 premiers DIG '2ØB' indiquent l'ADD HEX du 1er REG STAT et, avec SIZE ØØØ, nous devons obtenir 2ØB = DEC 523 car: 523-512=11 : Souvenez-vous que nos 319 REG sont compris entre les ADD DEC 193 & ....512 (JPC N5P2) !

SIZE Ø3Ø RCL c XEQ "NH" nous donnera l'ADD 1ED = DEC 493 et 523-493=3Ø= notre nouveau SIZE. Nous verrons plus loin ce qui s'est passé.

Les 3 derniers DIG trouvés en ALPHA à la suite de "NH" représentent l'ADD HEX du .END. permanent qui se trouve toujours à la fin du dernier PRGM situé en MEM principale. Si le dernier PRGM est PACKé (par GTO .. par exemple), ce .END. (entre 2 points) apparaîtra dans un CATALOG 1, à la suite du dernier END (pas de points) et si le dernier PRGM n'est pas PACKé, le .END. permanent sera le 'END' de ce dernier PRGM.

Si nous prenons l'exemple ci-dessus, 1CF avec SIZE ØØØ, est égal à DEC 463 et, 511-463=48 et 48 est le nbre de REG occupés par le (ou les) PRGM situé en MEM principale. Pour mon exemple, j'ai un CAT 1 avec un PRGM de 166 Octets et un de 169 Octets, d'où: 166+169= 335 et 335/7=48 (INT supérieur). OK ?

Mais que se passe-t'il lorsque nous modifions le SIZE ? Nous déplaçons simplement le RIDeau et de SIZE ØØØ à SIZE Ø3Ø, nous avons "baissé" le RID pour réserver, en Haut de la MEM, la place des 3Ø REG (de ØØ à 29, le REG ØØ étant en Bas, à la limite du RID). Les REG 'PRGM' se trouvent donc décalés vers le Bas.

HEX 2ØØ = DEC 512 représente l'ADD du REG ØØ qui est l'ADD extrême supérieure avec SIZE ØØØ. Avec SIZE Ø3Ø, vous devez trouver 1E2=DEC482 car 512-482=3Ø . OK bis ?

Mais pourquoi, avec SIZE Ø3Ø, trouve-t'on 1ED=DEC 493 comme ADD du 1er REG STAT ? 512-493= 19 et 19+11=3Ø . Là il n'est pas certain que vous ayiez saisi l'astuce qui est pourtant bien simple.

Les REG de données (SIZE) sont placés de Bas en Haut, le REG ØØ étant placé en Bas, comme indiqué précédemment.

J'espère que le REG c n'a maintenant plus aucun mystère pour vous et nous allons passer à une application qui me semble intéressante et dont je vous avais parlé dans un précédent JPC (N?P?).

Taper le PRGM suivant : LBL "Σ/" RCL c CLA STO M ATOX ATOX 2 XTOA 8Ø XTOA -2 AROT RCL M STO c et n'hésitez pas à l'XEQécuter ! Essayez ensuite de faire Σ+ ou Σ- ou CLΣ . Le message NONEXISTENT est normal car nous avons placé les REG STAT dans la partie "vide" de la MEM, entre les REG d'ETAT et ceux du système X-FUNCTIONS.

Je ne vais pas vous indiquer ce que fait le PRGM ci-avant mais simplement vous donner ce qui est introduit à la place des 2 premiers Octets du REG c : Ø2 8Ø car Ø28 est l'ADD HEX à laquelle nous adressons le 1er REG STAT.....

Vous venez de constater qu'il est parfaitement possible de manipuler le REG c mais à la seule condition de ne pas toucher au terrible 169.....

J'allais oublier de vous indiquer comment replacer les REG STAT dans une condition "normale". Il suffit simplement de faire : XEQ "ΣREG 11 et tout rentre dans l'ordre.

L'ADD du .END. permanent peut être obtenue avec le PRGM "E?" de la PPC-ROM, l'ADD du RID avec "C?" publié dans JPC N5P2&3 (ces 2 ADD étant les ADD DEC ABS) et "Σ?" , toujours de la PPC-ROM, donne le N° du 1er REG STAT.

La F "CX" ou "CU" (cf JPC N5P2) permet de déplacer le RID (Curtain, en anglais) autrement qu'avec SIZE et j'avais lancé un appel afin qu'un collègue se dévoue pour nous rédiger un article sur cette possibilité intéressante. Si vous avez assimilé les subtilités du déplacement du RID, vous pouvez tout découvrir (?) en faisant des essais. Placer en x l'ADD DEC où vous voulez placer le RID, XEQ "CX" et observez ! Un MEMORY LOST n'a jamais fait des dégats irrémédiables.....

Ecrivez-moi ou téléphonez-moi pour m'indiquer si je dois continuer dans cette voie !

Joyeuses recherches, *rS (T178+P2Ø)*

```
01◆LBL "Σ/"
02 RCL c
03 CLA
04 STO [
05 ATOX
06 ATOX
07 2
08 XTOA
09 80
10 XTOA
11 -2
12 AROT
13 RCL [
14 STO c
15 END
```

Bondy, le 23.10.83

END et COMPilation.

Tous ceux qui ont apprécié le PRGM "COMP" de Frédéric POUPON (JPC N5P28&40) se seront aperçus qu'il y avait un problème lorsque nous voulions transférer ce prgm sur CART, sur K7 et éventuellement sur "CB" (?).

Le problème rencontré provient du fait que le END "PACKé" fabriqué avec le PRGM "PP" de Daniel JACOB (JPC N5P15) n'est pas conservé (reproduit) à la suite d'un transfert du PRGM (CART, CB, K7). Je pense que vous aviez tous (?) remarqué cette particularité fort gênante et j'espère que le MC (MicroCode) permettra de créer une Fonction (WPP, par exemple) qui permettrait de conserver un END "Packé" avec le LECT de CART. Donc, avis aux Amateurs; Stéphane BARIZIEN pourrait nous donner son avis à ce sujet.

Avec le LECT de K7, la F "WRTA" (équivalente à "WALL") restitue les (ou le) PRGM tels qu'ils étaient lors du "WRTA", c'est-à-dire "PACKés" (END) ou non "PACKés" (.END.). Compte tenu de la capacité d'une K7, j'avoue que j'utilise beaucoup "WRTA" qui a en plus l'avantage de sauvegarder, donc de restituer, tout le reste.....

Avec les CB, il est possible, en refaisant la dernière RANGée (et qqf, l'avant-dernière), de "fabriquer" un END "sur mesures" mais je dois avouer que cela n'est pas facile et demande un certain temps (le même temps que peut mettre le fût du canon pour refroidir....). Si cela vous intéresse, je pourrais essayer de vous expliquer comment il faut procéder. Je crois que j'aurais dû écrire : "comment je procède" car il y a peut-être une autre solution (?). Vous comprendrez donc que pour l'instant je ne reproduirai pas en CB des PRGM dont les GTO et les XEQ seront COMPilés et je vous demande de ne pas m'en envoyer car cela m'obligerait à réintroduire les LBL (déCOMPiler ?), ce qui représente dans certains cas, un travail important et fastidieux. Ce n'est pas Frédéric qui me contredira !

J'en profite pour indiquer à Frédéric que nous attendons avec impatience son PRGM "SuperCOMP" qui acceptera les GTO et XEQ INDirects.

Heureuses Prgmmations COMPilées, (rS=T178+P2Ø)

Toujours en application de ee feuilleton sur les premiers pas de bébé, nous allons voir d'autres programmes vous permettant de mieux comprendre le fonctionement interne de la machine.

```
   PRP "ID55"

01♦LBL "ID55"
  02 RCL d
  03 CLA
  04 STO [
  05 STO \
 06 "┣+++++++"
  07 RCL [
  08 STO d
  09 FC?C 07
  10 SF 07
  11 FS? 07
  12 SF 01
  13 FC? 07
  14 CF 01
  15 RCL d
  16 STO [
  17 "┣*"
  18 RCL \
  19 STO d
  20 END
 53 BYTES
```

1° PROGRAMME: ID55 & ID55X (Inversion du Drapeau 55 avec ou sans Xfonction)

Le drapeau 55 est celui qui contrôle l'existance de l'imprimante.

Effectivement, tout le monde a remarqué que si un programme tourne, quand l'imprimante est branchée, il met plus de temps à s'éxécuter que si elle est inexistante.

Ce programme a pour but de "faire croire" à votre calculateur que celle-ci n'éxiste plus, alors qu'en fait, elle est branchée, en mais ses fonctions ne sont plus opérantes.

Il éxiste deux façons pour pouvoir retrouver la réutilisation de l'imprimante - soit de charger complètement le registre BUFFER, soit de réarmer le le drapeau 55.

```
      PRP "ID55X"

   01♦LBL "ID55X"
      02 RCL d
      03 CLA
      04 STO [
      05 - E
      06 AROT
      07 ATOX
      08 X<>F
      09 FC?C 00
      10 SF 00
      11 FS? 00
      12 SF 06
      13 FC? 00
      14 CF 06
      15 X<>F
      16 XTOA
      17 RCL [
      18 STO d
      19 END
END     45 BYTES
```

ID55(X), fait les deux operations puisqu'il change l'état de cet indicateur, il arme et désarme simultanément les indicateurs binaires 55 et 49. Comme nous avons vu précédemment, le 55 contôle l'existance de l'imprimante et le drapeau 49 contrôle l'allumage du voyant BAT qui est l'annonciateur de charge de labatterie. C'est le voyant BAT qui nous interresse et nous permettra de connaitre l'état du drapeau 55. Car sans avoir besoin de faire un test FS? 55, nous saurons si l'imprimante est utilisable ou non.

Dans le programme ID55, nous changeons l'état du drapeau 55 en étudiant l'état du drapeau Ø7 et en fonction de celà, nous agissons en conséquence sur le drapeau 49 par l'intermédiaire du drapeau Ø1 pour le mettre dans le même état que le 55.
Dans le programme ID55X, la fonction X()F, nous permet de remplacer respectivement les drapeaux 55 et 49 par les drapeaux ØØ et Ø6. -E peut être remplacé par -1.

```
 01*LBL "COD"
"|ABCDEFG"  ,006  STO L

 05*LBL 00
 E  RCL J  X<> d  X<>Y
CF 00  CF 01  CF 02
FS?C 03  GTO 01  SF 04
FC?C 07  SF 07  FS? 07
CHS  FS? 06  CHS  SF 06
X<0?  CF 06  X<0?  SF 05

 27*LBL 01
ST* X  FS?C 04  SF 00
FS?C 05  SF 01  FS?C 06
SF 02  FS?C 07  SF 03
FS?C 11  GTO 01  SF 12
FC?C 15  SF 15  FS? 15
CHS  FS? 14  CHS  SF 14
X<0?  CF 14  X<0?  SF 13

 51*LBL 01
FS? 12  SF 04  FS? 13
SF 05  FS? 14  SF 06
FS? 15  SF 07  RDN
X<> d  RCL J  RCL \
RCL [  STO \  R↑  STO [
"|*"  X<> \  R↑  STO J
R↑  STO \  "|**"  RCL Z
STO [  ISG L  GTO 00
RTN

 80*LBL "DECO"
FIX 9  CLA  STO [  X<> L
SF 00

 86*LBL 02
"|***"  RDN  RCL [
X<> d  CF 00  CF 01
CF 02  CF 03  X<> d
"|*"  RCL \  "****"
ARCL Y  ASHF  ASTO Y
CLA  STO [  FS?C 00
GTO 02  CLA  ARCL X
ASHF  ASHF  ARCL Y
ARCL Z  X<> T  RCL L
AVIEW  END
LBL'COD
LBL'DECO
END          234 BYTES
```

USER KEYS:

```
 12 XROM 05,60    ( STO b )
-12 XROM 01,60    ( RCL b )
 13 "COD"         ( 'CODE )
-13 XROM 01,61    ( RCL c )
 14 "DECO"        ( 'DECODE )
 33 "STOR"        ( 'STO REGISTRE )
-33 XROM 05,53    ( STO M )
 34 "RCLR"        ( 'RCL REGISTRE )
-34 XROM 01,53    ( RCL M )
```

2°PROGRAMME: COD (codage d'un nombre hexadécimal donné en alpha)
L'utilisation en est trés simple, il suffit de placer un nombre hexadécimal en alpha et de faire partir la routine COD, le résultat sera le nombre non normalisé correspondant et donné en X.

```
41424344454647
                XEQ "COD"
 -1,424344454-53    ***
                    CLA
                  STO [
ABCDEFG
```

```
4A4B3A3B3C3=3?
                XEQ "COD"
 -0,100000000    ***
                    CLA
                  STO [
JK:;<=?
```

3°PROGRAMME: DECO (décodage d'un nombre non normalisé donné en X et résultat en ALPHA)
L'utilisation en est tout aussi simple que le programme précédant vous entrez votre nombre en X puis vous éxécutez la routine DECO.
remarque:il n'a pas été fait de traduction des caractères donnés par le calculateur.
effectivement, vous remarquerez que les caractères hexa. donnés sont: Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9, :, ;, (, =,), ?.
au lieu de : Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
Les premiers sont ceux réellement reconnus par la machine.

```
ABCDEFG
                RCL [
 -1,424344454-53    ***
              XEQ "DECO"
41424344454647
```

```
HIJKLMN
                RCL [
   -0,849505152    ***
              XEQ "DECO"
48494:4;4<(4=4)
```

```
/W012=?
                RCL [
   -0,657303132    ***
              XEQ "DECO"
56573031323=3?
```

4°PROGRAMME: RCLR (rappel d'un registre dont on donne la valeur absolue)
Ces registres que vous pouvez rappeler peuvent se situer n'importe ou danla mémoire (registres d'état, de donnée, de mémoire programme, ...).
attention: Ce programme ne doit pas être éxécuté pas à pas, ou ne doit pas être arrété en cours d'éxécution. Si vous ne possedez pas le module XFONCTION, les nombres rappelés et les registres concernés seront normalisés, sinon aucun soucis à vous faire. Ne pas brancher l'imprimante. Charger le COD.

5°PROGRAMME: STOR (chargement d'un nombre dans unregistre quelconque)
Même remarques que dans le programme précédant. Charger le programme DECO.

```
                PRP "RCLR"

 01♦LBL "RCLR"
CF 04  SF 25  EMDIR
FC? 25  GTO 00  SF 00
CLD  "@@"  CRFLD

 11♦LBL 00
"REG?"  AON  STOP  AOFF
GTO 00

 17♦LBL "STOR"
SF 04

 19♦LBL 00
ASTO 02  "⊢1♦♦0"  RCL [
"⊢♦♦♦♦"  STO [  "⊢♦×i"
XEQ "COD"  X<> †  X<> \
"⊢**"  ,  STO ]  "⊢**"
RCL ]  "⊢*"  STO ]
"⊢****"  RCL ]  STO [
"⊢♦"  RCL [  STO 00
STO 01  CLD  FC? 04
GTO 00  "CODE?"  AON
STOP  AOFF  XEQ "COD"

 51♦LBL 00
RCL 01  X<> c  RCL [
FS? 04  GTO 00  FC? 00
GTO 00  "@@"  SAVER  ,
SEEKPT  GETX  X<>Y  R↑
R↑  GTO 01

 68♦LBL 00
X<> 00

 70♦LBL 01
X<>Y  STO c  R↑  FS? 04
RCL Z  "REG-"  ARCL 02
AVIEW  FS? 04  STOP
FC?C 00  GTO 00  "@@"
PURFL  STOP

 86♦LBL 00
R↑  R↑  END
```

CODES DEC. :

```
LIGNE09, 59 & 83= 242 64
                  64
LIGNE21= 245 127 49 0 0
         16
LIGNE23= 245 127 0 0 0 0
LIGNE25= 244 127 0 1 105
LIGNE39= 242 127 0
```

AFFICHAGE	IMPRESSION
M	[
N	\
O	]
P	†

```
                PRP "RCR"

 01♦LBL "RCR"
CF 00  EMDIR  "@"  CRFLD
GTO 00

 07♦LBL "STR"
SF 00

 09♦LBL 00
"REG?"  AON  PROMPT
AOFF  ASTO 01  "⊢1"
XEQ "HN"  "⊢♦"  RCL [
FC?C 00  GTO 01  "CODE?"
AON  PROMPT  AOFF
XEQ "HN"  RDN  X<> c
X<> [  STO 00  X<> [
GTO 02

 32♦LBL 01
X<> c  "@"  SF 25  SAVER
.  SEEKPT  CLX  GETX
PURFL  X<>Y

 43♦LBL 02
STO c  X<>Y  "REG-"
ARCL 01  AVIEW  TONE †
END
```

```
                CAT 1
END          03 BYTES
LBL'COD
LBL'DECO
END          234 BYTES
LBL'NH
LBL'HN
END          227 BYTES
LBL'RCR
LBL'STR
END          111 BYTES
LBL'RCLR
LBL'STOR
.END.        202 BYTES
```

```
               SIZE 010
01♦LBL "?"
02 ENTER↑
03 ENTER↑
04 ENTER↑
05 "ABCDEFG"
06 ENTER↑
07 ENTER↑
08 ENTER↑
09 ENTER↑
10 END
```

```
              GTO ."?"
                 RCL b
       0.0000-13    ***
                    DCO
187
```

```
              XEQ "RCLR"
DIR EMPTY
187                  RUN
REG-187
                     DCO
84017474C21709
              XEQ "RCLR"
DIR EMPTY
186                  RUN
REG-186
                     DCO
C600F2003F8383
 87 RCL [
                  GTO .005
                     RCL b
                      DCO
6185
              XEQ "RCLR"
DIR EMPTY
185                  RUN
REG-185
                     DCO
83F74142434445
     A B C D E XEQ "RCLR"
DIR EMPTY
184                  RUN
REG-184
                     DCO
4647838383833CC
```

```
 10 ENTER↑
 11 ENTER↑
 12 ENTER↑
 13 ENTER↑
 14 ENTER↑
 15 ENTER↑
 16 ENTER↑
 17 ENTER↑
 18 ENTER↑

               GTO .018
                  RCL b
                   DCO
183
F6314041505455       XEQ "STOR"
  (I)(E)(M)(P)(T)(U)
                     RUN
 ),7)7,1:9,00:+01
        ),7)7,1:9,00:+01
REG-183
               GTO ."?"
               GTO .018
                PRP "?"
```

```
 01♦LBL "?"
 02 ENTER↑
 03 ENTER↑
 04 ENTER↑
05 "ABCDEFG"
 06 ENTER↑
 07 ENTER↑
 08 ENTER↑
 09 ENTER↑
 10 ENTER↑
 11 "10APTU"
 12 ENTER↑
 13 END
```

```
0C0
              XEQ "STOR"
F09F75016             CLA
F09F750196F711        RUN
        ),7)7,1:9,00:+01
 ),7)7,1:9,00:+01
REG-0C0
                   PRKEYS

USER KEYS:
 11 XROM 61,53
-11 LST
 12 XROM 27,55
-12 XROM 01,60
 13 "COD"
-13 XROM 01,61
 14 "DECO"
-32 KY
 33 "STOR"
-33 LST
 34 "RCLR"
-34 LST
 82 LST
                   GTO ..

PACKING
 01 TONE [
 02 ISG IND ]
```

Il est difficile de donner en trés peu de mots une explication globale du fonctionnement et de l'utilisation complète de ces deux programmes. J'espère que les quelques exemples que j'ai donné vont suffire pour que vous puissiez comprendre le mécanisme de RCR(L) ST(O)R. Les programmes RCR et STR , doivent être utilisé avec le programme HN de la PPCROM et donné dans un précédent JPC, et sont ce des versions adaptées par Daniel Jacob. Tandis que RCLR et STOR doivent être utilisé avec le programme COD de la page précédente.
Quand vous faîtes un RCL b, les troi s derniers octets donnent la position du pointeur soit le registre qui le contient et le quatrième avant la fin, donne la position du pointeur, mais sur quel bit il se trouve et nepeut donc prendre qu'une valeur comprise entre Ø et 6.

