

ŒIL DE LYNX



ÉDITORIAL

Avec ce numéro 4 s'achève la parution de l'Œil de LYNX, revue d'utilisateurs dont la publication remonte - déjà - au début de l'année 1984. Après un numéro 0 imparfait, nous avons essayé de mieux répondre à vos attentes : présentation plus agréable, et toujours plus de programmes, d'informations, de trucs et de conseils. Vous avez été nombreux à nous écrire et à nous transmettre vos observations, conseils et listings. Si l'Œil de Lynx est devenu une revue qui a pu satisfaire ses lecteurs, soyez-en remerciés, le mérite vous en revient en grande part.

Cette revue a voulu dès l'origine être un lien entre tous les utilisateurs de Lynx, et nous espérons y être parvenu. Avec ce numéro, nous avons essayé de faire mieux, avec un gros cahier de programmes Basic et des informations hard et soft sur le fonctionnement de notre micro.

Enfin, ce n'est pas parce que cette revue s'achève que Segimex ne reste pas à votre disposition, pour vous fournir les renseignements, programmes et périphériques disponibles. Nous essayerons, dans la mesure du possible de continuer à vous donner satisfaction, vos demandes devant désormais nous être adressées directement par courrier à l'adresse suivante :

SEGIMEX
140, Bd Haussmann
75008 Paris

N° 4 AVRIL
1985

GESTION DE DONNÉES

Le Lynx possède un Basic complet, avec un jeu d'instructions hors du commun. Certains programmeurs, plus familiers de Basic moins complets, éprouvent parfois des difficultés à maîtriser toutes les possibilités de leur machine. Jacques Morel a choisi de vous expliquer l'utilisation des instructions de gestion de données (EXT STORE, EXT FETCH, etc.) sous une forme simple et illustrée

Avec six instructions seulement, savez-vous qu'il est possible de transporter des données, variables ou constantes, d'un programme dans un autre, de les changer de nom, de les stocker sur une cassette, et le tout sans les effacer de la version originale.

LA GESTION DES DONNÉES

Les données utilisées dans un programme sont de deux types, celles introduites par le programmeur (les constantes) ou par l'utilisateur (après un GET, un INPUT, etc.) et les résultats intermédiaires calculés par l'ordinateur. Ces résultats intermédiaires peuvent être stockés par le programmeur pour une réutilisation ultérieure. Voici la manière de procéder.

Le Lynx dispose en effet d'une instruction capable de stocker des données calculées au cours d'un programme, EXT STORE. Ainsi le résultat A peut être sauvé sous la forme :

EXT STORE 12,A

Un deuxième résultat, B, sera sauvé sous la forme :

EXT STORE 12,B

Ou encore, les deux ensembles :

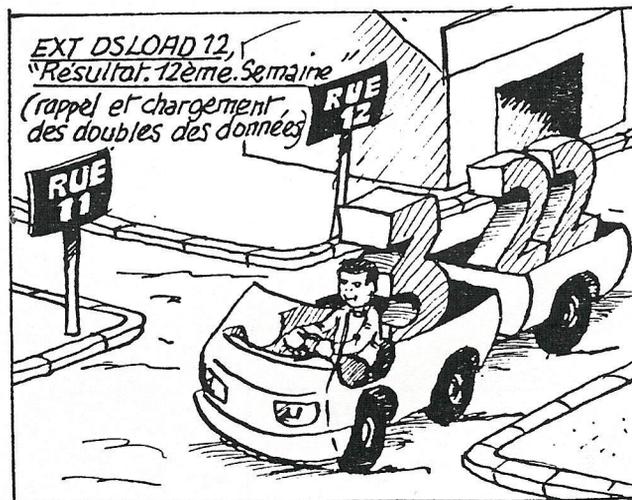
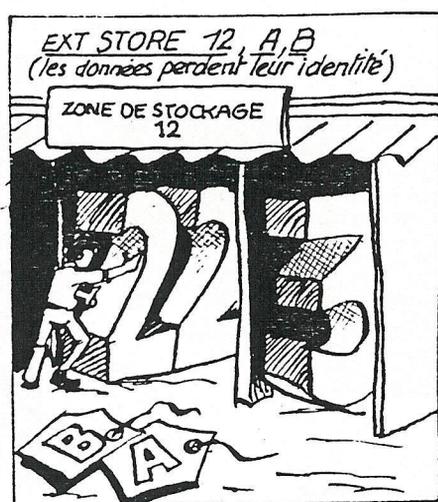
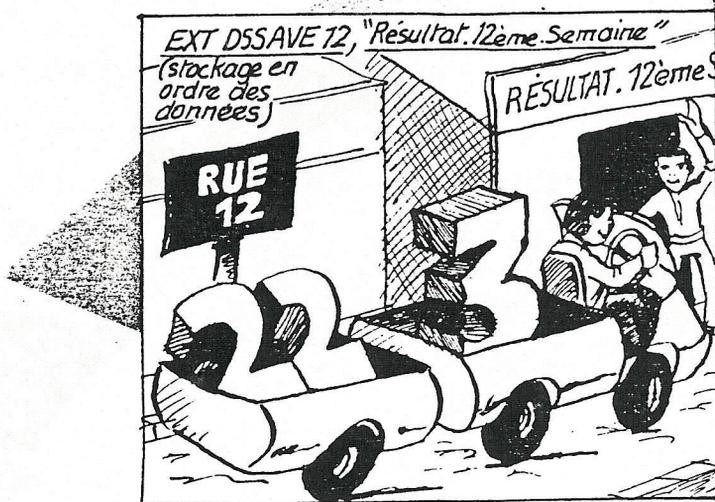
EXT STORE 12,A,B

Le nombre 12 après l'instruction est très important, puisqu'il représente l'adresse où sera stocké le résultat. Il perdra son identité (elle ne s'appellera plus A ou B) et ne pourra être retrouvé que grâce à cette nouvelle adresse.

L'instruction indique en fait au Lynx de stocker les données A et B dans l'"entrepôt" 12 (dessin 1). Cet entrepôt est en fait une adresse précise dans la mémoire de l'ordinateur. Pour pouvoir utiliser ces résultats dans un autre programme, il faut maintenant les enregistrer sur une cassette ou sur une disquette, par l'in-

struction EXT SAVE, (EXT DSAVE pour un enregistrement sur disquette), en leur donnant un nom aussi significatif que possible. Nous allons, par exemple, les sauver sous la dénomination de "résultats 12^{ème} semaine" :

EXT SAVE 12, "RÉSULTATS.12^{ème}.SEMAINE"



Les données A et B vont donc circuler par la "rue 12" pour être rangées dans un bâtiment particulier, appelé "RÉSULTATS.12^{ème}.SEMAINE" (dessin 2). Nous allons maintenant voir comment réutiliser ces données dans un autre programme. Il suffit de rappeler dans un premier temps les données du support magnétique (cassette ou disquette) et de les replacer "quelque part" dans la mémoire de l'ordinateur.