Ces programmes sont optimisables, mais maintenant je vous laisse le soin de le faire

Je vous souhaite à tous, une Heureuse Programmation

REGISTREMENT VOTRE

PHILIPPE

CHOC EN RETOUR

Le Mans le 10 / 02/ 84

Cher monsieur,

Je ne suis pas pour l'instant membre de votre chapitre de Paris, cependant je me suis procuré à la règle à cacul les numéros de décembre et de novembre de PPCPC. J'y ai relevé (N8P3) le programme GETK - SAVEK, incompatible avec la présence d'alarmes et je suppose du buffer. Je n'ai pas relevé dans le numéro 9 de réponse à votre appel au peuple pour résoudre les petits problèmes potentiellement désagréables etsusceptibles de se poser lors d'un GETK avec des alarmes en mémoire. C'est pourquoi je vous propose le fruit de mes rapides cogitations synthetiques. Je n'ai pas cherché à optimiser le programme, quant à sa trame de fond, c'est celle de Frédéric Poupon.
Peut être cela rendra-t-il service à l'un de vos membres.

```
PRP "SAVEK"

 01*LBL "SAVEK"
RCL [  "×i@**"  X<> [
X<> c  1  CHS  ENTER↑

 09*LBL 00
RDN   E  +  RCL IND X
SF 25  SQRT  FS? 25
GTO 04  X<> \  CLX  ATOX
16  -  ,  STO \  RDN
X=0?  GTO 00

 28*LBL 04
CLX  2  +  X<>Y  X<> c
X<>Y  CF 25  CRFLD  X<>Y
X<> c  X<>Y  RCL '
SAVEX  RDN  RCL e  SAVEX
RDN  3  -  E3  /

 50*LBL 01
RCL IND X  STO [
"├"FO""  RCL [  STO \
"├*******"  X<> \
STO IND Z  SAVEX  RDN
RDN  ISG X  GTO 01
GTO 03

 65*LBL "GETR"
FLSIZE  2  -  LASTX  *
CLKEYS  +  11

 74*LBL 05
SF 25  PASN  1  +
FC? 25  GTO 05  DSE X
GTO 05  CLX  SEEKPT
GETX  STO '  GETX  STO e
"×i@**"  RCL [  X<> c
,9  SF 25

 94*LBL 02
GETX  FC? 25  GTO 03
STO IND X  RDN  ISG X
GTO 02

102*LBL 03
RDN  STO c  CLST  CLA
END
LBL'SAVEK
LBL'GETR
END          201 BYTES
```

Heureuse Programmation

FRANCK WETTSTEIN
33, Rue De La Marriette
72000 LE MANS

CODES DEC. DES LIGNES 03 ET 009: 245 1 105 0 0
CODES DEC. DES LIGNES 3 ET 09: 245 1 105 28 0 0

- NE PAS UTILISER COMME PREMIER PROGRAMME EN RAM

- NON AGRESSIF POUR LES ALARMES OU LE BUFFER.

CASSETTES HP: LE CATALOGUE (SUITE)

Fichiers annulés:

Un fichier annulé est un fichier dont les octets 10 et 11 sont à zéro.Ces deux octets sont automatiquement mis à zéro par la 41 pendant un PURGE.On pourrait penser que les 32 octets devenus alors libres vont systématiquement être ré-utilisés pour un autre fichier à venir.
En réalité,la 41 fait en sorte que les fichiers-catalogue et les fichiers soient dans le même ordre.
Plus généralement,si le fichier-catalogue purgé est:

*le dernier dans le catalogue,l'emplacement sera réutilisé pour tout nouveau fichier;

*à l'intérieur du catalogue,l'emplacement sera réutilisé si et seulement si la longueur du fichier est compatible, c'est-à-dire que la longueur du fichier en REC est inférieure ou égale à la valeur des octets 18 et 19 dans le fichier-catalogue effacé(pour une réécriture,si les longueurs ne sont pas compatibles,le fichier-catalogue est effacé,puis replacé ailleurs dans le catalogue).

Utilisation:

Certains pourraient se demander à quoi sert tout ceci.Il est vrai que les routines contenues dans le module HPIL permettent de mener à bien toutes les opérations concernant le lecteur. Certes,la plupart du temps,celles-ci pourraient être "accélérées" ou simplifiées.Personnelement,ces informations me sont trés utiles et je m'en sers beaucoup plus souvent que les fonctions du module HPIL.Mais comme je l'ai déjà écrit,il faut posséder le module I/O,ou le HP-IL DEVEL,ou encore faire soi-même des routines en microcodes.
Néansmoins,tout le monde pourra comprendre certaines réactions de leur système,qui à première vue n'ont rien de rassurant pour l'utilisateur(je vous invite à réfléchir à ce petit problème:

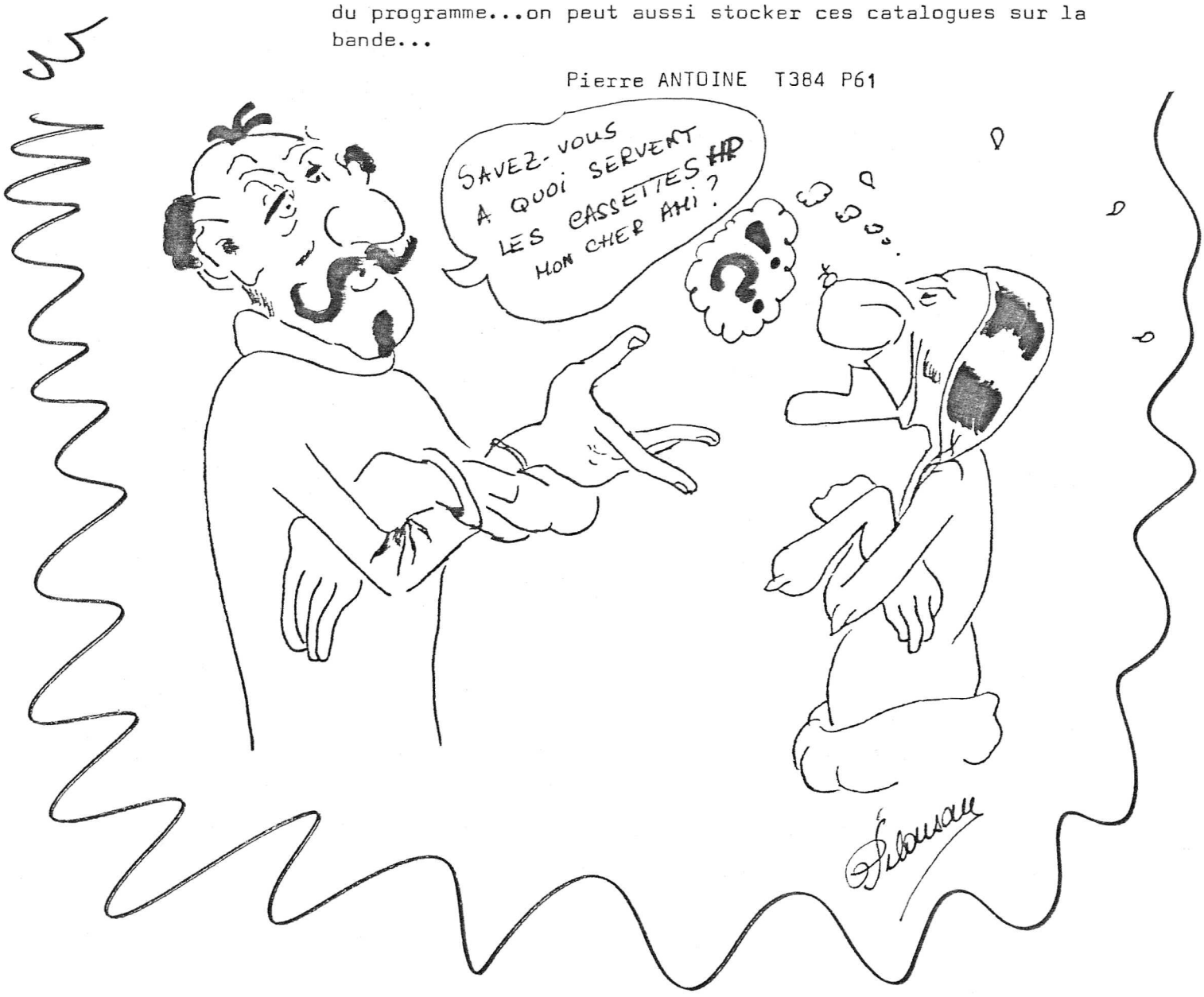
Une cassette à été formattée avec un NEWM 200.Après quelques temps d'utilisation,vous voulez créer un fichier. Mais voilà que la 41 vous affiche un "DIR FULL" alors qu'il n'y a que 2 fichiers sur toute la cassette! Incroyable,mais tout à fait possible.Le pire est qu'on ne peut rien y faire en général...)

*Création de fichiers inaccessibles par le module HPIL: ce sont les fichiers de type inconnus(octets 10 et 11 voir JPC Septembre 83).Ce type de fichier pourrait être destiné à envoyer des informations préalablement stockées sur la bande (vers VIDEO,imprimante,...),par exemple des tableaux,...
On peut utiliser les messages ordres dépendant récepteur et émetteur pour positionner directement la cassette et utiliser après le XFER du module I/O.

*Il est possible d'enregistrer le fichier après un programme avec exécution automatique(A) sans déclarer d'autre fichier. Pour cela il faut sauver d'abord le programme.Ensuite:
Sélectionner le Drive comme appareil principal (SELECT)
Lire l'octet de position de la bande (3 DEVT INAN module I/O)
Décrémenter d'une unité le numéro du REC (c'est le 3ème octet dans ALPHA)
Envoyer la nouvelle position (4 DEVL OUTAN).A noter!C'est la position du fichier.

Mettre le lecteur en position d'écriture partielle (6 DEVL) et envoyer les octets vers le Drive (OUTXB,ACCHR,OUTA,OUTP, OUTBUFX,...)
Fermer l'enregistrement (8 DEVL)
Reste à déclarer les octets.Pour cela il faut positionner la bande sur le REC qui contient le fichier-catalogue du programme et modifier les octets 18 et 19 (il faut ajouter la longueur en REC des octets rajoutés).Encore une fois,il faut utiliser les ordres dépendant récepteur et émetteur...
Ainsi,une fois le programme chargé,la bande est positionnée au bon endroit pour le transfert des octets (0 DEVT x XFERN, INXB,INBUFX,INA,IND,...)
*On peut modifier le catalogue comme on le veut.Mais attention il ne faut pas oublier que tous les octets sont importants et que ceux-ci doivent être correctement spécifier sans en oubli oublier.
*On peut également utiliser la Mémoire Tampon 1 pour mettre des pseudo-catalogue.Par exemple si on met en MT1 le fichier catalogue du programme "PROGRAM"(voir JPC Septembre 83) et que l'on fait READP,1a 41 va directement positionner la bande au REC 1/82 et commencer à transférer les 327 octets du programme...on peut aussi stocker ces catalogues sur la bande...

Pierre ANTOINE T384 P61



Frédéric TOUPON. # 35.

JOUONS A PUISSANCE QUATRE sur un damier 6x6.

JEUX

Puissance 4, vous connaissez, il s'agit de réaliser des alignements de quatre pions sur une grille. La principale difference avec le morpion reside dans le fait que vous ne pouvez, dans chaque colonne, jouer que dans la premiere case libre. Consequece pratique: le numero de la colonne suffit a definir le coup.

Je vous propose ce mois-ci d'y jouer contre votre adversaire préférée. Je présente tout de suite mes excuses aux plus fauchés d'entre nous, mais le module X-Fonctions est absolument indispensable (débrouillez vous pour l'acheter, il vaut vraiment le coup!). Apres avoir fait de la place dans votre mémoire etendue, chargez le programme (vous prendrez bien soin, au moment de l'enregistrement, de lever le flag 11), le départ automatique vous demandera les cartes de données. Puis, le message "PRET" vous indiquera que tout est O.K.

Pour demander a votre 41 d'ouvrir les hostilités, tapez la touche B. Pour lui indiquer quel est votre coup foireux, entrez le numéro de la colonne jouée et tapez A. Pour visualiser une colone, indiquez en le numéro a la chère petite, et tapez C. A l'affichage, un 0 indique une case vidé, un 1 une case que vous occupez, un 2, une case qu'elle s'est gentiment appropriée.

Les hostilités sont censées cesser des que la 41 (à moins que vous ne soyez une vraie bête) aligne 4 pions, elle se fera une joie de vous le faire remarquer. Pour le décompte des points, les règles suivantes sont communément admises: en cas de victoire de la 41, celle-ci marque 4 points, vous 0; en cas de victoire de votre part, vous marquez 3 points, la 41 en marque 1; en cas de match nul, la 41 marque 2 points, vous 1. Le premier qui fait 40 à 0 a mal tapé le programme et doit tout recommencer.

N'oubliez pas de compiler le pgm, il tournera plus vite!!!

Allez, Salut !

```
"†4"  15  PASN  01
"⊦-D"  CRFLD  PSIZE
RDTA  SAVER

 10↑LBL "†4"
"†4-D"  GETR  "ƒ↔#↔▵"
RCL [  STOFLAG  "PRET"
PROMPT

 18↑LBL A
STO [  15  +  RCL IND X
ISG IND Y  DEG  RCL [
RCL 12  *  +  24  +  E
STO IND Y

 33↑LBL B
5 E-3  STO 10  CLX
STO 03  -1 E4  STO 04



 40↑LBL 60
RCL 10  INT  RCL 12  *
RCL 10  16  +  RDN
RCL IND T  RCL 15  CF 02
X=Y?  SF 02  CLX  RCL 12
X=Y?  GTO 65  RDN  31  +
+  STO 00  CLX  STO 09
CF 01  XEQ 08  CLX
X<> 09   E4  X<=Y?
GTO 09  RDN  STO 06
FS? 02  ISG 00  FS? 02
GTO 03  SF 01  RCL 07
STO IND 00  ISG 00  DEG
XEQ 08  RCL 00  RCL 09
ST- 06  SIGN  -  ,
STO IND Y

 91↑LBL 03
SF 06  RCL 04  RCL 06
X<Y?  CF 06  X>Y?
STO 04  RCL 00  FS? 06
STO 03

102↑LBL 65
ISG 10  GTO 60
"JE JOUE "  DSE 03  DEG
RCL 07  STO IND 03
RCL 03  3  -  RCL 12  /
INT  3  -  ARCL X  AVIEW
BEEP  RCL 02  +
ISG IND X  DEG  STOP



126↑LBL 20
STO 01  1,004  STO 05
```

```
130↑LBL 10
RCL 00  RCL 01  RCL 05
INT  *  -  STO [  RCL 13
STO \  CLX  STO 08



142↑LBL 00
RCL 01  ST+ [  RCL 30
RCL IND [  X=Y?  GTO 05
ST+ 08  DSE \  GTO 00
RCL 08  RCL 07  MOD
X=0?  GTO 02  RCL 08
LASTX  /  INT  X≠0?
GTO 05  RCL 08  "×x̄α∫"
FS? 01  "×↔β↓"  AROT
ATOX  RCL 11  -  10↑X
ST+ 09  GTO 05

174↑LBL 02
RCL 08  "×↔β↓"  FS? 01
CLA  AROT  ATOX  RCL 11
-  10↑X  ST+ 09

185↑LBL 05
ISG 05  GTO 10  RTN



189↑LBL 09
CLA  RCL 10   E  +
ARCL X  "⊦ JE GAGNE"
AVIEW  BEEP  BEEP  BEEP
STOP

201↑LBL 08
RCL 12  XEQ 20   E
XEQ 20  RCL 14  XEQ 20
RCL 15  GTO 20

210↑LBL C
" "  ARCL X  "⊦  "
RCL 12  *  25  +  RCL X
RCL 07  +   E3  /  +



224↑LBL 04
ARCL IND X  AVIEW  ISG X
GTO 04  STOP  .END.
```

```
R00=  0
R01=  0
R02=  15
R03=  0
R04=  0
R05=  0
R06=  0
R07=  5
R08=  0
R09=  0
R10=  0
R11=  3
R12=  7
R13=  4
R14=  8
R15=  6
R16=  1
R17=  1
R18=  1
R19=  1
R20=  1
R21=  1
R22=  1
R23=  1
R24=  9
R25=  9
R26=  9
R27=  9
R28=  9
R29=  9
R30=  9
R31=  9
R32=  0
R33=  0
R34=  0
R35=  0
R36=  0
R37=  0
R38=  9
R39=  0
R40=  0
```

```
R41=  0
R42=  0
R43=  0
R44=  0
R45=  9
R46=  0
R47=  0
R48=  0
R49=  0
R50=  0
R51=  0
R52=  9
R53=  0
R54=  0
R55=  0
R56=  0
R57=  0
R58=  0
R59=  9
R60=  0
R61=  0
R62=  0
R63=  0
R64=  0
R65=  0
R66=  9
R67=  0
R68=  0
R69=  0
R70=  0
R71=  0
R72=  0
R73=  9
R74=  9
R75=  9
R76=  9
R77=  9
R78=  9
R79=  9
R80=  9
```

```
L1: 242,94,52,
L5: 243,127,45,68,
L11: 244,94,52,45,68,
L13: 247,31,240,0,0,35,0
,8,
L16: 244,80,82,69,84,
L105: 248,74,69,32,74,79
,85,69,32,
L164: 244,1,2,4,6,
L166: 244,1,3,5,7,
L176: 244,1,3,5,7,
L195: 250,127,32,74,69,3
2,71,65,71,78,69,
L211: 241,32,
L213: 243,127,32,32,
```

Programme "XEDIT"

```
 02*LBL "XEDIT"
SIZE? 18 X>Y? PSIZE
 E9 STO 16 STO 17 598
ENDIR - "NOM ?" AON
TONE 7 PROMPT AOFF
RCL [ RCLPTA STO 13
FLSIZE "≠" CLFL SAVEX
R↑ R↑ X<>Y ENTER↑
"×iλ    ****" 256 /
XTOA X<> L MOD XTOA
R↑ RCL \ X<> c RCL [
STO 00 X<>Y STO c
RCL Z CLA STO [ 7
ALENG - X=0? GTO 00


 51*LBL 05
"├ " DSE X GTO 05



 55*LBL 00
2 SF 25

 58*LBL 06
RCLPT + SEEKPT GETX
RCL [ X=Y? GTO 01
GETX "ABCDEFGH" STO [
ASHF RDN ALENG ATOX
ATOX R↑ STO [ RDN
RCL Z DSE X ABS X≠0?
GTO 00 RDN X<>Y R↑



 85*LBL 00
RDN X<>Y 16 MOD
LASTX X↑2 * + GTO 06


 95*LBL 01
GETX RCLPT STO 02 ,
SEEKPT X<>Y GETX +
DSE X STO 03 RCLFLAG
STO 11 CF 05 RCL 02
STO 00 SEEKPT 4,01
STO 01 STO 15 XEQ 14


116*LBL 61
FIX 0 CF 29 CLA
RCL 00 RCL 02 -
FS?C 05 GTO 00 E2
X>Y? "0" CLX E1
X>Y? "├0" ARCL Y "├-"
RCL 01 RCL 15 -
GTO 65

138*LBL 00
7 * RCL 01 RCL 15 -
+ E3 X>Y? "0" CLX
 E2 X>Y? "├0" CLX
 E1 X>Y? "├0" X<>Y
```

```
157*LBL 65
ARCL X "├: "
RCL IND 01 E2 X>Y?
"├0" CLX E1 X>Y?
"├0" X<>Y ARCL X "├-"
XTOA

172*LBL 62
CF 22 FIX 9 SF 25 ABS
FC?C 25 CLX X<=0? ,1
TONE 3 PROMPT FS? 48
GTO 63 X<0? GTO 00
FS? 22 STO IND 01
ISG 01 GTO 61 XEQ 12
RCL 00 RCL 03 X<=Y?
GTO 01 XEQ 09 GTO 61


198*LBL 01
SIGN ST- 01 GTO 99



202*LBL 00
SIGN ST+ 01 RCL 15
RCL 01 E + X≠Y?
GTO 61 XEQ 12 DSE 00
RCL 02 RCL 00 X<Y?
GTO 00 SEEKPT XEQ 14
10,01 STO 01 GTO 61


222*LBL 00
SIGN ST+ 01 ST+ 00
GTO 98

227*LBL 12
CLA RCL 04 XTOA
RCL 05 XTOA RCL 06
XTOA RCL 07 XTOA
RCL 08 XTOA RCL 09
XTOA RCL 10 XTOA
RCLPT DSE X SEEKPT
RCL [ SAVEX RTN


249*LBL 14
CLA GETX STO [ 4,1 7


255*LBL 13
ALENG X=Y? GTO 00 CLX
GTO 01

261*LBL 00
RDN ATOX

264*LBL 01
STO IND Z ISG Z RDN
DSE X GTO 13 RTN
```

Au début, il y eut la compilation manuelle des GTO. Puis, il y eut le fantastique programme "COMP" de Frédéric Poupon. Mais ceci engendra de nouveaux problèmes: comment modifier des programmes déjà compilés ?

Le vénérable CRIC, ainsi que quelques autres outils, furent pendant longtemps mis à rude épreuve. Mais maintenant, vous allez pouvoir utiliser XEDIT : éditeur de programmes en mémoire étendue.(entre parenthèses, c'est ce genre d'outil que j'aurais aimé voir figurer dans les modules HP).

Le but est donc de modifier dans la mémoire étendue, des programmes compilés, ou simplement "semi-compilés" (sans enlever les Labels), et le tout dans l'esprit 41C.

La conception est modulaire, et non optimisée. Vous pourrez donc, si vous en sentez le besoin, ajouter vos propres modules, ou enlever ceux dont vous ne vous servirez pas. N'oubliez pas de faire partager aux adhérents de PPC-PC vos découvertes.

Le programme fait 1587 octets dans cette version (c'est la 3°, comme l'indique le 2 de la ligne 1). L'utilisation nécessite une configuration mémoire de 018 registres de données au minimum. Il a été conçu pour une configuration comportant 1 module XFonctions, et 2 modules XMémory. Si vous ne disposez pas de tous ces modules, vous devrez changer la valeur de quelques constantes.

Par convention, et pour simplifier la partie "recherche du fichier", vous devrez placer un fichier ayant pour nom: "≠", de type données, et de longueur 2 registres. Ce fichier doit obligatoirement se situer au début de la mémoire étendue

Pour illustrer le mode d'emploi, nous utiliserons XEDIT sur lui même en mémoire étendue.

Chaque module a son utilisation propre, mais ils sont tous appelés de la même manière, à partir du module de base, ç'est à dire le programme principal.

Module de base:

- Charger le programme en mémoire étendue, après vous être assuré que le fichier "≠" était en place. "XEDIT", SAVEP

XEQ"XEDIT"

Entrez le nom du fichier programme

R/S

Voilà, vous êtes dans le programme. Toute commande vous ramènera à l'étape suivante:

- Vous voyez à l'affichage "RRR-O: CCC-S"

RRR est la valeur du pointeur en registres

O est la place de l'octet pointé dans le registre

CCC est le code décimal de l'octet pointé

S est son symbole ASCII (exemple: 065-A)

- Vous voulez
  - + passer à l'octet suivant: faîtes R/S
  - + revenir à l'octet précédent: faîtes CHS, R/S
  - + Changer la valeur de l'octet: entrez la nouvelle valeur, R/S
  - + éxécuter une commande: passez en mode Alpha, entrez le nom, restez en mode Alpha, R/S.

Simple, non ?

Voyons maintenant les messages d'erreur du module de base:

"FL NOT FOUND": le fichier n'existe pas en mémoire étendue.

"LIM. INF.": la limite inférieure du fichier programme est atteinte.

"LIM. SUP.": la limite supérieure du fichier programme est atteinte.

"PARDON ?": la commande demandée n'existe pas.

Le module de base comporte aussi les instructions suivantes:

"FIN" : Pour terminer de travailler sur le fichier, et remettre la mémoire étendue en place.

Attention: cette commande n'est pas valable si on a modifié le programme: toute tentative de rappel par GETP ou GETSUB se soldera par un "CHKSUM ERR". Par contre, elle est valable si on envisage d'arrêter de travailler temporairement sur ce fichier, sans vouloir le rappeler. Pour pouvoir rappeler ce fichier, il faut utiliser la commande "CHK":

```
271*LBL 99
XEQ 00  "⊢SUP."  GTO 62


275*LBL 98
XEQ 00  "⊢INF."  GTO 62


279*LBL 00
"LIMITE "  TONE 0  RTN


283*LBL 97
"PAS DE GTO"  TONE 0
GTO 62

287*LBL 96
"TROP LONG"  TONE 0
GTO 62

291*LBL 63
AOFF  ASTO 12  XEQ 12
SF 25  GTO IND 12
"PARDON ?"  TONE 3
GTO 62

300*LBL "PT"
CF 25  "REG. ↑ OCT."
CLST  TONE 7  PROMPT
X<0?  GTO 61

308*LBL 64
RCL X  7  /  INT  RCL Z
+  X<0?  GTO 98  RCL 02
+  RCL 03  X<Y?  GTO 99
RDN  STO 00  RDN  7  MOD
RCL 15  +  STO 01
RCL 00  SEEKPT  XEQ 14
GTO 61

334*LBL "PTO"
CF 25  SF 05  GTO 61


338*LBL "PTR"
CF 05  CF 25  GTO 61


342*LBL "POS"
CF 25  "XTOA"  41  PASN
CLST  "CARACTERES ?"
AVIEW  CLA  TONE 7  STOP
AOFF  ASTO X  CLA  41
PASN  RCL 03  RCL 00
SEEKPT  -  ENTER↑  SIGN
+  STO a  GETX  STO [
STO \  R↑  SIGN

371*LBL 07
DSE a  FS? 48  GTO 00
GETX  X<> [  STO \
LASTX  POSA  X<0?
GTO 07  RCL 03  RCL 02
-  RCL a  -  X<>Y
GTO 64
```

```
389*LBL 00
RCL 00  ENTER↑  SIGN  +
SEEKPT  TONE 0
"PAS TROUVE"  GTO 62


398*LBL "LSY"
SF 06  RCL 16  ENTER↑
SIGN  +  STO 12  GTO 27


406*LBL "NL?"
CF 06  RCL 17  RCL 00  7
*  RCL 01  RCL 15  -  +
STO 12  GTO 00


418*LBL "NUM"
 E  CHS  "LIGNE ?"
PROMPT  X<0?  GTO 61
SF 06  STO 12  RCL 16
X<>Y

429*LBL 00
X=Y?  GTO 01  X>Y?
GTO 00  SIGN  STO 16
RCL 02  7  *  STO 17


440*LBL 27

441*LBL 00
XEQ 20

443*LBL 28
RCL 15  STO 01  RCL 17
7  MOD  ST+ 01  RCL 17
LASTX  /  INT  STO 00


455*LBL 21
RCL 00  SEEKPT  XEQ 14


459*LBL 01
"L "  FIX 0  CF 29
RCL 16  E2  X>Y?  "⊢0"
CLX  E1  X>Y?  "⊢0"
X<>Y  GTO 65

473*LBL 20
RCL 17  STO Y  7  MOD
STO 04  -  ENTER↑  SIGN
ST- 16  -  STO 06
RCL 03  7  *  STO 05
RCL 17  LASTX  /  SEEKPT
GETX  STO [  LASTX
STO \  RCL 04  X=0?
GTO 23
```

"CHK" : restaure le CHKSUM, et branche ensuite sur la partie "FIN". Tout est remis en place, vous n'avez plus qu'à rappeler le programme.

Module de pointeur:
Il comprend 2 instructions, qui ne font qu"agrémenter l'affichage.
"PTO" : Tout à l'heure, vous avez vu la repréeentation du pointeur par un numéro de registre, et un numéro d'octet. Il est possible de changer pour avoir l'adresse en octets (nombre à 4 chiffres: vous verrez: "0000: CCC-S")
"PTR" : remet l'affichage en mode "Pointeur registre"
Le drapeau 05 est utilisé pour repérer le mode choisi. Il n'y a aucun message d'erreur.

Module de modification du pointeur:
"PT" : Pour positionner le pointeur à un endroit précis du programme.
Après le R/S, vous voyez à l'affichage: "REG./OCT."
Rentrez le numéro de registre, ENTER↑, puis le numéro de l'octet. Vous pouvez aussi ne rentrer que l'adresse du pointeur en octets (PTO).
La pile est nettoyée avant l'arrêt: faîtes simplement R/S si vous désirez revenir au début du fichier programme.
- Si vous vous êtes trompé, et ne désirez pas changer la valeur du pointeur, entrez un nombre négatif et faîtes R/S.
Il n'y a aucun message d'erreur propre à ce module, mais "LIM. INF." et "LIM. SUP." s'appliquent à ce cas.
Attention: n'enlevez pas ce module: il contient le LBL 64 qui est utile à d'autres modules.

Module de recherche dans le programme:
"POS" : (analogie avec POSFL)
- permet de rechercher la position d'un octet ou d'un groupe d'octets (jusqu'à 6).
- Après R/S, vous voyez "CARACTERES ?"
Rentrez ces caractères dans le registre Alpha
La pile est utilisable, XTOA est assigné à la touche ENTER↑. Terminez par R/S, que vous soyez en mode Alpha ou non.
- Après éxécution, le pointeur se trouve transféré au premier octet du groupe.
- Si les codes ne sont pas dans le fichier, vous aurez droit au message d'erreur:
 "PAS TROUVE".
Attention: la recherche commence à partir de la valeur actuelle du pointeur (analogie avec POSFL), donc si votre groupe de codes se trouve avant l'octet pointé, XEDIT ne le trouvera pas.

Module de compilation:
Si vous voulez compiler des GTO manuellement, changer la compilation de l'un d'entre eux, vous brancher directement sur une instruction ou tout ce qui peut passer par votre tête de synthétiseur fou.
Avant de compiler un GTO il faut repérer l'adresse d'arrivée.
"LBL": enregistre cette position,
"GTO": compile le GTO
La première commande ne change rien à la valeur du pointeur, ou des codes qui s'y trouvent.
Les messages d'erreur de ce module sont:
"TROP LONG": si le saut est supérieur à 112 octets.
"PAS DE GTO": Pour pouvoir compiler un GTO, encore faut il qu'il y en ait un !