Cette opération s'effectue à l'aide d'une nouvelle instruction, EXT LOAD, (ou EXT DLOAD pour les disquettes) :
EXT LOAD 12, "RÉSULTATS.12^{ème}.SEMAINE"

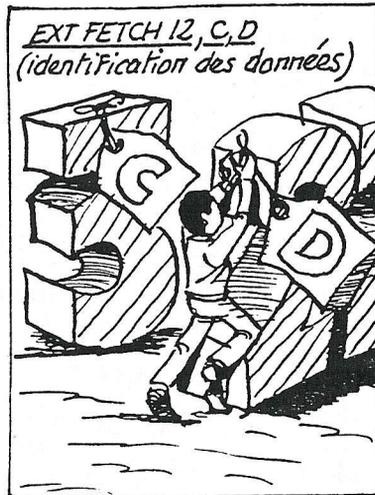
Les données reprennent donc la "rue" 12 dans l'autre sens, de la cassette vers la mémoire vive (bien évidemment, les données originales restent enregistrées sur la bande magnétique) (dessin 3).

La dernière opération à effectuer est de réintégrer ces données sous forme de variables dans un programme Basic. Cela peut se faire sous le même nom qu'au début (A et B dans notre exemple) ou sous un nom totalement différent. La commande correspondante est :

EXT FETCH 12,C,D

Nos deux données s'appellent maintenant C et D, et peuvent être réutilisées dans un autre programme Basic (dessin 4).

Ces quatre instructions permettent de manipuler facilement des variables Basic, indépendamment des programmes dans lesquelles elles figurent. Une dernière remarque : si vous désirez utiliser deux fois la même adresse (12 dans notre exemple)



pour stocker des données, il est d'abord nécessaire de "faire le ménage", c'est-à-dire d'éliminer les précédents occupants. On utilise pour cela l'instruction EXT WIPE 12, qui efface le contenu de l'adresse 12 (dessin 5). Il est aussi nécessaire d'effacer les données originales sur le support magnétique, grâce à l'instruction EXT ERASE, suivi du nom donné à ces constantes, ici "RÉSULTATS.12^{ème}.SEMAINE" (dessin 6).

Voici pour finir deux petits exemples d'utilisation manipulant plusieurs fois les mêmes variables dans un programme Basic.

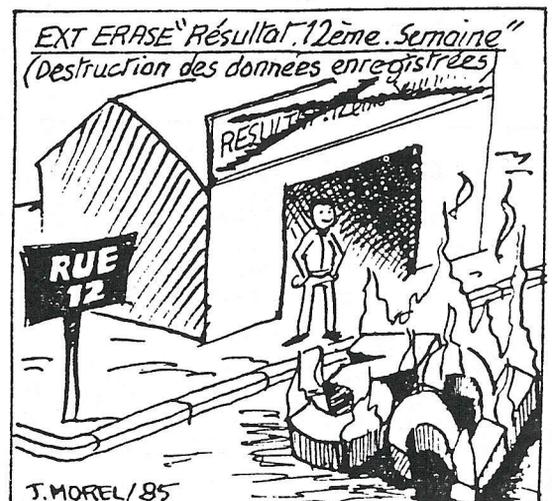
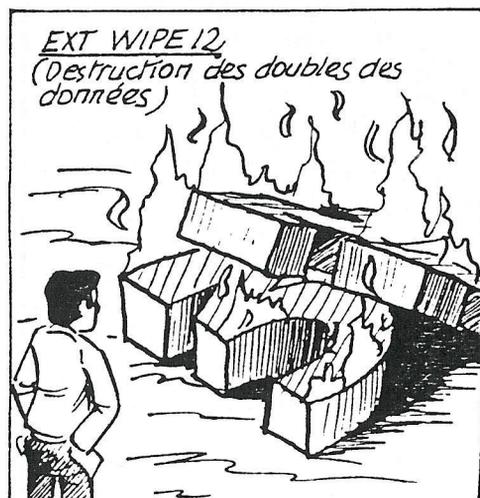
Dans un premier temps, nous allons écrire un programme utilisant les résultats de la 12^{ème} semaine, en provenance de la cassette, pour les additionner à ceux de la 13^{ème} semaine (cumul sur deux semaines) sans altérer chacun de ses résultats.

```
EXT SLOAD 12, "RÉSULTATS.12ème.SEMAINE"
100 INPUT "NOUVELLE SEMAINE:";X
110 EXT FETCH X-1,C,D
120 INPUT "NOMBRE DE POINTS:";E
130 INPUT "CLASSEMENT GÉNÉRAL:";F
140 A=C+E,B=D+F
150 EXT STORE X,A,B
EXT SSAVE 13, "RÉSULTATS.13ème.SEMAINE"
```

L'exemple suivant permet, toujours à partir des mêmes données, de réaliser un classement hebdomadaire.

```
160 EXT SLOAD 1, "CLASSEMENT. HEBDOMADAIRE"
170 DIM A(10),B(10)
180 FOR J=1 TO 10
190 EXT FETCH 1,A(J)
200 NEXT J
210 EXT WIPE 1
220 FOR J=1 TO 10
230 INPUT B(J)
240 A(J)=A(J)+B(J)
250 EXT STORE 1,A(J)
260 NEXT J
280 EXT ERASE "CLASSEMENT. HEBDOMADAIRE"
290 EXT SSAVE 1, "CLASSEMENT HEBDOMADAIRE"
230 END
```

Attention à ne pas oublier l'EXT ERASE de la ligne 280, car on ne peut sauver deux données sous le même nom.



JEUX DE MOTS

MAXI-MOTS

Qui ne connaît le célèbre jeu d'Armand Jammot, les Chiffres et les Lettres. Si votre Lynx ne peut remplacer la verve et la façon de Patrice Laffont, ni le savoir de Max Favalleli, il lui est toutefois possible de vous faire jouer à ce jeu, chez vous, grâce au programme écrit par M. Sauget

Votre Lynx est en effet capable de remplacer la charmante hôtesse chargée de placer la lettre et les chiffres choisis aléatoirement par les joueurs, mais aussi de vérifier la validité des réponses (mots et calculs) et d'assurer le suivi des scores, le contrôle des temps de jeu. Seule limite, il ne sait pas — encore — vous annoncer si un meilleur résultat aurait pu être trouvé !

Le programme de M. Sauget mêle astucieusement le Basic et le langage machine, et nécessite donc une bonne connaissance de la programmation pour être modifié. Mais tous les lynxistes pourront sans peine rentrer ce listing et le sauvegarder sur une cassette pour jouer en famille.