Module de numérotation des lignes de programme:
Vous avez peut être constaté qu'il était assez délicat de se repérer dans le fichier à l'aide des seules adresses: il est plus facile de se repérer avec une unité de mesure qui nous est familière: la ligne de programme.
Pendant l'éxécution d'une de ces trois routines, le numéro de la ligne en cours d'examen est affiché.
"NL?": remplace à l'affichage l'adresse du pointeur par le numéro de la ligne.

```
500*LBL 22
XEQ 11  DSE 04  GTO 22


504*LBL 23
XEQ 11

506*LBL 24
X<0?  GTO 23  SIGN
ST+ 16  VIEW 16  RCL 06
STO 17  FC? 06  GTO 00
RCL 16  RCL 12  X=Y?
RTN

520*LBL 00
15  LASTX  X<=Y?  GTO 23
29  X<=Y?  GTO 00



528*LBL 25
XEQ 11  15  X<>Y  X<=Y?
GTO 24  28  X<>Y  X>Y?
GTO 24  GTO 25

539*LBL 00
CLX  32  X<=Y?  GTO 00
XEQ 11  240  X<Y?
GTO 29  X<>Y  GTO 24



550*LBL 00
CLX  144  X>Y?  GTO 23
CLX  207  X=Y?  GTO 01
DSE X  X=Y?  GTO 01  CLX
192  X<=Y?  GTO 00



566*LBL 01
XEQ 11  GTO 23

569*LBL 00
CLX  240  X<=Y?  GTO 29
CLX  206  X<=Y?  GTO 00
XEQ 11  XEQ 11  240
X>Y?  GTO 23

583*LBL 29
X=Y?  GTO 23  -  STO 10


588*LBL 10
XEQ 11  DSE 10  GTO 10
GTO 23

593*LBL 00
XEQ 11  XEQ 11  GTO 23
```

```
597*LBL 11
CLX  X<> \  ALENG  X<>Y
-  X=0?  ATOX  ISG 06
""  RCL 06  RCL 05  X=Y?
GTO 28  FS? 06  GTO 00
CLX  RCL 12  X<Y?
GTO 21

617*LBL 00
DSE L  GTO 00  GETX
STO [  R↑  7  STO \
X<>Y  RTN

627*LBL 00
LASTX  STO \  R↑  RTN



632*LBL "GTO"
CF 25  RCL 00  7  *
RCL 01  +  STO 12
RCL IND 01  16  /  INT
14  X=Y?  GTO 59  DSE X
X=Y?  GTO 59  2  -  X≠Y?
GTO 97  112  RCL 14
RCL 12  LASTX  +  -  ABS
X>Y?  GTO 96  LASTX
SIGN  X>0?  CLST  LASTX
ABS  X<>Y  128  *  CHS
RCL Y  RCL Z  7  /
ST+ Z  RDN  LASTX  MOD
16  *  +  STO 12  RCL 01
10,01  X≠Y?  GTO 00
XEQ 14  RCL 12  INT
STO 04  XEQ 12  RCL 00
RCL 01  GTO 64

697*LBL 00
RCL 01  ISG X  R↑  INT
STO IND Y  GTO 61



704*LBL 59
16  *  RCL 14  RCL 12  -
 E  -  STO 12  ABS  7
MOD  ST+ X  +  RCL 12  7
/  INT  STO 12  ABS  256
/  INT  +  STO IND 01
RCL 01  10,01  X=Y?
XEQ 09  ISG 01  RCL 12
ABS  256  MOD
STO IND 01  RCL 01
10,01  X=Y?  XEQ 09
ISG 01  RCL IND 01  128
MOD  LASTX  RCL 12  SIGN
X>0?  CLX  CHS  *  +
STO IND 01  XEQ 12  CLST
RCL 00  RCL 02  -  7  *
RCL 01  RCL 15  -  +  2
-  GTO 64
```

"NUM": (équivalent à GTO.---)
vous voyez à l'affichage: "LIGNE ?":
- + entrez le numéro de la ligne où vous voutransférer le pointeur.
- + si vous vous êtes trompés, si vous ne voulez pas déplacer le pointeur, faîtes simplement R/S (ou un nombre négatif, R/S)

"LSV" (Ligne SuiVante):
déplace le pointeur à la ligne suivante (équivalent à SST)
Pour des raisons de temps, la recherche commence toujours à la dernière position du pointeur de ligne, si c'est possible, sinon à la ligne Ø.
Le pointeur "lignes de programme" est distinct du pointeur normal d'octets: on peut très bien être à l'octet ØØØ2, par exemple, et l'éxécution de LSV peut nous déplacer à la ligne 876.
A titre indicatif, il faut à une 41 accélérérée à 1,9 fois sa vitesse normale 20 minutes pour parvenir à la fin de XEDIT par NUM.
Pour atteindre la dernière ligne d'un programme, il suffit d'entrer un nombre très grand à la question "LIGNE ?", dans NUM.

Examinons maintenant le programme lui même:
Le principe est simple: réunir toute la mémoire étendue en un seul fichier, de type données, de nom "≠". La taille de ce fichier est la taille totale de la mémoire étendue. Ce fichier contient donc le fichier programme demandé. On le traite comme une suite de registres de données. Ceci est possible car GETX et SAVEX ne normalisent pas les nombres,
Pour faciliter les manipulations, et pour accélérer le traitement, le contenu du registre en cours d'étude est transféré octet par octet dans les mémoires Ø4 à 1Ø. Ce gracupe de mémoires est appellé "image du registre" en mémoire centrale.
Dès qu'une commande est demandée, on sauvegarde le contenu de ces mémoires avant son éxécution.
Enfin, la structure permet d'jouter des modules sans avoir besoin de changer quoi que ce soit dans le programme principal.
Pour créer sa propre commande, il suffit de la placer dans un endroit innoccupé, et de commencer par un Label Alphanumérique comportant au maximum 6 caractères .
Ce module doit respecter les mémoires et les données utilisées par les autres commandes.
Le nom figure dans le registre 12: celui-ci peut donc servir de registre brouillon (variable locale).
Attention: le drapeau 25 est armé à l'entrée du module. Le pointeur (SEEKPT,RCLPT) est positionné sur le registre suivant.

Examinons maintenant les principaux points d'entrée du programme principal.
LBL 61: pour revenir à l'affichage "RRR-O: CCC-S" (ou "0000: CCC-S" suivant le drapeau 05)
LBL 62: Revenir simplement au point d'arrêt principal, et afficher le registre Alpha.
LBL 63: Pour éxécuter une commande: il faut le nom en Alpha.
LBL 64: Equivalent à la commande "PT", en X: nombre d'octets, en Y: nombre de registres.
LBL 65: Pour ajouter au contenu du registre Alpha "X: CCC-S" Xest le nombre contenu dans X, et le FIX Ø, CF29 est exigé.
La zone des Labels 9Ø est réservée aux messages d'erreur.
Quelques sous-programmes peuvent vous être utiles:
LBL 12: recopie les mémoires Ø4 à 1Ø dans le registre précédent celui pointé.
Ici, il n'y a pas de boucle, pour économiser du temps.
LBL 14: opération inverse de la précédente: extraction des codes du registre pointé, et stockage dans les mémoires Ø4 à 1Ø.
LBL Ø8: ajoute X octets nuls au registre Alpha.

```
770*LBL 09
XEQ 12  XEQ 14  RCL 15
STO 01  SIGN  ST+ 00
RTN

778*LBL "LBL"
CF 25  RCL 00  7  *
RCL 01  +  STO 14
GTO 61

787*LBL 08
"├*"  DSE X  GTO 08  RTN


792*LBL "CHK"
CF 25  RCL 03  RCL 02
SEEKPT  -  ,

799*LBL 02
GETX  STO [  7  R↑  R↑


805*LBL 03
ATOX  +  DSE Z  GTO 03
DSE Y  GTO 02  GETX
STO [  RCL 13  7  MOD
STO 13  X≠0?  XEQ 08
STO [  X<> L  RCL \  R↑


824*LBL 04
ATOX  +  DSE Z  GTO 04
X<>Y  STO [  X<>Y  256
MOD  XTOA  7  RCL 13  -
ENTER↑  SIGN  -  X≠0?
XEQ 08  RCLPT  DSE X
SEEKPT  RCL [  SAVEX



848*LBL "FIN"
CF 25  "=i\    *****x̄"
RCL \  X<> c  RCL [
STO 00  X<>Y  STO c
RCL 11  STOFLAG  CLST
CLA  TONE 5  STOP  .END.

                SAVEP
               FLSIZE
     228,0000000    ***
               RCLPT
   1.593,000000    ***
```

Attention: XEDIT ne supporte pas de travailler sur le dernier fichier d'une mémoire étendue saturée (c'est à dire quand un EMDIR retourne Ø dans X). Il est très susceptible...

Regardons maintenant l'occupation et le rôle des mémoires dans le module de base:
00: Pointeur de registres
01: Pointeur d'octet (à l'intérieur du registre)
02: Adresse du 1° registre du programme
03: Adresse de dernier registre du programme
04)
05 (
06)
07 (Image du registre
08)
09 (
10)
11: Drapeaux (RCLFLAG)
12: Nom de la commande demandée (la cas échéant)
13: Taille du programme en octets
14: Adresse du Label pour la compilation
15: Constante=4,01
16: n° de la dernière ligne de programme examinée
17: Adresse correspondante en octets
et dans le fichier "≠"
00: Taille du programme en registres
01: inutilisé

Ces mémoires sont inchangées dans la plupart des modules. Cependant, les n° de lignes utilisent certaines des mémoires ci-dessus comme variables à usage local:
04: nombre d'octets jusqu'à la ligne de début
05: Adresse de fin
06: Adresse de l'octet en cours
10: Contrôle de boucle pour les octets de type Fn
12: n° de ligne, ou adresse, suivant le drapeau Ø6
16: N° de ligne en cours
17: Adresse du début de la ligne en cours
M: registre en cours
N: Pointeur de l'octet dans ce registre

Pour rentrer ce programme, vous aurez peut-être besoin des codes décimaux des chaînes de caractères:

n° de ligne codes décimaux des chaînes de caract.

29: 1,1Ø5,11,224,192,32,Ø,Ø,Ø,Ø
52: 127,32
6Ø6: Chaine nulle (FØ)
788: 127,Ø
85Ø: 1,1Ø5,11,224,192,32,Ø,Ø,Ø,Ø,Ø,Ø,2

Si vous n'avez pas la chance de posséder 2 modules XMemory, vous devrez changer la valeur de la constante à la ligne 1Ø .
Si vous possédez un XFonctions seul , mettez 122.
Si vous avez en plus un module XMemry, mettez 362.
J'ai placé cette constante, plutôt qu'une question, car il n'est pas très agréable de se voir interrogé par une stupide bécane qui ne sait même pas reconnaître ce qu'elle porte dans ses ports d'extension.

Voilà, je pense vous avoir tout dit au sujet de ce programme. J'éspère qu'il vous sera aussi utile qu'il l'est pour moi.
N'oubliez pas: c'est un feuilleton, dont vous serez les réalisateurs pour les prochains épisodes...

Vu lá longueur du programme, vous apprécierez certainement la "progammathèque" que met en place André OISEL. Envoyez-lui vos cartes magnétiques (15 pistes), ou votre cassette, accompagnées d'une enveloppe timbrée pour le renvoi.
Voici son adresse:
5, Les Nouveaux Horizons
ELANCOURT 78310 MAUREPAS
N'hésitez pas à lui mettre un petit mot: c'est un grand service qu'il rend à tous les membres du club.

Pour tout renseignement sur un point précis du programme, écrivez-moi :
Pierre DAVID
33 Bld St Martin
75003 PARIS

A tous les programmeurs fous et martyriseurs de machines, je dis A Bientôt !

P.S. : Les TONEs du programme XEDIT ne sont pas des TONE légaux. Voici les codes à rentrer:

Ligne	Code décimal	Fonction correspondante
15	87	10^x
181	73	EMS+
281	100	x>0?
285	100	x>0?
289	100	x>0?
298	53	STO 05
304	87	10^x
351	87	10^x
861	45	RCL 13

Dans l'ensemble, ils sont plus courts que les "normaux", ce qui est plus agréable à l'usage.

TABLE des valeurs & positions des drapeaux dans le registre d

binaire	décimal							
0000 0001	1	F07	F15	F23	F31	F39	F47	F55
0000 0010	2	F06	F14	F22	F30	F38	F46	F54
0000 0100	4	F05	F13	F21	F29	F37	F45	F53
0000 1000	8	F04	F12	F20	F28	F36	F44	F52
0001 0000	16	F03	F11	F19	F27	F35	F43	F51
0010 0000	32	F02	F10	F18	F26	F34	F42	F50
0100 0000	64	F01	F09	F17	F25	F33	F41	F49
1000 0000	128	F00	F08	F16	F24	F32	F40	F48

Grâce au PRGM "FLAGS" il est facile de calculer les sommes des décimales de chaque colonne de drapeaux concernée afin de fournir le code pour le PRGM de ROBERT SCHWARTZ "LBX" permettant de sortir une chaîne alpha programmable.
Le formatage des drapeaux se fait ensuite par:CLX,chaîne obtenue, X()M, et STO d.

```
LBL'FLAGS
END          129 BYTES
.END.        04 BYTES
```

```
01♦LBL "FLAGS"
02 CLRG
03 FIX 0
04 9
05 STO 09
06♦LBL 01
07 STO IND 09
08 1
09 ST+ 09
10 "FLAG ?"
11 PROMPT
12 FS?C 22
13 GTO 01
14 .007
15 STO 08
16 256
17♦LBL 02
18 2
19 /
20 STO IND 08
21 ISG 08
22 GTO 02
23 1
24 ST- 09
25  E3
26 ST/ 09
27  E1
28 ST+ 09
29♦LBL 03
30 102
31 RCL IND 09
32 8
33 /
34 +
35 STO 08
36 RCL IND 09
37 8
38 MOD
39 RCL IND X
40 ST+ IND 08
41 ISG 09
42 GTO 03
43♦LBL A
44 ","
45 ASTO H
46 "247,"
47 ARCL A
48 ARCL H
49 ARCL B
50 ARCL H
51 ARCL C
52 ARCL H
53 ARCL D
54 ARCL H
55 ARCL E
56 ARCL H
57 PRA
58 CLA
59 ARCL F
60 ARCL H
61 ARCL G
62 ARCL H
63 PRA
64 END
```

PRP "F"

```
01♦LBL "F"
02 - E
03 STO 01
04 CLX
05 "♦♦α♦α="
06 X<> [
07 STO d

08♦LBL "A"
09  E
10 ST+ 01
11 FC? IND 01
12 GTO "A"
13 "OUI F- "
14 ARCL 01
15 PRA
16 GTO "A"
17 END
```

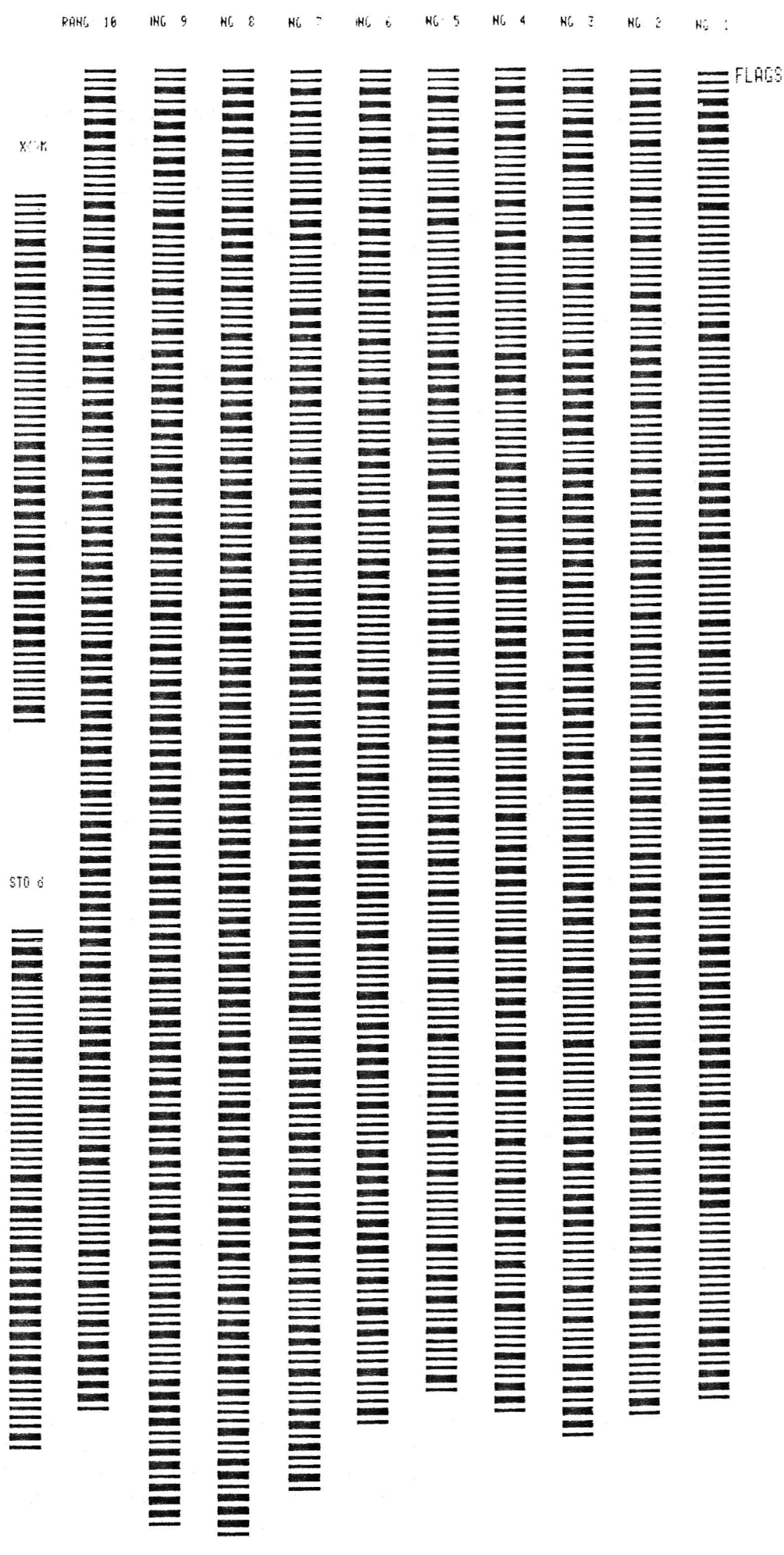
CAT 1

```
LBL'F
LBL'A
END          52 BYTES

LIGNE01 = 192 0 242 0 70
LIGNE02 = 28 27
LIGNE05 = 247 0 0 4 0
          128 1
LIGNE08 = 192 0 242 0 65
LIGNE12 & 16 = 29 241 65

F
                  ASTO X
                XEQ IND X
OUI F- 21,0000
OUI F- 37,0000
OUI F- 40,0000
OUI F- 55,0000
```

	0000	0
016	0001	1
032	0010	2
048	0011	3
064	0100	4
080	0101	5
096	0110	6
112	0111	7
128	1000	8
144	1001	9
160	1010	10 A
176	1011	11 B
192	1100	12 C
208	1101	13 D
224	1110	14 E
240	1111	15 F

```
LBL"FLAGS
END          129 BYTES
.END.        04 BYTES
```

```
01♦LBL "FLAGS"
02 CLRG
03 FIX 0
04 9
05 STO 09
06♦LBL 01
07 STO IND 09
08 1
09 ST+ 09
10 "FLAG ?"
11 PROMPT
12 FS?C 22
13 GTO 01
14 .007
15 STO 08
16 256
17♦LBL 02
18 2
19 /
20 STO IND 08
21 ISG 08
22 GTO 02
23 1
24 ST- 09
25  E3
26 ST/ 09
27  E1
28 ST+ 09
29♦LBL 03
30 102
31 RCL IND 09
32 8
33 /
34 +
35 STO 08
36 RCL IND 09
37 8
38 MOD
39 RCL IND X
40 ST+ IND 08
41 ISG 09
42 GTO 03
43♦LBL A
44 ","
45 ASTO H
46 "247,"
47 ARCL A
48 ARCL H
49 ARCL B
50 ARCL H
51 ARCL C
52 ARCL H
53 ARCL D
54 ARCL H
55 ARCL E
56 ARCL H
57 PRA
58 CLA
59 ARCL F
60 ARCL H
61 ARCL G
62 ARCL H
63 PRA
64 END
```



TISSERAND
42, Av. du Gal de Gaulle
77500 CHELLES

Si il est très facile d'utiliser les lettres minuscules avec les imprimantes thermiques, il est beaucoup plus difficile de le faire dans les fichiers ASCII du X-Fonctions, ou avec les imprimantes grande largeur comme la SEIKOSHA GP-100/HP-IL, pour laquelle les indicateurs 12 et 13 sont inopérants. Il faut alors soit générer les caractères un à un par leur code ASCII, ou utiliser la progammation synthétique, soit utiliser un M.L.D.L.,très pratique, mais coûteux.

Les programmes ci-dessous, un peu lents, permettent de pallier à tout ces inconvéniants:

"MA" remplace chaque minuscule rencontrée en mémoire ∝ par sa majuscule correspondante, en laissant les autres caractères inchangés.

"MI" remplace de même chaque majuscule en sa minuscule correspondante.

"MAMI" exécute l'un des programmes MA ou MI, conformément à l'indicateur 13

"MAMIL" fait comme MAMI, mais ajoute en tête de la chaîne ∝ un caractère de contrôledes imprimantes, conformémant à l'indicateur 12. Ne pas mettre plus de 23 caractères en ∝ avant d'utiliser ce programme!

Ces programmes ont été optimisés pour modifier le moins possible l'état de la 41; seuls sont modifiés les registres Z et T.

Ils nécessitent la précence du module X-FONCTIONS.

	CF 12	SF 12
CF 13	Majuscules Simple largeur	Majuscules Double largeur
SF 13	Minuscules Simple largeur	Minuscules Double largeur

Jean-Claude BECKER

```
01*LBL "MAMIL"
FS? 12 14 FC? 12 15
XTOA RDN -1 ARCT
 10*LBL "MAMI"
FS? 13 GTO "MI"
 13*LBL "MA"
ALENG ,99 - 1 E3 /
 19*LBL 01
ATOX 97 X<>Y X<Y?
GTO 02 X<>Y RDN 122
X<>Y X>Y? GTO 02 32
-
 33*LBL 02
XTOA RDN RDN ISG X
GTO 01 AVIEW RTN
 41*LBL "MI"
ALENG ,99 - 1 E3 /
 47*LBL 03
ATOX 65 X<>Y X<Y?
GTO 04 X<>Y RDN 90
X<>Y X>Y? GTO 04 32
+
 61*LBL 04
XTOA RDN RDN ISG X
GTO 03 AVIEW END
```

```
CF 12
CF 13
ABC/abc = 10+8, $= ?
XEQ "MA"
ABC/ABC = 10+8, $= ?
XEQ "MA"
ABC/ABC = 10+8, $= ?
XEQ "MI"
abc/abc = 10+8, $= ?
XEQ "MI"
abc/abc = 10+8, $= ?
XEQ "MAMIL"
¤ABC/ABC = 10+8, $= ?
```

COURRIER DU COEUR

Olivier Dancer 39, rue St Fargeau 75020 PARIS, vends:
1imprimante 82143 (à 70% de son prix neuf, accu neuf à charge rapide
1 lecteur de codes barres, 1 interface HPIL, 1 bloc d'accus 82120 (HP41)
1 module XFONCTION, 1 module HOME MANAGEMENT (82106), 1 module AVIATION,
1 valise de rangement (500FF), 1 boite d'eprom BE-01-4k (DIDIER JEHL) (400FF)

LE PETIT THEATRE

Bondy, le 30.10.83

MLDL ou LECTeur d'EPROM ?

Avant de décider de l'achat d'un tel matériel, il semble judicieux de savoir ce qu'on désire obtenir et pour cela, il est indispensable de connaître les possibilités offertes.

Un lecteur d'EPROM, comme son nom l'indique, ne permet que de lire des EPROM "préPRGMmées" et qui peuvent donc contenir des Fonctions (donc des PRGM !) telles que, par exemple, celles d'un module genre X-FUNCTIONS ou, nos PRGM personnels écrits en langage normal (avec PS, éventuellement) ou en MC. En résumé, un LECT d'EPROM permet d'obtenir le même résultat que celui obtenu avec un module d'application (genre MATH1) ou un module d'extension de Fonctions (genre HP-IL·DEV).avec, en compensation de l'inconvénient du "fil-à-la-patte", la possibilité de disposer d'un plus grand nbre de F que celles commercialisées par belle-mère HP, d'utiliser des PRGM personnels qui n'encombrent pas la MEM principale mais dans ce dernier cas, il est nécessaire de s'adresser à un Club ou à un professionnel qui introduira en EPROM vos PRGM préférés. Précisons tout de suite qu'un MLDL ne permet pas le travail précité (la PRGMmation d'EPROM).

Si nous voulons rester Français, je vous indique deux adresses où vous pourrez obtenir un LECT d'EPROM: 1°) LECT ref. BE-01-4K (4kOctets)

Didier JEHL (8116T80P39)
Appartement N° 8
9, rue Utrillo
Résidence les Hochettes
62000 ARRAS
tél.: (31) 23 00 77

2°) Etienne POUPEE
SCIP
83, avenue de PARIS
94800 VILLEJUIF
tél.: (1) 678 17 57

J'allais oublier qu'il est possible d'effacer le contenu des EPROM en les passant sous des rayons UV et donc, de réutiliser ces EPROM "PURFL" ou plutôt, "MEMORY LOSTées".

Un MLDL permet de PRGMmer en MC (LANGage MACHine) et comporte également un lecteur d'EPROM qui est bien utile car il permet, avec un jeu d'EPROM adéquat (-MLEPROM 1H de Stéphane BARIZIEN, commercialisé par E.POUPEE), une PRGMmation MC beaucoup plus facile. Ce lecteur d'EPROM permet également d'être utilisé comme un LECT d'EPROM "ordinaire" mais si vous disposez déjà d'un lecteur, il pourra être utilisé simultanément avec un MLDL en veillant simplement à ne pas se tromper dans les adressages ROM. Rassurez-vous, les erreurs ne sont pas dangereuses pour votre cher matériel !

La PRGMmation en MC s'effectue dans la RAM du MLDL mais, comme je vous l'ai indiqué précédemment, il n'est pas possible de copier vos PRGM dans des EPROM car pour ce faire il est indispensable de disposer d'un PRGMmeur d'EPROM dont le prix est assez dissuasif.... Dans cette RAM il est également possible de STO des PRGM en LANG normal (avec ou sans PS) et les PRGM en MC peuvent être sauvegardés sur CART ou sur K7.Cette dernière possibilité permet d'envoyer vos créations MC afin qu'elles soient mises en EPROM. Notre Président pourra vous indiquer les possibilités du PC.

Je ne suis pas sûr d'avoir été plus clair que ce qui a déjà été écrit sur le sujet et j'invite les connaisseurs, utilisateurs d'EPROM (LECT, MLDL ou autres POTOCODER), à donner toute indication complémentaire (ou rectificative ?).

Je vais terminer en indiquant le mode d'emploi de 2 F en MC contenues dans l'EPROM de Stéphane BARIZIEN (utilisations possibles avec un simple LECT d'EPROM) :

ASN "LB" à la touche de son choix (si PPC-ROM dans un port, le ou la retirer), touche L, par ex.
mode PRGM ENTER↑ (ou toute autre F)
L (touche STO = touche 33) en mode USER
AFFichage : LB
156 pp ENTER↑
L Ø1Ø ← pp FIX Ø

le FIX Ø (zéro) obtenu est en fait un FIX 9, c'est-à-dire un 'trompe-couillon'

ENTER↑ L 246 L 114 L 111 LØ98 L 1Ø1 L 114 L 116 ← AFFichage : pp 'ØØbeØØ

qui est bien la CHaîne de CAR désirée et tout ceci sans aucun PRGM "de service" en MEM principale et avec une rapidité d'exécution surprenante.

Préfixe (DEC) ENTER↑
suffixe d° d°

CODE touche XEQ "XYZASN" et en une fraction de seconde, nous avons ASN une F à la touche adéquate.

Vous aurez compris pourquoi désormais je vais indiquer les CODEs DEC sur 3 chiffres.....

J'indiquerais que je ne comprends pas pourquoi cette technique n'a pas, en France, rencontré le succès qu'elle méritait mais j'espère que nous allons essayer de rattraper notre retard.

rS (T178+P2Ø)

P.S.: En me relisant, je me suis aperçu que j'avais écrit une âneries. Un MLDL ne comprend pas toujours un lecteur d'EPROM incorporé. Dans la majeure partie des cas, c'est même le contraire et je dois dire que je pense que c'est judicieux. En effet, il ne me semble pas réaliste d'envisager de PRGMmer en MC dans le train ou, d'une manière générale, ailleurs que dans une ambiance calme alors qu'on peut admettre d'utiliser des PRGM avec le LECT d'EPROM dans sa poche. Les 2 éléments séparés ont donc ma préférence.

Aujourd'hui dans le petit théatre des microcodes, nous allons voir la formation des XEQs et des GTOs.

Nous savons tous que le branchement à une adresse se fait soit par un saut (JC ou JNC, qui sont des instructions de classe 3), soit par un branchement absolu (XEQ ou GTO, qui sont des instructions de classe 1.

* Les saut relatifs sont sur 1 mot ROM, et il est impossible de faire un saut de plus de +63 octets (en descendant le programme) ou de moins de -64 octets (en remontant dans le prgramme).

* Les branchements à une adresse absolue et qui sont des instructions (?NCXQ, ?CXQ, ?NCGO et ?CGO) sont sur deux mots ROM et dépendent de l'état du drapeau CARRY. On se branche à une adresse absolue car l'éxécution se fera toujours à la même adresse, quelquesoit la position des ROM. Il faudra être sur que cette adresse ne sera jamais changée

par exemple on pourra faire des branchements dans chacune des trois ROM du système opératoires, dans l'imprimante (toujours en page 6), l'HPIL (toujours en page 5), le lecteur de cartes magnétiques (toujours en page E),

* Sont possibles aussi, les branchements à une adresse relative. C'est un branchement qui se fait dans la même ROM quelquesoit la page dans laquelle elle se trouve. Ce branchement est independant du port contenant la ROM (l'adresse est réallouée en fonction de la page dans laquelle se trouve le programme). Une ROM quelquonque (ROM HP, MLDL, PROTOCODER, EPROM, ...) peut être mise dans n'importe quelle page du calculateur (pages 3 à F), et l'appel normal des instructions de classe 1 sont utilisées pour des sous routines dans le même block de 4k de la ROM et un branchement à l'adresse Axxx fonctionnera parfaitement si le programme utilisé reste en page A soit la ROM dans les adresses allant de AØØØ à AFFF.

Ceux qui désirent faire leur propre ROM et avec des branchement dans différentes adresses du programme, indépendemment du port utilisé, doivent:

-utiliser des branchements écrits sur trois MOT ROM;

-ne pas avoir besoin du registre C (du CPU) car celui-ci est perdu et ne peut contenir les données utilisées dans la sous sous routine appelée;

-le CPU doit être en mode héxadécimal (26Ø SETHEX)

-ils ne peuvent être utilisés tels que et doivent être pris aprés 1 saut relatif.

```
F000 001
F001 001
F002 000
F003 00B
F004 000
F005 000
F006 089 "I"
F007 001 "A"
F008 013 "S"
F009 013 "S"
F00A 005 "E"
F00B 3B0
F00C 002 ?NCGO 00EF
F00D 025
F00E 3C1 ?CXQ F009
F00F 023 JNC F013 +04
F010 349
F011 08C PORT DEP:
F012 00D XQ F00D
F013 3E0 RTN
```

```
C000 001
C001 001
C002 000
C003 00B
C004 000
C005 000
C006 089 "I"
C007 001 "A"
C008 013 "S"
C009 013 "S"
C00A 005 "E"
C00B 3B0
C00C 002 ?NCGO 00EF
C00D 025
C00E 3C1 ?CXQ F009
C00F 023 JNC C013 +04
C010 349
C011 08C PORT DEP:
C012 00D XQ C00D
C013 3E0 RTN
```

Vous pouvez remarquer que le seul branchement (XEQ) qui s'est adapté à sa nouvelle adresse est le dernier.
En page F, il se branche en: FØØD
En page C, il se branche en: CØØD.

XQ 00F

1° MOT: 349 1101001001
2° MOT: 08C
3° MOT: 00F

XQ 33E

1° MOT: 349 1101001001
2° MOT: 08C
3° MOT: 33E

XQ 4DD

1° MOT: 36D 1101101101
2° MOT: 08C
3° MOT: 0DD

XQ 554

1° MOT: 36D 1101101101
2° MOT: 08C
3° MOT: 154

XQ 777

1° MOT: 36D 1101101101
2° MOT: 08C
3° MOT: 377

XQ 9BC

1° MOT: 391 1110010001
2° MOT: 08C
3° MOT: 1BC

XQ C00

1° MOT: 3B5 1110110101
2° MOT: 08C
3° MOT: 000

XQ FFF

1° MOT: 3B5 1110110101
2° MOT: 08C
3° MOT: 3FF

GO 00F

1° MOT: 341 1101000001
2° MOT: 08C
3° MOT: 00F

GO 33E

1° MOT: 341 1101000001
2° MOT: 08C
3° MOT: 33E

GO 4DD

1° MOT: 365 1101100101
2° MOT: 08C
3° MOT: 0DD

GO 554

1° MOT: 365 1101100101
2° MOT: 08C
3° MOT: 154

GO 777

1° MOT: 365 1101100101
2° MOT: 08C
3° MOT: 377

GO 9BC

1° MOT: 389 1110001001
2° MOT: 08C
3° MOT: 1BC

GO C00

1° MOT: 3AD 1110101101
2° MOT: 08C
3° MOT: 000

GO FFF

1° MOT: 3AD 1110101101
2° MOT: 08C
3° MOT: 3FF

Vous remarquerez que le 1°mot caractérise le branchement, le deuxième mot est toujours Ø8C, et le troisième mot est toujours les trois derniers caractères de l'adresse ou on désire se brancher.

La structure d'un XEQ est:
11Ø1ØØ1ØØ1 (349)
Le passage d'un XEQXnnn à un XEQXn+4nnse fait en ajoutant IØØ1ØØ (24) à chaque fois.