En effet, la présentation très claire permet à chacun de jouer, même sans connaître le fonctionnement Lynx, et sans un mode d'emploi très compliqué. En effet, le choix entre les différentes opérations se fait par un système de menu inscrit en clair sur l'écran : choix d'une voyelle ou d'une consonne, choix d'une rangée pour le tirage des chiffres. Signalons pour finir que les lettres et les chiffres proposés par la machine ne sont pas choisis totalement au hasard, mais reprennent la répartition réelle du jeu : il y a — heureusement ! — plus de chances de tirer un E qu'un Y, et la plaque 127 ne risque pas d'être tirée.

Bonne chance et amusez-vous...

```
100 VDU 4.24
110 LET A$="***MAXI-MOTS***"
120 PRINT @ 40.7:A$:
130 VDU 25.10.10.10.19
140 PRINT "MAXI-MOTS est un jeu de lettres et de chiffres. Il se joue à 2 joueurs."
150 VDU 10.10
160 LET K$=CHR$(31)
170 PRINT "1) entrez vos prénoms":K$:
180 PRINT "2) touche V pour voyelle":K$:
190 PRINT "   touche C pour consonne":K$:
:
200 PRINT "3) entrez votre nombre de lettres":K$:
210 PRINT "4) introduisez votre mot":K$:
```

```
220 PRINT "5) indiquez si il est valable (0) ou (N)":K$:
230 PRINT "6) pour les chiffres choisissez une des rangées de 1 à 4":K$:
240 PRINT "7) entrez votre nombre":K$:
250 PRINT "8) introduisez votre calcul":K$:
260 PRINT "9) indiquez si il est valable (0) ou (N)":K$:
270 PRINT @ 42.220:"BONNE CHANCE"
280 LET w=HIMEM-40592
290 RESERVE HIMEM-(w+146)
300 FOR I=HIMEM TO HIMEM+126
310   READ A
320   POKE I,A
330 NEXT I
340 CODE 01 06 00 21 8A 9E 11 89 9E 1A
ED B1 28 39 13 2E 8B 01 05 00 1A ED B1 2
8 2E 13 2E 8C 01 04 00 1A ED B1 28 23 13
2E 8D 01 03 00 1A ED B1 28 18 13 2E 8E
01 02 00 1A ED B1 28 0D 13 2B 03 1A ED B
1 28 05 AF 32 8F 9E C9 3E 01 32 8F 9E C9
350 DATA 69.65.73.69.79.69.65.85.69.65
360 DATA 73.85.69.79.69.73.65.65.69.85
370 DATA 79.73.65.69.69.85.73.73.65.89
380 DATA 69.65.69.79.69.79.85.73.65.69
390 DATA 85.69.79.73.69
400 DATA 82.68.70.61.83.76.84.77.71.78
410 DATA 86.74.83.66.76.72.80.82.83.84
420 DATA 83.68.67.70.75.76.76.82.87.77
430 DATA 78.84.86.83.67.80.72.84.71.68
440 DATA 66.82.70.76.84.76.77.78.78.83
450 DATA 78.81.83.90.82.84.88.82
460 DATA 3.9.5.25.2.8.100.4.10.9.6.50.7.
8.1.2.10.5.75.1.3.6.4.7
470 DATA 4.126.7.125.16.127.27.128.38.12
9.49.130
480 DATA 4.132.7.131.16.133.27.134.38.13
5.49.136
490 DATA 4.138.7.137.16.139.27.140.38.14
1.49.142
500 LET F$=""
510 DIM E$(21)(9)
520 LET E$(1)="CONSONNE OU VOYELLE ?"
530 LET E$(2)="COMBIEN DE LETTRES ?"
540 LET E$(3)="ENTREZ VOTRE MOT"
550 LET E$(4)="LE MOT EST-IL VALABLE"
560 LET E$(5)="No DE RANGEE (1 à 4)"
570 LET E$(6)="ENTREZ VOTRE NOMBRE"
580 LET E$(7)="ENTREZ VOTRE CALCUL"
590 LET E$(8)="LE COMPTE EST-IL BON"
600 LET E$(9)="
610 PRINT @ 10.220:"PUSSEZ UNE TOUCHE
POUR CONTINUER"
620 LET A=GETN
630 VDU 4.24.10.10.10
640 INPUT "JOUEUR 1 VOTRE PRENOM ?":B$
650 INPUT "JOUEUR 2 VOTRE PRENOM ?":C$
660 PAPER 1
670 PROTECT 0
680 VDU 4.1.2.2.0
690 LET A=1.B=0.C=0.F=1.G=2.L=45
700 PROC INIT
710 INK 4
720 PROTECT 1
740 PROC REINI
750 PROC DECI
```

PAR M. SAUGET

```
760 FOR J=1 TO 2
770 PROC TIRAGE
780 PROC CV
790 PROC TEMPS
800 PROC RESU
810 PROC SCORE
820 IF J=1 THEN RESTORE (470)
830 PROC REINI
840 VDU 24
850 PROC INIT
860 NEXT J
870 PROC CHIFF
880 PROC TEMPS
890 PROC RECHIFF
900 PROC SCORE
910 PROC DECI
920 PROC INIT
930 GOTO 760
940 DEFPROC DECI
950 IF F=1 THEN LET D#=B#
960 IF G=1 THEN LET D#=C#
970 ENDFPROC
980 DEFPROC TIRAGE
990 FOR I=1 TO 6
1000 READ I.O
1010 POKE LCTN(340)+I.O
1020 NEXT I
1030 LET X=40573
1040 FOR I=0 TO 5
1050 RANDOM
1060 LET V=INT(RND*44)+1
1070 POKE X+I,V
1080 NEXT I
1090 CALL LCTN(340)
1100 IF PEEK(40591)=1 THEN GOTO 1030
1110 FOR I=1 TO 6
1120 READ I.O
1130 POKE LCTN(340)+I.O
1140 NEXT I
1150 LET X=40579
1160 FOR I=0 TO 5
1170 RANDOM
1180 LET V=INT(RND*57)+1
1190 POKE X+I,V
1200 NEXT I
1210 CALL LCTN(340)
1220 IF PEEK(40591)=1 THEN GOTO 1150
1230 FOR I=1 TO 6
1240 READ I.O
1250 POKE LCTN(340)+I.O
1260 NEXT I
1270 LET X=40585
1280 FOR I=0 TO 5
1290 RANDOM
1300 LET V=INT(RND*23)+1
1310 POKE X+I,V
1320 NEXT I
1330 CALL LCTN(340)
1340 IF PEEK(40591)=1 THEN GOTO 1270
1350 RESTORE (470)
1360 ENDFPROC
1370 DEFPROC CV
1380 PRINT @ 50,210:E$(1)
1390 FOR I=1 TO 9
1400 PRINT @ 10,210:D$
1410 VDU 24
1420 LET Q#=GET#
1430 IF NOTQ#="C" AND NOTQ#="V" THEN
GOTO 1420
1440 IF Q#="C" THEN GOTO 1490
1450 LET R=40445+(PEEK(h))
1460 PRINT @ D.60:CHR$(PEEK(R))
1470 LET h=h+1,D=D+10
1480 GOTO 1520
1490 LET R=40490+(PEEK(k))
1500 PRINT @ D.60:CHR$(PEEK(R))
1510 LET k=k+1,D=D+10
1520 SWAP F,G
1530 PROC DECI
1540 VDU 25
1550 PRINT @ 10,210:F#
1560 NEXT I
1570 SWAP F,G
1580 PROC DECI
1590 LET D=20
1600 PRINT @ 50,210:E$(9)
1610 ENDFPROC
1620 DEFPROC TEMPS
1630 BEEP 250,150.17
1640 FOR I=L TO 0 STEP -1
1650 IF I=9 THEN PRINT @ 103,55:" "
1660 PRINT @ 103,55:I
1670 PAUSE 3500
1680 NEXT I
1690 BEEP 400,300.17
1700 ENDFPROC
1710 DEFPROC RESU
1720 PRINT @ 10,210:D$: @ 50,210:E$(2)
1730 LET X=GETN-48
1740 SWAP F,G
1750 PROC DECI
1760 PRINT @ 10,210:F$: @ 10,210:D#
1770 LET Z=GETN-48
1780 IF X>Z THEN SWAP F,G
1790 IF Z>X THEN LET M=1
1800 IF Z>X THEN SWAP Z,X
1810 LET Z=0,V=X
1820 PROC DECI
1830 PRINT @ 10,210:F$: @ 10,210:D$: @
50,210:E$(3)
1840 VDU 24
1850 FOR I=1 TO V
1860 LET U#=GET#
1870 PRINT @ E.90:U#
1880 LET E=E+10
1890 NEXT I
1900 VDU 25
1910 PRINT @ 10,210:E$(4)
1920 LET V#=GET#
1930 IF V#="0" THEN ENDFPROC
1940 IF M=1 THEN LET n=1
1950 LET M=1
1960 SWAP F,G
1970 PROC DECI
1980 ENDFPROC
1990 DEFPROC SCORE
2000 IF D#=B# THEN LET B=B+X
2010 IF D#=C# THEN LET C=C+X
2020 IF M<>1 OR n=1 THEN SWAP F,G
2030 PROC DECI
2040 LET M=0,A=A+1
2050 VDU 4
2060 PRINT @ 40,40:B: @ 40,55:C: @ 83,5
5:A: @ 103,55:L
```

JEUX DE MOTS

```

2070 ENDPROC
2080 DEFPROC CHIFF
2090 PRINT @ 50.210:E$(5)
2100 FOR I=1 TO 6
2110 PRINT @ 10.210:D$
2120 LET V$=GET$
2130 PRINT @ 55.200:V$: @ 10.210:F$
2140 SWAP F,G
2150 PROC DECI
2160 NEXT I
2170 VDU 24
2180 FOR I=1 TO 6
2190 LET R=40549+(PEEK(a))
2200 PRINT @ D,60:PEEK(R)
2210 LET a=a+1,D=D+15
2220 NEXT I
2230 LET T=INT(RND*899)+100
2240 PRINT @ 56.40:T
2250 VDU 25
2260 PRINT @ 50.210:E$(9): @ 55.200:" "
2270 ENDPROC
2280 DEFPROC RECHIFF
2290 VDU 25
2300 PRINT @ 10.210:D$: @ 50.210:E$(6)
2310 INPUT P
2320 PRINT @ 10.150:P:" ";D$
2330 SWAP F,G
2340 PROC DECI
2350 PRINT @ 10.210:F$: @ 10.210:D$
2360 INPUT S
2370 PRINT @ 10.170:S:" ";D$
2380 LET X=ABS(T-P),Z=ABS(T-S)
2390 IF X<Z THEN SWAP F,G
2400 IF Z<X THEN LET M=1
2410 IF X=0 OR Z=0 THEN LET N=8
2420 PROC DECI
2430 PRINT @ 10.210:F$: @ 10.210:D$: @
50.210:E$(7)
2440 INPUT P
2450 VDU 24
2460 PRINT @ 56.90:P
2470 VDU 25
2480 PRINT @ 50.210:E$(8)
2490 LET U$=GET$
2500 IF U$="0" THEN LET X=N
2510 IF U$="0" THEN ENDPROC
2520 IF M=1 THEN LET n=1
2530 SWAP F,G
2540 PROC DECI
2550 LET M=1,X=N
2560 ENDPROC
2570 DEFPROC INIT
2580 VDU 24
2590 PRINT @ 40.7:A$:
2600 VDU 25
2610 LET D=20,E=D,M=0,N=6,n=0,h=40573,k=
n+s,a=h-12
2620 PRINT @ 5.40:B$: @ 5.55:C$: @ 75.4
0:"TIRAGE": @ 100.40:"TEMPS": @ 51.40:"F
TS": @ 51.55:"PTS":
2630 ENDPROC
2640 DEFPROC REINI
2650 VDU 25
2660 PRINT @ 40.40:B: @ 40.55:C: @ 83.5
5:A: @ 103.55:L
2670 ENDPROC

```

La programmation des jeux de lettres (pendu, saint-ex, master-mind, lettres, mots croisés, mot le plus long, etc.) fait partie d'un des passe-temps favoris de nombreux hobbystes micro, et en particulier des Lynxistes, pour deux raisons. D'une part, il est possible de réaliser des applications intéressantes, avec un look professionnel, sans avoir recours au langage machine, et d'autre part, les instructions de manipulations de chaînes de caractères sont riches en astuces, pour parvenir aux résultats désirés.

Pour les Lynxistes de fraîche date, voici recensés quelques trucs classiques, mais pas toujours faciles à trouver tout seul.