La structure d'un GTO est:
11Ø1ØØØØ1 (341)
Le passage au GTO suivant adresse précédante + 4ØØ, se fait en ajoutant la même valeur soit
1ØØ1ØØ (24)

PHILIPPE

END

SIZE 0

SIZE 1

SIZE 2

SIZE 3

SIZE 4

SIZE 5

SIZE 6

SIZE 7

SIZE 8

SIZE 9

SIZE 10

SIZE 11

SIZE 12

SIZE 13

SIZE 14

SIZE 15

SIZE 16

SIZE 17

SIZE 18

SIZE 19

SIZE 20

SIZE 21

SIZE 22

SIZE 23

SIZE 24

SIZE 25

SIZE 26

SIZE 27

SIZE 28

SIZE 29

SIZE 30

SIZE 40

SIZE 50

SIZE 60

SIZE 70

SIZE 80

SIZE 100

SIZE 120

SIZE 150

SIZE 180

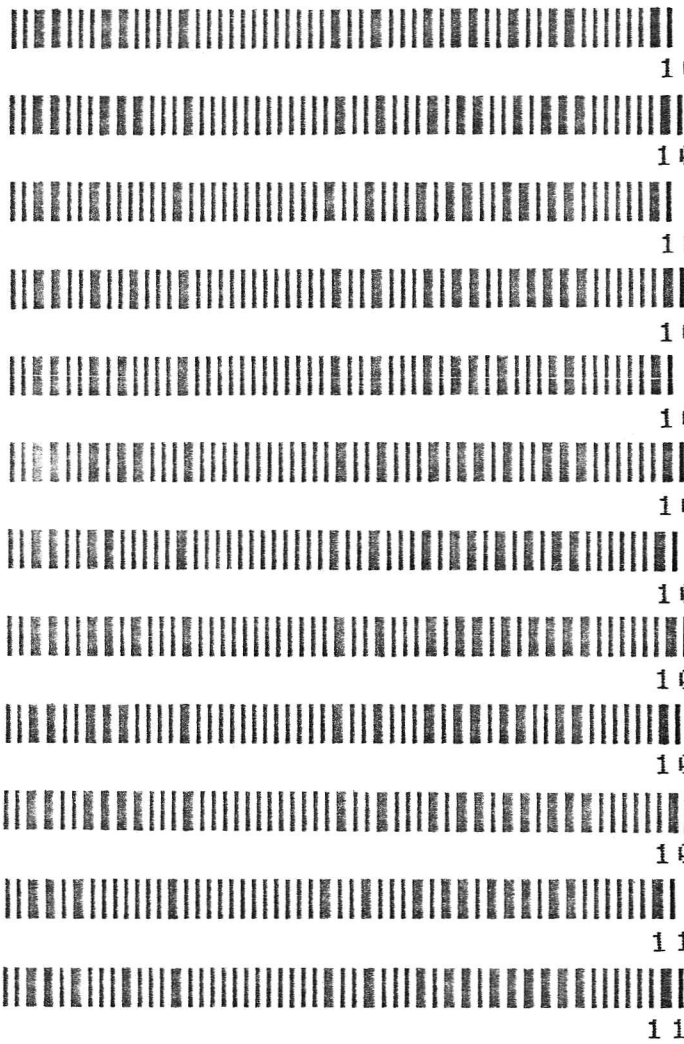
SIZE 200

SIZE 250

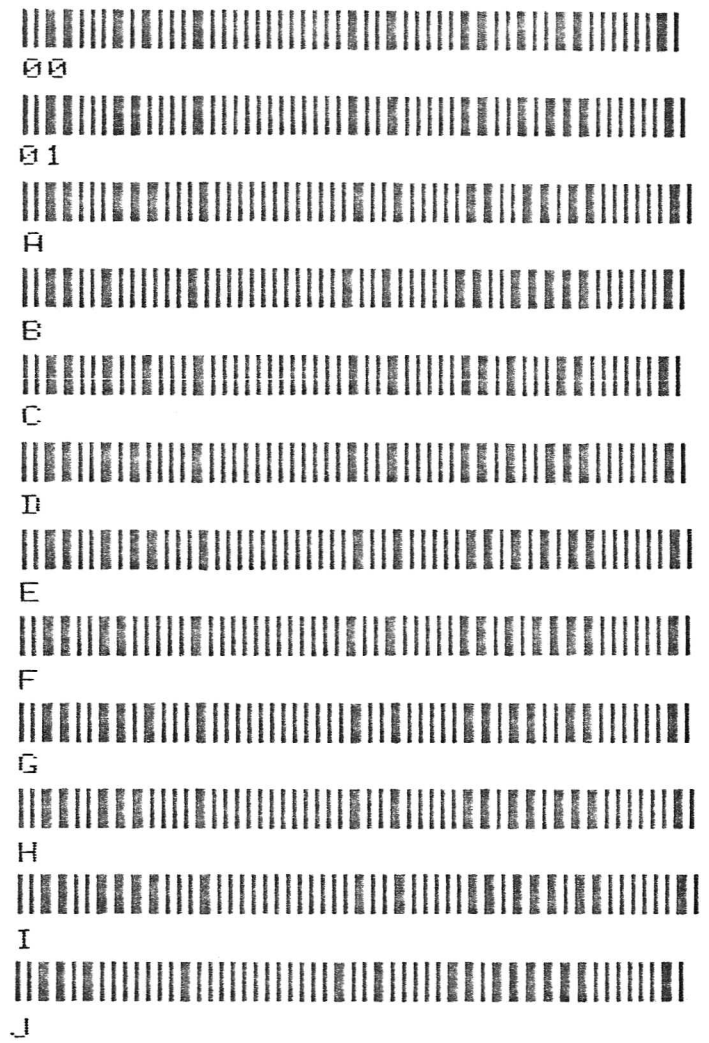
SIZE 255

P 24

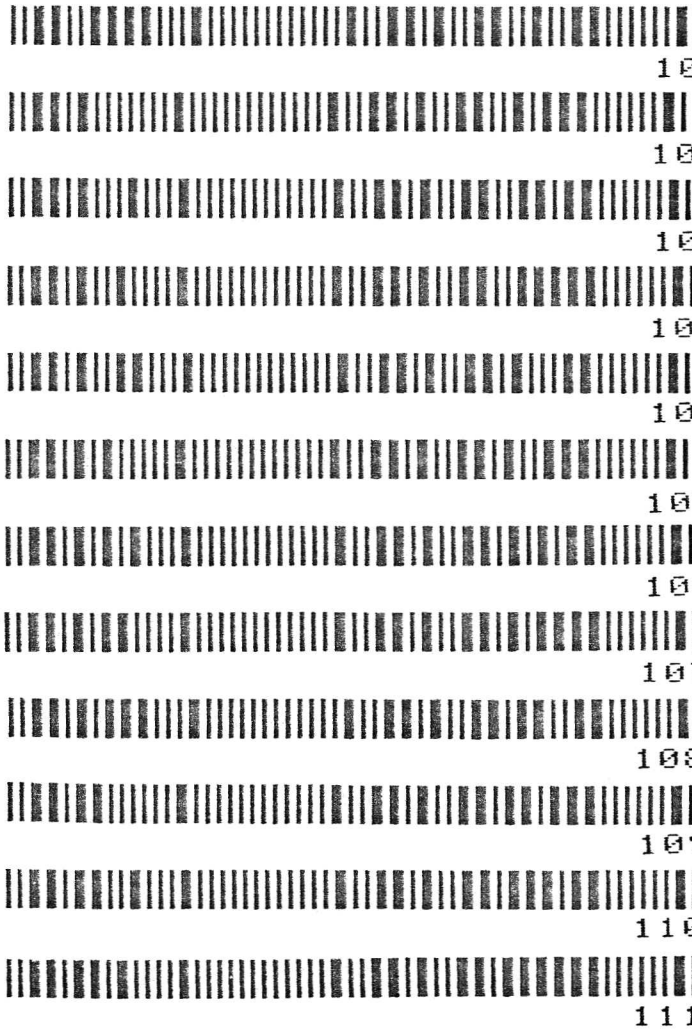
STO 100 a 111



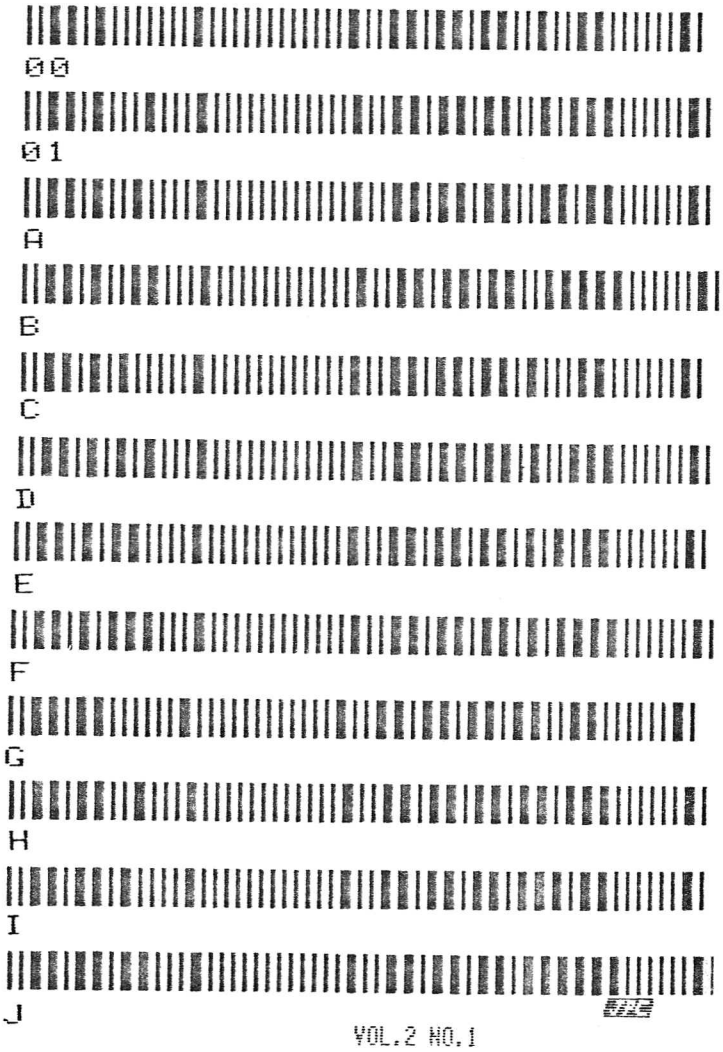
RCL 100 a 111



ASTO 100 a 111



ARCL 100 a 111



P54

STO IND 100 a 111

100

101

102

103

104

105

106

107

108

109

110

111

RCL IND 100 a 111

00

01

A

B

C

D

E

F

G

H

I

J

X<> IND 100 a 111

100

101

102

103

104

105

106

107

108

109

110

111

ISG IND 100 a 111

00

01

A

B

C

D

E

F

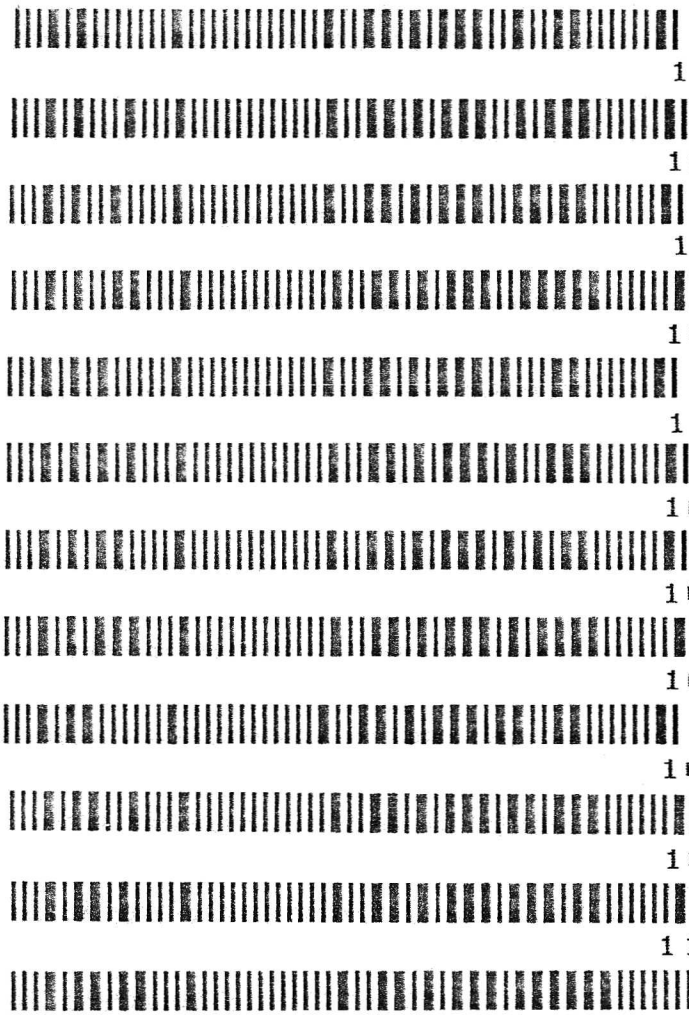
G

H

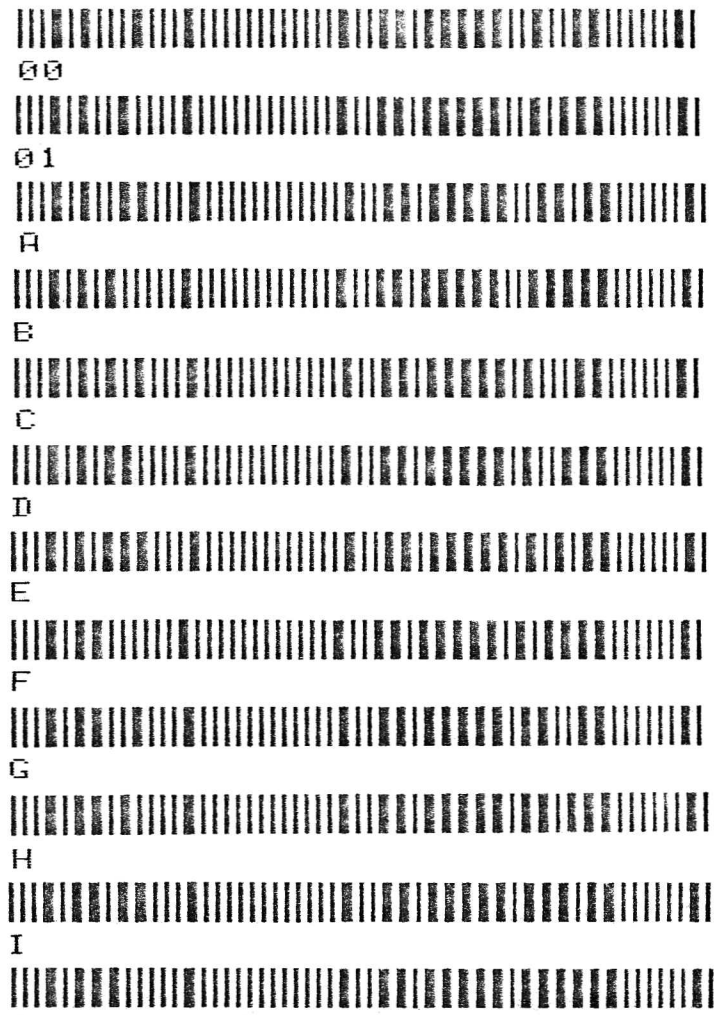
I

J

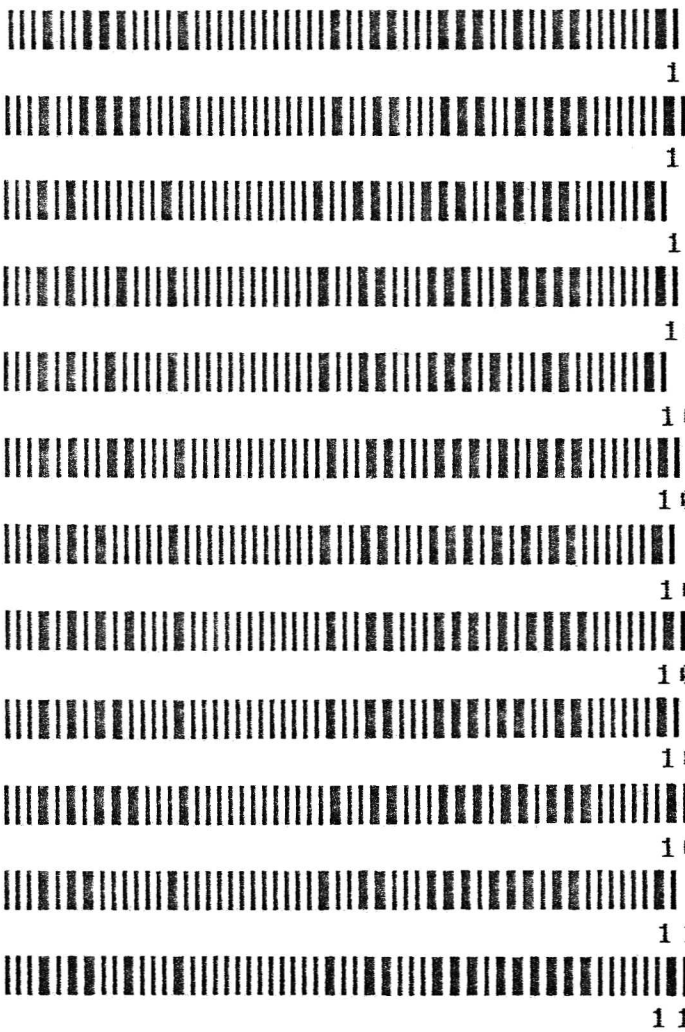
ASTO IND 100 a 111



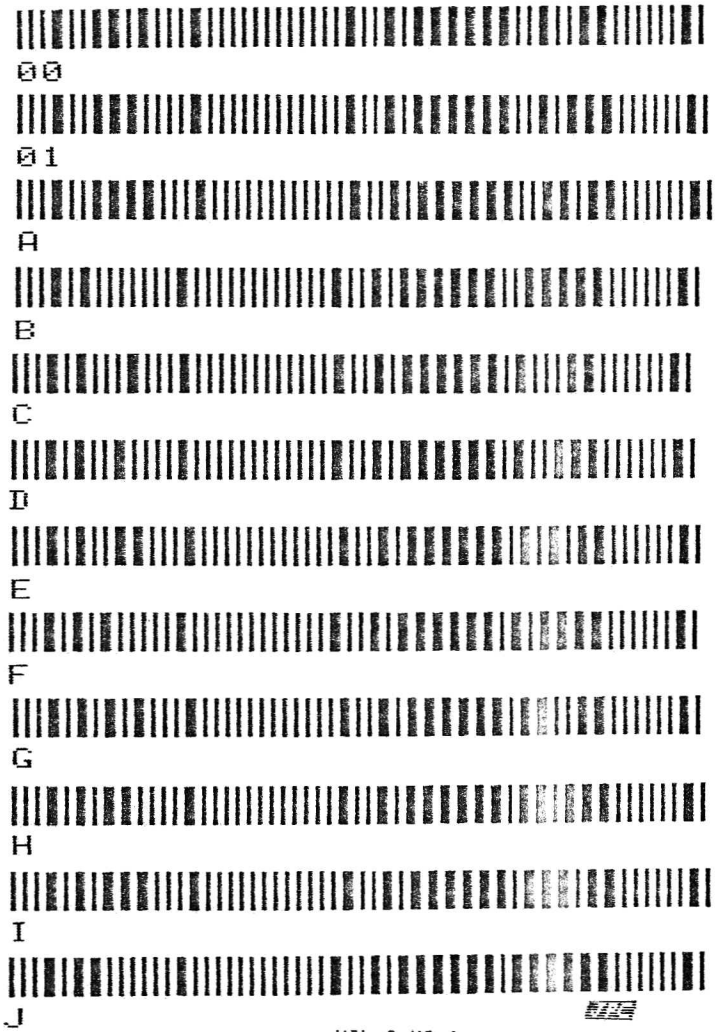
ARCL IND 100 a 111



VIEW IND 100 a 111

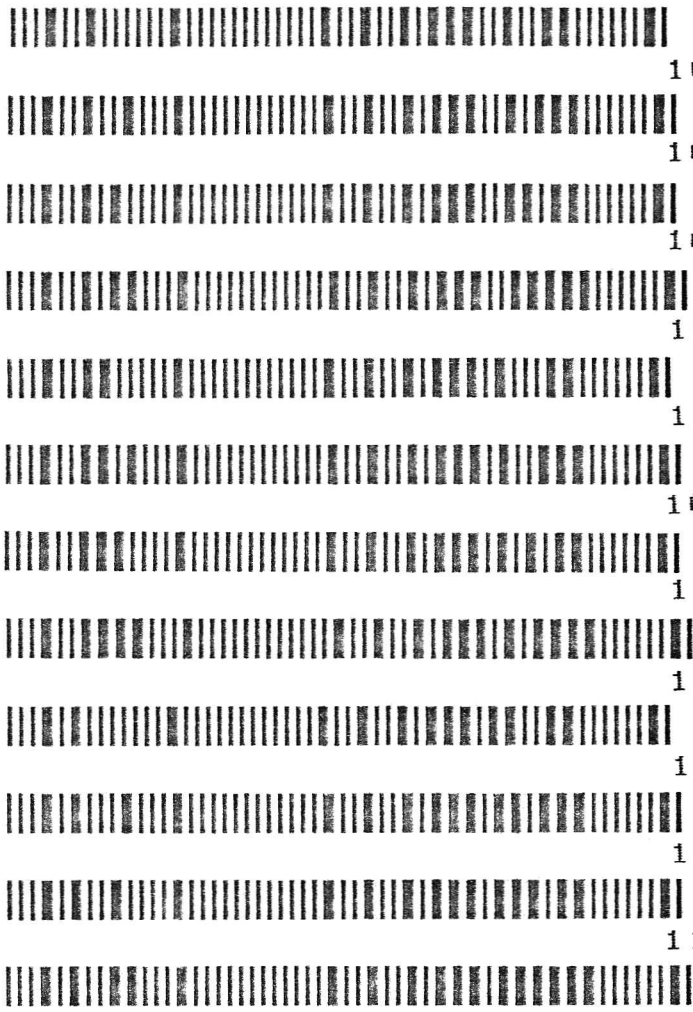


DSE IND 100 a 111

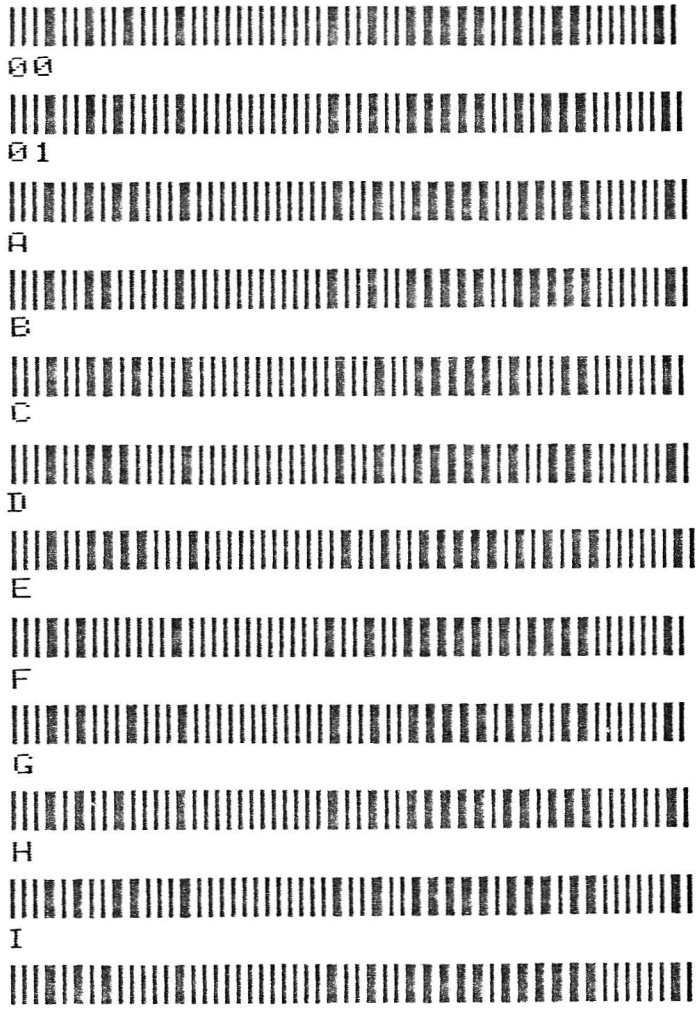


P54

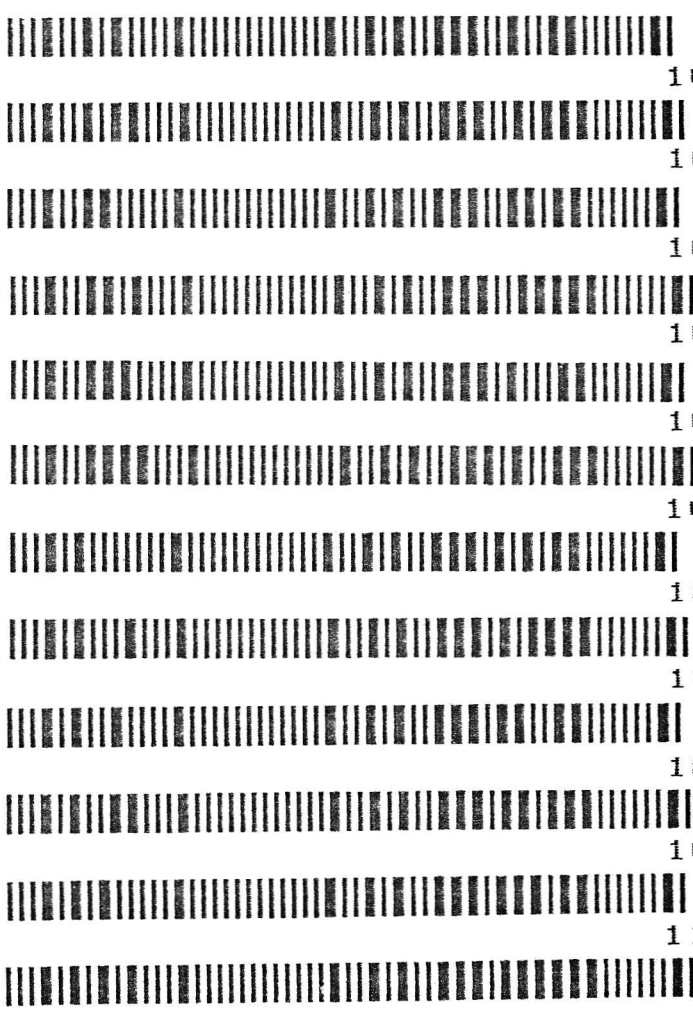
ST+ IND 100 a 111



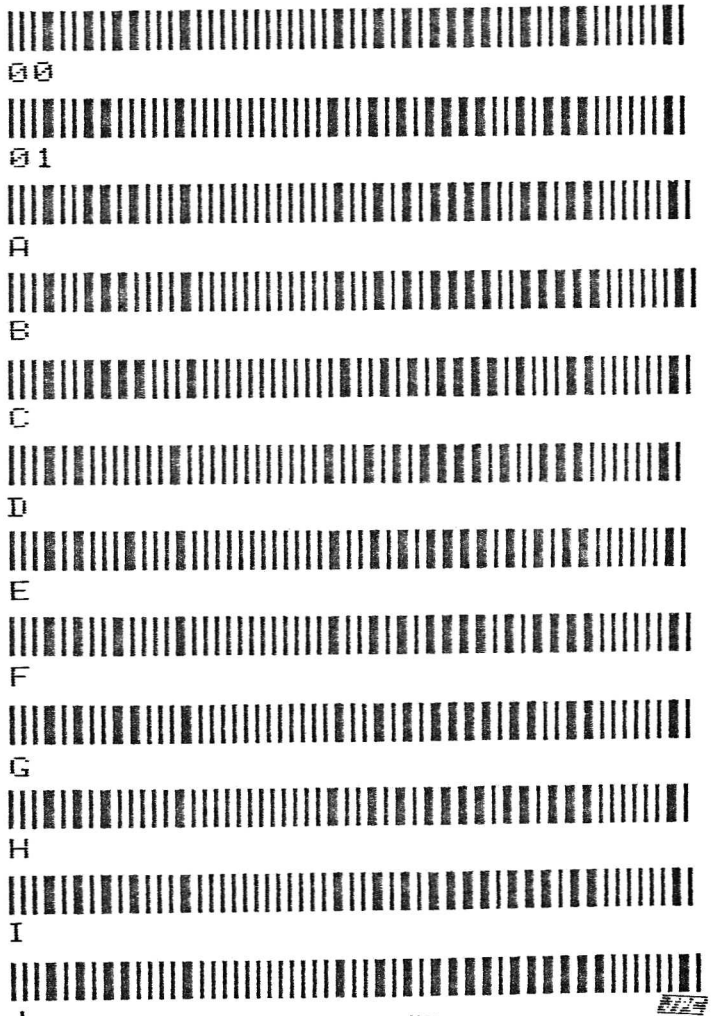
ST- IND 100 a 111



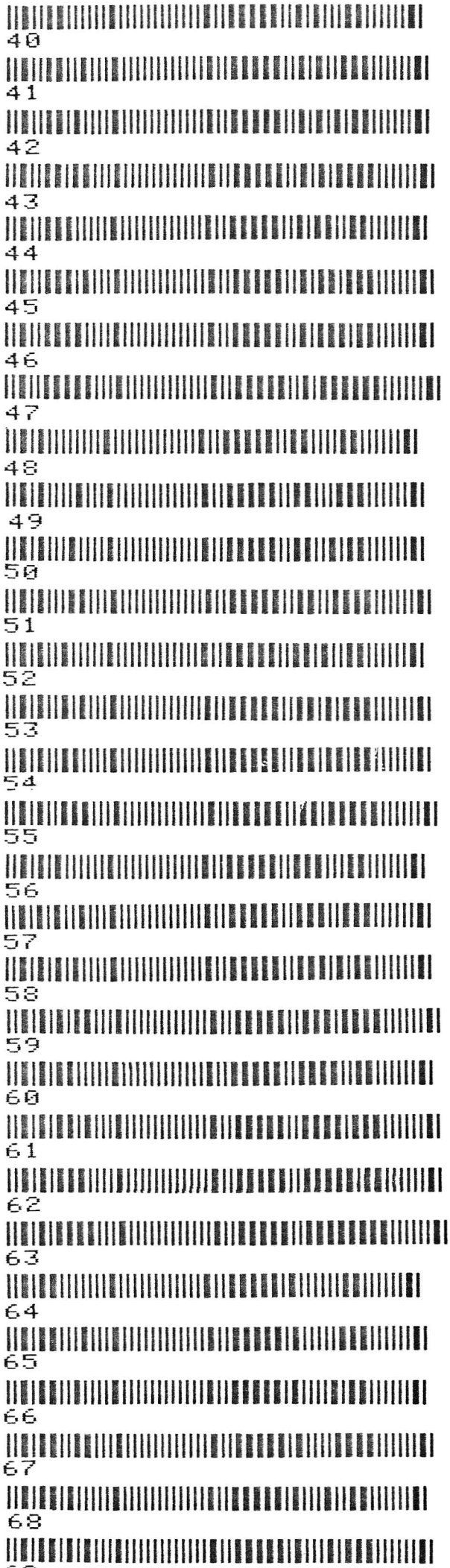
ST* IND 100 a 111



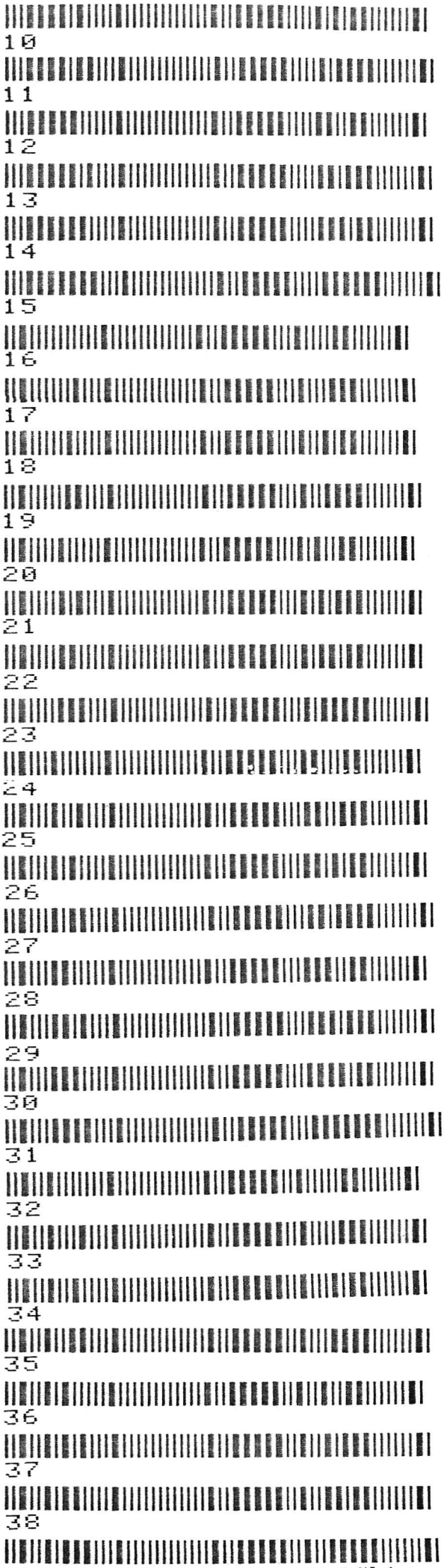
ST/ IND 100 a 111



TONES 40 a 69

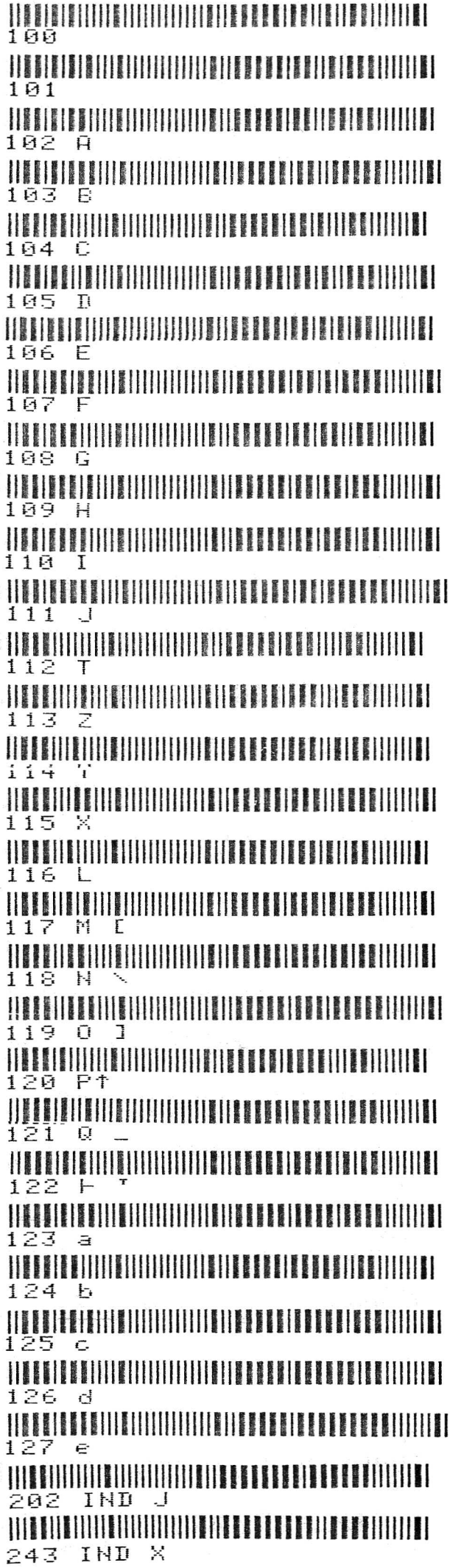


TONES 10 a 39

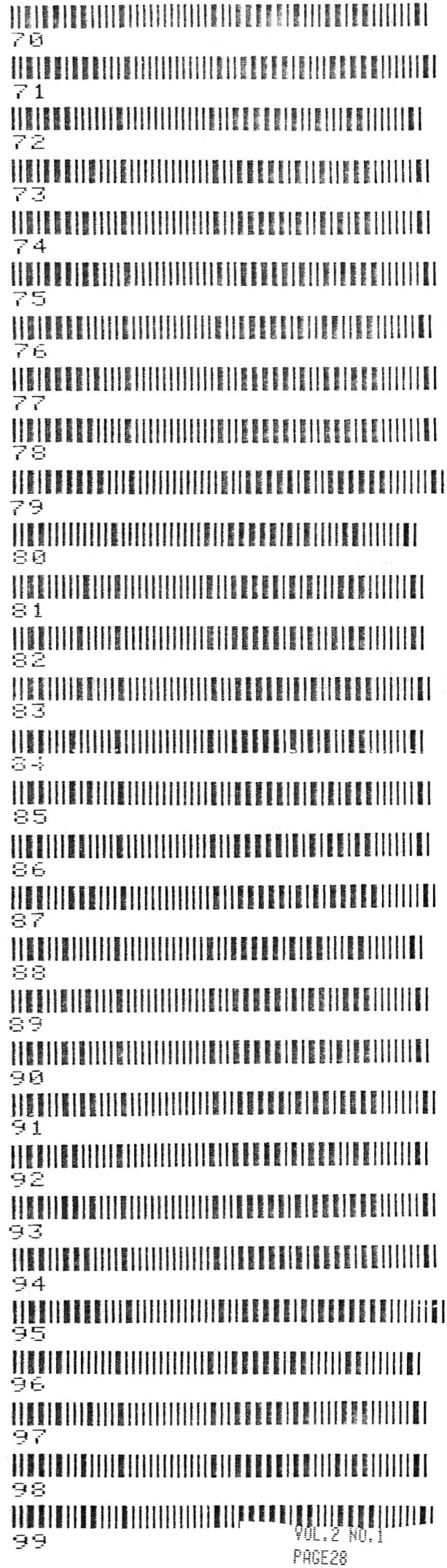


P54

TONES 100 a 127



TONES 70 a 99



P54

XROM 20,0

XROM 20,1

XROM 20,2

XROM 20,3

XROM 20,4

XROM 20,5

XROM 20,6

XROM 20,7

XROM 20,8

XROM 20,9

XROM 20,10

XROM 20,11

XROM 20,12

XROM 20,13

XROM 20,14

XROM 20,15

XROM 20,16

XROM 20,17

XROM 20,18

XROM 20,19

XROM 20,20

XROM 20,21

XROM 20,22

XROM 20,23

XROM 20,24

XROM 20,25

XROM 20,26

XROM 20,27

XROM 20,28

XROM 20,29

XROM 20,30

XROM 20,31

XROM 20,32

XROM 20,33

XROM 20,34

XROM 20,35

XROM 20,36

XROM 20,37

XROM 20,38

XROM 20,39

XROM 20,40

XROM 20,41

XROM 20,42

XROM 20,43

XROM 20,44

XROM 20,45

XROM 20,46

XROM 20,47

XROM 20,48

XROM 20,49

XROM 20,50

XROM 20,51

XROM 20,52

XROM 20,53

XROM 20,54

XROM 20,55

XROM 20,56

XROM 20,57

XROM 20,58

XROM 20,59

XROM 20,60

XROM 20,61

XROM 20,62

XROM 20,63

XROM 21,0

XROM 21,1

XROM 21,2

XROM 21,3

XROM 21,4

XROM 21,5

XROM 21,6

XROM 21,7

XROM 21,8

XROM 21,9

XROM 21,10

XROM 21,11

XROM 21,12

XROM 21,13

XROM 21,14

XROM 21,15

XROM 21,16

XROM 21,17

XROM 21,18

XROM 21,19

XROM 21,20

XROM 21,21

XROM 21,22

XROM 21,23

XROM 21,24

XROM 21,25

XROM 21,26

XROM 21,27

XROM 21,28

XROM 21,29

XROM 21,30

XROM 21,31

XROM 21,32

XROM 21,33

XROM 21,34

XROM 21,35

XROM 21,36

XROM 21,37

XROM 21,38

XROM 21,39

XROM 21,40

XROM 21,41

XROM 21,42

XROM 21,43

XROM 21,44

XROM 21,45

XROM 21,46

XROM 21,47

XROM 21,48

XROM 21,49

XROM 21,50

XROM 21,51

XROM 21,52

XROM 21,53

XROM 21,54

XROM 21,55

XROM 21,56

XROM 21,57

XROM 21,58

XROM 21,59

XROM 21,60

XROM 21,61

XROM 21,62

XROM 21,63