– Le tirage au sort de lettres ou de mots est à la base de nombreux jeux. Deux solutions peuvent être efficaces : soit tirer les lettres aléatoirement à partir de leur code ASCII, soit dans un tableau DATA. Voici deux exemples qui donnent une lettre majuscule au hasard :

```

10 A=RND(1)*27+65
20 A$=CHR$(A)

```

S	M	T	O	M	I	A	X
---	---	---	---	---	---	---	---

```

10 DATA BONJOUR, BONSOIR, SALUT
20 A=RND(1)*3
30 FOR I=1 TO A
35 DIM A$(3)
40 READ A$(I)
50 NEXT I
60 NOM$=A$(I)

```

Les manipulations de chaîne de caractères portent sur des variables suivies du symbole \$. Or ce type de variables chaînes peuvent contenir n'importe quelle suite de caractère, aussi incohérente soit-elle (même "\$(\$,\$)"%). Pour donner un aspect fini à vos logiciels, il est conseillé de tester les INPUT des utilisateurs. Plutôt que de tester lettre à lettre, il est conseillé de transcrire les chaînes de caractères en code ASCII, et à vérifier que les valeurs sont comprises entre 65 (A) et 90 (Z).

– L'opération peut être faite en sens inverse, c'est-à-dire qu'il est souvent conseillé de saisir un nombre sous la forme d'une chaîne de caractères, et de le transformer ensuite grâce à l'instruction VAL.

MOTS CROISÉS

Ce deuxième jeu n'est pas moins célèbre que le précédent, puisqu'il s'agit tout simplement de l'adaptation micro-informatique des classiques mots croisés ! Ce jeu est en fait à deux niveaux, ce qui présente tout son intérêt. Dans un premier temps, il vous propose de remplir une grille classique de 9 x 9 cases, en affichant sur demande la définition du mot recherché.

Attention, toutes ne sont pas si évidentes que ça, en particulier le 4 horizontal : "A craindre avec une mauvaise conduite", qui s'applique plus à la plomberie qu'aux nouveaux systèmes éducatifs...

Mais ce programme est particulièrement intéressant pour les débutants, puisqu'il est entièrement écrit en Basic Lynx, et très structuré. Il est donc très facile de le modifier, et en particulier de créer ses propres grilles, en gardant le squelette de programme. A cette fin, toutes les définitions et les mots à trouver sont regroupés à la fin du programme, dans les lignes de DATA (2770 à 3070).

```

100 CLS
110 PROTECT 0
120 INK 6
130 CLS
140 DIM M$(10)(31),D$(61)(31),P$(1)(31),
P(31),F(9*9),D(31)
150 LET I=0,D=0,C=0,R$="",A$="",F=0,L=31
.N=0,S=0,J=0,T=0,X=0
160 FOR J=1 TO 31
170 LET D(J)=TRUE
180 READ P$(J),P(J),M$(J),D$(J)
190 NEXT J
200 FOR J=0 TO 81
210 LET F(J)=32
220 NEXT J
230 GOSUB 570 PROC INT
240 GOSUB 1080 PROC CHOIX
250 GOSUB 1140 PROC DESSIN
260 GOSUB 1590 PROC DEFINITION
270 GOSUB 2060 PROC MOT
280 IF D=1 AND NOT A$=M$(S) THEN INK 2
290 ELSE GOTO 340
300 PRINT @ 91.100:"FAUX":
310 PAUSE 15000
320 INK 6
330 GOTO 260
340 IF D=1 AND A$=M$(S) THEN LET L=L-1
350 ELSE GOTO 380
360 GOSUB 2230 PROC AFFICHAGE
370 GOTO 470
380 LET L=L-1
390 GOSUB 2230
400 IF NOT A$=M$(S) THEN LET F=F+1,D(S)=
FALSE

```

```

410 IF A$=M$(S) AND NOT D(S) THEN LET D(
S)=TRUE,F=F-1
420 IF F<0 THEN LET F=0
430 IF L>16 THEN GOTO 470
440 PRINT @ 5.210:"IMPRESSION DU NOMBRE
DE FAUTES ";R$
450 LET R$=GET$
460 IF R$="0" THEN GOSUB 2530 PROC FAUX
470 IF L>12 THEN GOTO 260
480 GOSUB 2580 PROC VERIF
490 IF X<>83 THEN GOTO 260
500 CLS
510 INK CYAN
520 PRINT @ 20.100:"VOUS AVEZ REUSSI EN
":T;" COUPS":
530 PRINT @ 20.220:
540 INPUT "VOULEZ VOUS REJOUER ";R$
550 IF R$="0" THEN RUN
560 END
570 CLS DEFPROC INT
580 INK 5
590 PRINT TAB (15):"MOTS CROISES":CHR$(
31):
600 PRINT :CHR$(31):
610 PRINT TAB (12):"B.L. 1984":CH
R$(31):
620 PRINT CHR$(31):
630 INK 6
640 PAPER 0
650 PRINT "Le curseur marque la place du
mot sur la grille et la definition appar
ait en bas de celle-ci."
660 PRINT
670 PRINT "Pour faire defiler les de
finitions appuyez sur la ":
680 INK 2
690 VDU 18
700 PRINT "barre d'espacement.":CHR$(31)
:
710 VDU 18
720 INK MAGENTA
730 PRINT "Si vous choisissez une defini
tion."
740 INK 2
750 VDU 18
760 PRINT "Appuyez sur return":
770 VDU 18
780 INK 6
790 PRINT " pour selectionner votre c
hoix . puis valider votre mot en ":
800 INK 2
810 VDU 18
820 PRINT "appuyant sur return.":CHR$(31)
):
830 VDU 18
840 INK 6
850 PRINT "Pour changer d'option en cour
s de partie":
860 INK 2
870 VDU 18
880 PRINT "appuyez sur la touche 0.":CHR
$(31):
890 VDU 18
900 PRINT
910 VDU 18
920 PRINT "TOUCHE 1":
930 VDU 18