XROM 22,0

XROM 22,1

XROM 22,2

XROM 22,3

XROM 22,4

XROM 22,5

XROM 22,6

XROM 22,7

XROM 22,8

XROM 22,9

XROM 22,10

XROM 22,11

XROM 22,12

XROM 22,13

XROM 22,14

XROM 22,15

XROM 22,16

XROM 22,17

XROM 22,18

XROM 22,19

XROM 22,20

XROM 22,21

XROM 22,22

XROM 22,23

XROM 22,24

XROM 22,25

XROM 22,26

XROM 22,27

XROM 22,28

XROM 22,29

XROM 22,30

XROM 22,31

XROM 22,32

XROM 22,33

XROM 22,34

XROM 22,35

XROM 22,36

XROM 22,37

XROM 22,38

XROM 22,39

XROM 22,40

XROM 22,41

XROM 22,42

XROM 22,43

XROM 22,44

XROM 22,45

XROM 22,46

XROM 22,47

XROM 22,48

XROM 22,49

XROM 22,50

XROM 22,51

XROM 22,52

XROM 22,53

XROM 22,54

XROM 22,55

XROM 22,56

XROM 22,57

XROM 22,58

XROM 22,59

XROM 22,60

XROM 22,61

XROM 22,62

XROM 22,63

XROM 23,0

XROM 23,1

XROM 23,2

XROM 23,3

XROM 23,4

XROM 23,5

XROM 23,6

XROM 23,7

XROM 23,8

XROM 23,9

XROM 23,10

XROM 23,11

XROM 23,12

XROM 23,13

XROM 23,14

XROM 23,15

XROM 23,16

XROM 23,17

XROM 23,18

XROM 23,19

XROM 23,20

XROM 23,21

XROM 23,22

XROM 23,23

XROM 23,24

XROM 23,25

XROM 23,26

XROM 23,27

XROM 23,28

XROM 23,29

XROM 23,30

XROM 23,31

XROM 23,32

XROM 23,33

XROM 23,34

XROM 23,35

XROM 23,36

XROM 23,37

XROM 23,38

XROM 23,39

XROM 23,40

XROM 23,41

XROM 23,42

XROM 23,43

XROM 23,44

XROM 23,45

XROM 23,46

XROM 23,47

XROM 23,48

XROM 23,49

XROM 23,50

XROM 23,51

XROM 23,52

XROM 23,53

XROM 23,54

XROM 23,55

XROM 23,56

XROM 23,57

XROM 23,58

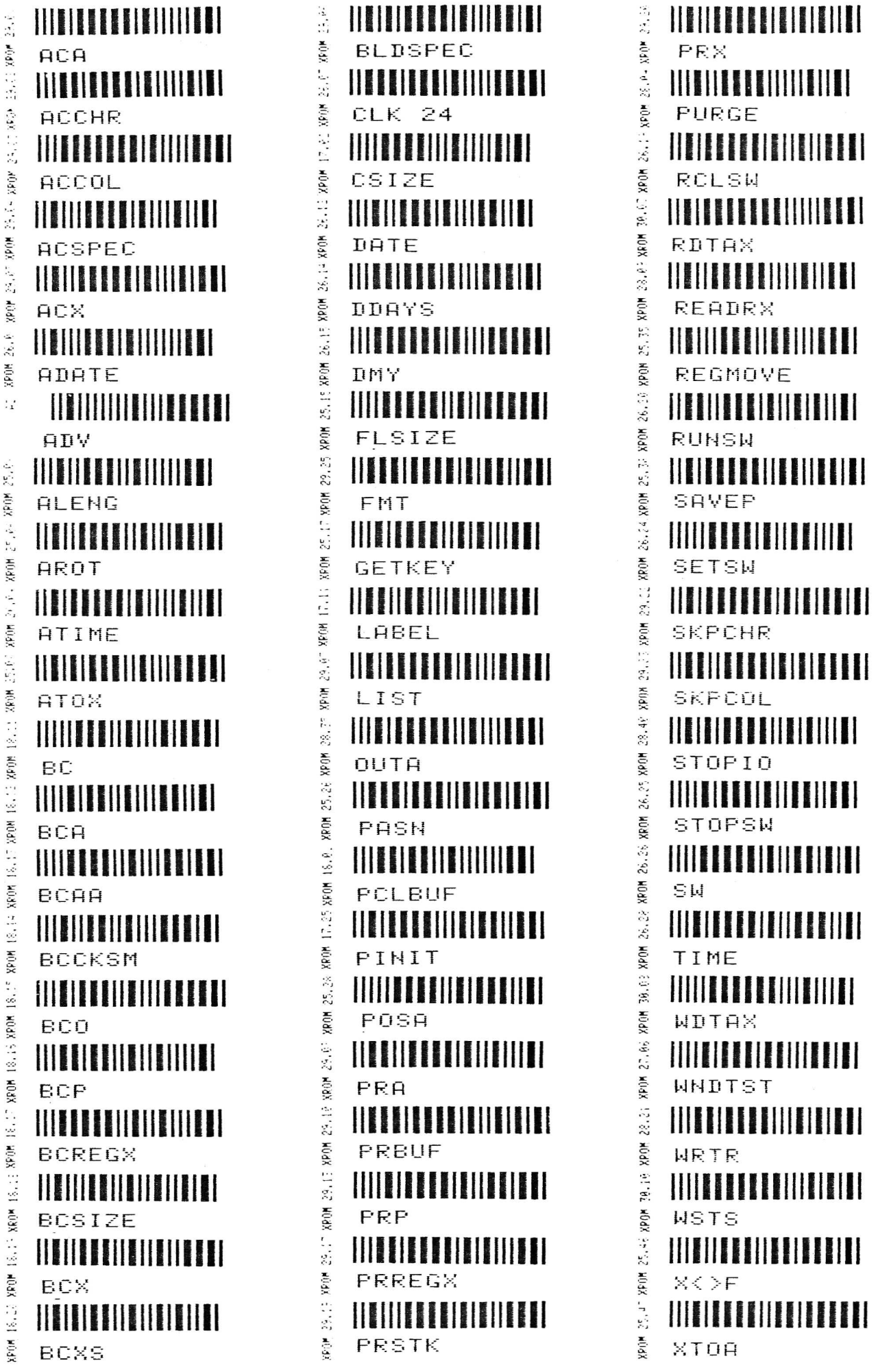
XROM 23,59

XROM 23,60

XROM 23,61

XROM 23,62

XROM 23,63


ACA
ACCHR
ACCOL
ACSPEC
ACX
ADATE
ADV
ALENG
AROT
ATIME
ATOX
BC
BCA
BCAA
BCCKSM
BCO
BCP
BCREGX
BCSIZE
BCX
BCXS
BLDSPEC
CLK 24
CSIZE
DATE
DDAYS
DMY
FLSIZE
FMT
GETKEY
LABEL
LIST
OUTA
PASN
PCLBUF
PINIT
POSA
PRA
PRBUF
PRP
PRREGX
PRSTK
PRX
PURGE
RCLSW
RDTAX
READRX
REGMOVE
RUNSW
SAVEP
SETSW
SKPCHR
SKPCOL
STOPIO
STOPSW
SW
TIME
WDTAX
WNDTST
WRTR
WSTS
X<>F
XTOA

P 54

NOM	PROFESSION - INTERETS	VILLE	PPC n°	PPCPC n°
DURANT		DRANCY		67
	18, rue Gasset 93700			
MICKAEL POCKSTEINER	PPC AUTRICHE	VIENNE		68
	ZICHYGASSE 12 A1140	AUTRICHE		
PASCAL LEROY	ETUDIANT	CORENC		69
	82,avenue Du Gressivaudan 38700			
HUGUES OBERLE	programmation synthét.	ZIMMERSHEIM		70
	1, rue Beaulieu 68440			
LAURENT HOFFART	ETUDIANT HARD + SOFT	LIMOURS		71
	11, rue des Glycines 91470			
PIERRE GREMAUD	ETUDIANT MICRO + PS	BOULOGNE		72
	65bis, rue du point du jour 92100			
STEPHANE HUAULME	ETUDIANT MATH, PHYSIQUE	SURESNES		73
	31, boulevard H. Sellier 92150			
JEAN-PHILIPPE IMBACH		MONTPELLIER		74
	1108, rue de Las Sorbes 34000			
PHILIPPE CANUEL	INGENIEUR MECANIQUE CONTACTS	COURBEVOIE		75
	Sagittaire Nord - app t 282 - 2, rue de L'Alma 92400			
DOMINIQUE TALON	ETUDIANT INFORMATIQUE ECHANGES	COURCELLES		76
	90100			
PHILIPPE DAVASE	CONTACTS	SARTROUVILLE		77
	7, rue Ho Chi Minh - Les Indes 78500			
STEPHANE SAUNIER	ETUDIANT INFORMATIQUES CONTACTS	VERNEUIL s/ SEINE		78
	11, allée de la source 78480			
FREDERIC VADEZ	MICROCODE	IVRY		79
	2, rue Du Colombier 94200			
ROGER CHARPENTIER	INGENIEUR CONTACTS, MICROCODE	GAGNY		80
	20, rue des Rosiers 93220			
VINCENT HERLICQ	ETUDIANT P. SYNTH, MICROCODE	NEUILLY s/ SEINE		81
	3, rue Bertheaux Dumas 92200			
CHRISTIAN JEGOUZO	INSPECTEUR PTT CONTACTS	PARIS		82
	7, rue De La Voute 75012			
JEAN-LOUIS ATTENOUX	ETUDIANT MATH SPE OPTIMISATION DELA 41	YERRES		83