```

JEUX DE MOTS

```

940 INK 4
950 PRINT " : Avec verification du mot":
CHR$(31);
960 PRINT
970 INK 2
980 VDU 18
990 PRINT "TOUCHE 2":
1000 VDU 18
1010 INK 4
1020 PRINT " : Avec affichage sur demand
e
a mi-partie du nombre de
mots
faux."
1030 INK 7
1040 PRINT
1050 PRINT " CHOISISSEZ UNE OPTION (1
/2) ?":
1060 RETURN
1070 REM -----
1080 REM CHOIX OPTIONS DEFPROC CHOIX
1090 LET R#=KEY$
1100 IF ASC(R#)<>49 AND ASC(R#)<>50 THEN
GOTO 1090
1110 LET O=VAL(R#)
1120 RETURN
1130 REM -----
1140 REM DEBUT DESSIN DEFPROC DESSIN
1150 CLS
1160 INK 3
1170 PRINT @ 90.20:"OPTION : ":O:
1180 INK 5
1190 PRINT @ 88.50:"MOTS CROISES":
1200 PRINT @ 88.60:"B.L. ":
1210 PRINT @ 97.70:"1984":
1220 INK 4
1230 FOR J=11 TO 155 STEP 16
1240 MOVE J,162
1250 PLOT 3,0.-144
1260 NEXT J
1270 FOR J=18 TO 162 STEP 16
1280 MOVE 11,J
1290 PLOT 3,144,0
1300 NEXT J
1310 LET K=1
1320 FOR J=8 TO 75 STEP 8
1330 INK RED
1340 PRINT @ J,5:K:
1350 LET K=K+1
1360 NEXT J
1370 LET K=1
1380 FOR J=25 TO 153 STEP 16
1390 INK BLUE
1400 PRINT @ 0,J:K:
1410 LET K=K+1
1420 NEXT J
1430 INK CYAN
1440 PROC CARRE(108,122,35)
1450 PROC CARRE(60,74,51)
1460 PROC CARRE(44,58,67)
1470 PROC CARRE(140,154,67)
1480 PROC CARRE(76,90,83)
1490 PROC CARRE(124,138,83)
1500 PROC CARRE(28,42,99)
1510 PROC CARRE(108,122,115)
1520 PROC CARRE(28,42,131)
1530 PROC CARRE(76,90,131)
1540 PROC CARRE(92,106,131)
1550 PROC CARRE(44,58,147)

1560 INK 6
1570 RETURN
1580 REM -----
1590 REM CHOIX DE LA DEFINITION DEFPROC DEFINITION
1600 IF S=31 THEN LET S=0
1610 PRINT @ 91.100:" ": @ 5.210:"
":
1620 LET J=S+1
1630 INK RED
1640 IF ASC(P$(J))>58 THEN INK 1
1650 PRINT @ 5.190:LEFT$(D$(J),7):
1660 INK 6
1670 PRINT MID$(D$(J),8,50):CHR$(31):
1680 IF P$(J)>"9" THEN GOTO 1730
1690 REM
1700 INK 2
1710 PRINT @ 8*VAL(P$(J)).(16*P$(J))+8:C
HR$(32):CHR$(22):CHR$(126):
1720 GOTO 1760
1730 REM
1740 INK 1
1750 PRINT @ 8*(P$(J)).(16*((ASC(P$(J)))
-64))+8:CHR$(32):CHR$(22):CHR$(123):
1760 LET R#=GET$
1770 LET C=ASC(R#)
1780 IF C=48 THEN IF O=1 THEN LET O=2
1790 ELSE GOTO 1830
1800 INK 3
1810 PRINT @ 117.20:O:
1820 GOTO 260
1830 IF C=48 THEN IF O=2 THEN LET O=1
1840 ELSE GOTO 1880
1850 INK 3
1860 PRINT @ 117.20:O:
1870 GOTO 260
1880 IF P$(J)>"9" THEN INK YELLOW
1890 ELSE GOTO 1920
1900 PRINT @ 8*(P$(J)).(16*((ASC(P$(J)))
-64))+8:CHR$(F(P$(J)*9-((ASC(P$(J))-64)
)):
1910 GOTO 1940
1920 INK YELLOW
1930 PRINT @ 8*VAL(P$(J)).(16*P$(J))+8:C
HR$(F(VAL(P$(J))*9-(P$(J))):
1940 IF C=13 THEN LET S=J
1950 ELSE GOTO 1980
1960 LET C=J
1970 GOTO 2040
1980 IF C<>32 THEN GOTO 1630
1990 LET J=J+1
2000 IF J>31 THEN LET J=1
2010 PRINT @ 80.120:" ":
2020 GOTO 1630
2030 REM
2040 RETURN
2050 REM -----
2060 REM ENTREE DU MOT DEFPROC MOT
2070 INK RED
2080 IF S>15 THEN INK BLUE
2090 LET T=T+1
2100 PRINT @ 41.210:"
":
2110 PRINT @ 5.210:
2120 INPUT "VOTRE MOT :":A$
2130 IF A$="" THEN RETURN
2140 IF LEN(A$)>9 OR LEN(A$)<>LEN(M$(S))
THEN INK RED

```

```

2150 ELSE GOTO 2200
2160 PRINT @ 91.100;"FAUX":
2170 PAUSE 15000
2180 PRINT @ 91.100;"  ":
2190 GOTO 2070
2200 REM
2210 RETURN
2220 REM -----
-
                DEFPROC AFFICHAGE
2230 REM SOUS-PROGRAMME D'AFFICHAGE
2240 INK YELLOW
2250 IF A$="" THEN INK 6
2260 ELSE GOTO 2280
2270 RETURN
2280 LET K=1
2290 IF C>15 THEN GOTO 2400
2300 FOR I=0 TO LEN(A$)-1
2310 LET F(VAL(P$(S))*9-(P(S)+I))=ASC(
MID$(A$,I+1,1))
2320 NEXT I
2330 LET J=0
2340 PRINT @ 8*VAL(P$(S)).J*8+(16*(P(S)
)+8):MID$(A$,K,1):
2350 LET K=K+1
2360 IF K>LEN(A$) THEN GOTO 2500
2370 LET J=J+2
2380 IF J>20 THEN GOTO 2500
2390 GOTO 2340
2400 LET J=0
2410 FOR I=0 TO LEN(A$)-1
2420 LET F((P(S)+I)*9-((ASC(P$(S)))-64
))=ASC(MID$(A$,I+1,1))
2430 NEXT I
2440 PRINT @ J*4+(8*(P(S))).(16*((ASC(P
$(S))-64))+8):MID$(A$,K,1):
2450 LET K=K+1
2460 IF K>LEN(A$) THEN GOTO 2500
2470 LET J=J+2
2480 IF J>20 THEN GOTO 2500
2490 GOTO 2440
2500 INK 6
2510 RETURN
2520 REM -----
                DEFPROC FAUX
2530 REM AFFICHAGE DU NOMBRE DE MOTS FAU
X
2540 INK RED
2550 PRINT @ 80.120:F;" MOTS FAUX ":
2560 RETURN
2570 REM -----
2580 REM VERIFICATION DE LA GRILLE A PAR
TIR DU 18eme MOT ENTRE DEFPROC VERIF
2590 LET X=0
2600 FOR W=1 TO 15
2610 FOR I=0 TO 8
2620 IF F(VAL(P$(W))*9-(P(W)+I))=ASC
(MID$(M$(W),I+1,1)) THEN LET X=X+1
2630 NEXT I
2640 NEXT W
2650 FOR W=16 TO 31
2660 FOR I=0 TO 8
2670 IF F((P(W)*9)+I-((ASC(P$(W)))-6
4))=ASC(MID$(M$(W),I+1,1)) THEN LET X=X
+1
2680 NEXT I
2690 NEXT W
2700 RETURN

```

```

2710 DEFPROC CARRE(x,v,z)
2720 FOR J=x TO v
2730 MOVE J,z
2740 PLOT 3.0.14
2750 NEXT J
2760 ENDPROC
2770 DATA 1.1,APPAREILS.1 VER : INSTRUME
NTS
2780 DATA 2.1,CELLE.2 VER : DEMONSTRATIF
2790 DATA 3.1,TRI.3 VER : CHOIX
2800 DATA 3.5,EMOI.3 VER : TROUBLE
2810 DATA 4.1,II.4 VER : ROMAINS
2820 DATA 4.4,FRANCE.4 VER : HEXAGONE
2830 DATA 5.1,VETU.5 VER : HABILLE
2840 DATA 5.6,CE.5 VER : DEMONSTRATIF
2850 DATA 6.1,IRRITAS.6 VER : EXASPERRAS
2860 DATA 7.3,OTER.7 VER : ENLEVER
2870 DATA 7.8,BE.7 VER : SYMBOLE DU GLIC
INUM
2880 DATA 8.1,EMIE.8 VER : EMIETTE
2890 DATA 8.6,OREE.8 VER : BOUT DE BOIS
2900 DATA 9.1,SAS.9 VER : CRIBLE
2910 DATA 9.5,ONCES.9 VER : PETITS POIDS
2920 DATA A.1,ACTIVITES.1 HOR : OCCUPATI
ONS
2930 DATA B.1,PERIER.2 HOR : ACTEUR FRAN
CAIS
2940 DATA B.8,MA.2 HOR : POSSESSIF
2950 DATA C.1,PLI.3 HOR : LETTRE
2960 DATA C.5,TROIS.3 HOR : REGLE DE..
2970 DATA D.1,AL.4 HOR : ALUMINIUM
2980 DATA D.4,FUITE.4 HOR : A CRAINDRE A
VEC UNE MAUVAISE CONDUITE
2990 DATA E.1,REER.5 HOR : RAIRE
3000 DATA E.6,TE.5 HOR : REGLE
3010 DATA F.3,MACARON.6 HOR : PATISSERIE
3020 DATA G.1,ICONES.7 HOR : IMAGES
3030 DATA G.8,RC.7 HOR : RESPONSABILTE C
IVILE
3040 DATA H.3,IC.8 HOR : CIRCUIT INTEGRE
AMERICAIN
3050 DATA H.7,BEE.8 HOR : GRANDE OUVERTE
3060 DATA I.1,SI.9 HOR : ADVERBE
3070 DATA I.4,EPIEES.9 HOR : OBSERVEES

```



SKETCH PAD

Le petit programme suivant fait apparaître un point à l'écran, qui peut être déplacé lentement ou rapidement sur l'écran, et utilisé pour réaliser des dessins. Les flèches de déplacement du curseur, utilisées seules ou en combinaison, peuvent déplacer le point dans toutes les directions. Appuyées en même temps que la barre d'espace, elles permettent un déplacement plus rapide. La touche RETURN trace une ligne entre les deux positions, et la touche C efface l'écran.

```

100 LET X=100, Y=100, V=0
120 TEXT
130 REPEAT
140 PROC KEYREAD
150 UNTIL 0
160 END
170 REM *****
2000 DEFPROC KEYREAD
2005 IF KEYS="C" OR "C" THEN CLS
2010 IF NOT(INP(&0480) BNAND 8)
D=D*1.2 THEN LET
2015 ELSE LET D=1
2020 IF NOT(INP(&0080) BNAND 16)
PROC MOVE (0,-1)
2030 IF NOT(INP(&0080)BNAND 32)
THEN
PROC MOVE (0,1)
2040 IF NOT(INP(&0980) BNAND 32)
THEN
PROC MOVE (1,0)
2050 IF NOT(INP(&0980) BNAND 4)
THEN
PROC MOVE (-1,0)
2060 IF INP(&0980) BNAND 8 THEN
ENDPROC
2070 INK GREEN
2080 DOT X,Y
2090 LET V=1
2095 ENDPROC
    
```

```

2099 REM *****
2100 DEFPROC MOVE(B,C)
2110 INK GREEN*V
2120 DOT X,Y
2130 LET X=(X+B*D) MOD 256
2140 LET Y=(Y+C*D) MOD 256
2150 PROC READDOT
2160 INK GREEN*V+4
2170 DOT X,Y
2180 ENDPROC
2199 REM *****
2200 DEFPROC READDOT
2210 CALL &0070,&C000+Y*32+X
DIV 8
2220 LET A=2**(7-X MOD 8)
2230 LET V+SGN(HL BNAND A)
2240 ENDPROC
2250 END
    
```

SORTEZ DE VOTRE RÉSERVE

L'utilisation de l'instruction RESERVE (pour réserver un espace mémoire suffisant pour redéfinir un jeu de caractères) pose parfois quelques problèmes lorsqu'on l'utilise comme habitude, sous la forme :

```
10 RESERVE HIMEM -100
```

en début de programme.

En effet, on n'est jamais sûr, lorsque le programme effectue plusieurs RUN successifs, que l'espace mémoire restant sera suffisant. Il est alors plus prudent d'écrire l'instruction sous la forme.

```
10 RESERVE HIMEM 40852
```

Cependant, le programme peut encore se planter si un espace mémoire plus important a déjà été réservé par un autre programme. La forme la plus sécurisante est alors :

```
10 IF HIMEM>40852 THEN
RESERVE 40852
```

VERT ALTERNATIF

Le Lynx possède quatre banques de couleurs, les standards rouge, vert, bleu, et une quatrième appelée "vert alternatif". Ce vert alternatif peut être utilisé sur des écrans séparés, pour afficher du texte, des graphismes, ou même réaliser des animations rapides.

Le Lynx Basic ne possède pas d'instruction permettant d'afficher le vert alternatif sur l'écran. Il est donc nécessaire de changer la position dans la mémoire, ou le Basic "envoie" les données à la banque verte, pour les détourner sur la banque verte alternative.