	70, rue Danton 91330			
JEAN-MICHEL RENAUDIN	PILOTE DE LIGNE NAVIGATION	L'ISLE ADAM		84
	19, rue De Paris 95290			
CHRISTIAN MORLOT	ETUDIANT OPTIMISATION DE LA 41	PUTEAUX		85
	141, Résidence Des Rosiers 92800			
PIERRE COLLIGNON	ETUDIANT INFORMATIQUE, PHOTO	VITRY s/ SEINE		86
	49, rue Berlioz 94400			
PIERRE LANGLOIS	ETUDIANT PREPA. ASTRONOMIE, MATHEMATIQUE	PARIS		87
	3, rueDe La Montagne Ste Genevieve 75005			
FRANCOIS ECKERT	PROFESSEUR, CHARGE DE RECHERCHE	ST GERMAIN LES ARPAJON		88
	45, RUE du Dr Babin 91290			
MICHEL GOURGEOT	ENSEIGNANT PROGRAMMATION, FICHIERS	NOGENT		89
	3, rue Schweitzer 60100	CREIL		
FRANCIS ROZANGE				90
	13, rue Pelée			
JEAN-CLAUDE	MARIN D'ETAT (ELECTRONICIEN) ECHANGES	PONT-A-		91
VANDEVELDE	2, impasse de la passerelle 62115	VENDIN		
WALTER BRANDNER		VIENNE		92
	Johanagasse 35A/14 A-1050	AUTRICHE		

PPC PARIS CHAPTER Janvier 1984

Association regie par la loi de 1901, enregistree
a Paris le 2 decembre 1982 sous le numero 82/3240

Bulletin d'adhesion

Nom ¦_¦
Prenom ¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦ Date de naissance ¦_¦_¦/¦_¦_¦/¦_¦_¦
Adresse ¦_¦
¦_¦
¦_¦
Commune ¦_¦
Code Postal ¦_¦_¦_¦_¦_¦ Pays ¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦_¦
Telephone Domicile ¦_¦_¦/¦_¦_¦_¦_¦_¦_¦_¦_¦ Bureau ¦_¦_¦/¦_¦_¦_¦_¦_¦_¦_¦_¦

===

Profession ____
Interets ____

Materiel H-P en votre possession ______________________________

Autre materiel micro informatique ______________________________

===

Comment avez-vous connu PPC Paris Chapter?
Publicite ______________ Magazine ______________ Autre Club ____________
Relations, membres du club, autres ______________________________

===

Que recherchez-vous au sein du PPC Paris Chapter? ______________________

===

Je souhaite adherer au club PPC Paris Chapter conformement aux statuts de l'association. Au mieux de ma connaissance, je declare avoir le droit de fournir tous les programmes et informations que je vous enverrais (sans enfreindre des obligations de secret a l'egard d'autres personnes ou organismes) pour publication dans le journal de liaison, sans obligations ni responsabilite d'aucune sorte (en cas d'utilisation frauduleuse) de la part des dirigeants du PPC Paris Chapter.

Date ¦_¦_¦/¦_¦_¦/19¦_¦_¦
Signature precedee de la mention "Lu et approuve" ______________

===

Le montant de la cotisation au PPC Paris Chapter s'eleve a 30,00 FF.
Paiement effectue le ¦_¦_¦/¦_¦_¦/19¦_¦_¦ a l'ordre de PPC Paris Chapter.
par [ ] Cheque Bancaire Nr __________ Banque __________
[ ] Cheque Postal 3 Volets Nr ____________________
[ ] Mandat Lettre
Joindre a votre inscription une enveloppe timbree a votre adresse.

===

L'abonnement pour un an au journal J-P-C s'eleve a 150,00 FF.
Les anciens numeros sont disponibles : Nr 1 a 3 : 15,00 FF piece, Nr 4 a 10 : 20,00 FF piece.
Frais d'envoi 3,10 FF par numero ou 13,10 FF les 10 numeros.
[ ] Je m'abonne a compter du ¦_¦_¦/¦_¦_¦/19¦_¦_¦ [ ] Je commande les numeros ________________
Paiement effectue le ¦_¦_¦/¦_¦_¦/19¦_¦_¦ a l'ordre de PPC Paris Chapter.
[ ] Cheque Bancaire Nr __________ Banque __________
[ ] Cheque Postal 3 Volets Nr ____________________
[ ] Mandat Lettre

===

Veuillez envoyer toute correspondance a :
Mr Philippe GUEZ, 56 Rue Jean Jacques Rousseau, 75001 PARIS (FRANCE)
Preciser eventuellement votre numero de membre.

===