Cette opération est réalisable simplement grâce à l'instruction suivante :

```
DPOKE &6292,&8000 (Aφφφ)
```

le retour à la normale pouvant alors être obtenu par :

```
DPOKE &6292,&C000
```

Ces instructions n'affichent cependant pas le vert alternatif sur l'écran, ce qui s'obtient par l'instruction

```
OUT &80,16
```

Le programme Explosion illustre l'utilisation de cette nouvelle couleur, en affichant une explosion verte dans chacune des couleurs primaires, à l'intérieur d'une boucle REPEAT UNTIL, donnant l'effet d'un flash surprenant.

```

100 REM *****
110 TEXT
120 DPOKE &6292,&A000
130 CLS
140 PROTECT BLACK
150 PROC LINES(100,RED)
160 PROC LINES(60,YELLOW)
170 DPOKE &6292,&C000
180 PROCLINES(30,WHITE)
190 REPEAT
200 PAUSE ; 200
210 OUT &0080,4
220 PAUSE 250
230 BEEP 1,5,63
240 PAUSE 100
250 OUT &0080,16
260 UNTIL KEYN
270 END
280 REM *****
1000 DEFPROC LINES(X,I)
1010 INK I
1020 PROTECT WHITE-INK
1030 FOR Y=0 TO X
1040 MOVE 127,126
1050 PLOT 3,RAND(X)-X DIV 2,
RAND(X)-X DIV 2
1060 NEXT Y
1070 ENDPROC
    
```

ROLL OVER BEETHOVEN

Cette table donne les équivalences entre les notes de la gamme chromatique sur plus de quatre octaves et les valeurs à donner dans les instructions musicales du Basic Lynx, fréquence et enveloppe. Rappelons aux mélomanes que le A anglais correspond au LA français.

NOTE	FREQ	W/LEN
A	110	909
A#	116.6	857.3
B	123.5	809.7
C	130.8	764.5
C#	138.6	721.5
D	146.8	681.2
D#	155.6	642.7
E	164.8	606.8
F	174.6	572.7
F#	185	540.5
G	196	510.2
G#	207.7	481.5
A	220	454.5
A#	233.3	428.6
B	246.9	405
MID C	261.1	383
C	277.2	360.7
D	293.7	340.5
D#	311.1	321.5
E	329.6	303.4
F	349.2	286.4
F#	370	270.3
G	392	255
G#	415.3	240.8
A	440	227.3
A#	466.2	214.5
B	493.9	202.5
C	523.2	191
C#	554.4	180.3
D	587.3	170.3
D#	622.3	160.7
E	659.3	151.7
F	698.5	143.2
F#	740	135
G	784	127.5
G#	830.6	120.4
A	880	113.6
A#	932.3	107.3
B	987.8	101.3
C	1046.5	95.5
C#	1108.7	90.2
D	1174.7	85.3
D#	1244.5	80.3
E	1318.5	75.8
F	1396.9	71.6
F#	1480	67.5
G	1568	63.8
G#	1661.2	60.2
A	1760	56.8
A#	1864.6	53.6
B	1975.5	50.6
C	2093	47.8
C#	2217.5	45
D	2349.3	42.6
D#	2489	40.2
E	2637	37.9
F	2793.9	35.8
F#	2959.6	33.8
G	3135.9	31.9
G#	3322.4	30

ÉPARGNER LA MÉMOIRE

Lorsque l'on programme en Basic, on n'utilise pas, pour des raisons de clarté, toutes les possibilités de compression de lignes. Lors de la mise au point du programme, cependant, il est possible d'économiser beaucoup de place en mémoire. Les exemples suivants illustrent ce principe :

Le programme suivant utilise trop de lignes :

```
10 LET X=100
20 LET Y=200
30 VDU 24
40 VDU 7
50 PRINT A;
60 PRINT "=VENTES"
```

et peut être largement compressé, économisant 8 octets pour chaque ligne supprimée :

```
10 LET X=100, Y=200
20 VDU 24,7
30 PRINT A;"=VENTES"
```

De même, ce type de traduction d'une chaîne de caractère en valeur numérique :

```
10 LET A$=GET$
20 LET =VAL(A$)
30 IF A=0 THEN GOTO 10
```

peut avantageusement être remplacé par :

```
10 LET A=GETN-48
20 IF A<0 OR A>9 THEN GOTO 10
```

POKE ET PEEK SANS COLEGRAM

Sans robot pour les présenter, voici une liste de POKE aux résultats surprenants qui permettront aux Lynxistes de se transformer sans difficultés en véritables magiciens du clavier. :

POKE 25268,0 supprime l'affichage du listing d'un programme, comme DPOKE&62B3, &62B2

DPOKE 25239,2 provoque la répétition des caractères.

Voici pour finir la liste des adresses permettant de redéfinir les fonctions pré-programmées du clavier du Lynx. Il suffit en effet de taper :

POKE (adresse de la touche), code

Les adresses des touches vont de 25146 (pour la lettre A) jusqu'à 25171 pour la lettre Z. A titre d'exemple, voici quelques-uns des codes fonctions des instructions du Basic Lynx :

```
1 VDU
6 INK
7 PAPER
```

```
33 PLOT
19 CLS
63 RENUM
12 POKE
etc.
```

VITESSE SUPÉRIEURE

Lorsque l'on donne trop de travail au Lynx, la vitesse de travail est réduite d'autant. C'est donc une idée intéressante que de prendre certaines précautions lorsque l'on utilise les instructions de gestion d'écran. Voici une liste - non exhaustive - de quelques trucs simples, mais efficaces :

- PROTECT toutes les couleurs sauf celle utilisée.

- Faire suivre tous les PRINT DE "," évite à l'ordinateur de lire tous les caractères blanc avant de passer à la ligne suivante.

- CHR\$(31) est un substitut intéressant à un PRINT seul.

- Utiliser l'instruction TAB plutôt que des espaces entre guillemets pour positionner le texte.

ANNONCES

Cherche possesseur de Lynx non loin de mon domicile (banlieue nord-est) pour échange d'idées, de trucs et d'astuces.

Jacques MOREL
48, avenue Chappée
93320 Tremblay les Gonesses
861.49.43

Création d'un club d'utilisateurs de Lynx 48 et 96 Ko ouvert à tous les utilisateurs résidant à Pau et dans sa région.

Jérôme THIBERGE
11, lot. Curie
64230 Lescar

Habitant d'Aix-en-Provence cherche possesseur d'un Lynx 48 Ko pour contacts et échanges de programmes.

LIONEL
Tél. : (42) 64.55.61

Pour tous les Lynxistes anglophones et soucieux d'un retour aux sources, Segimex est heureux de vous communiquer l'adresse du NILUG (National Independent Lynx User Group), club britannique des utilisateurs de Lynx.

NILUG
Robert Poate
53 Kingswood Avenue
Saunderstead
South Croydon surrey
CR2 9DQ

GÉNÉRATEUR BASIC

Tous les utilisateurs de Lynx désireux de développer leurs propres programmes Basic sont confrontés aux mêmes problèmes : il est souvent long d'écrire les lignes d'instructions correspondant à des mises en page de texte ou de graphismes. Le programme développé par Claude Aussage permet de reproduire les utilitaires de développement existants sur les gros systèmes informatiques (générateurs d'écrans).

L'utilisateur dispose de douze fonctions accessibles par menu à l'aide des touches de déplacement du curseur. La fonction génère une ligne Basic comportant l'instruction choisie (un PRINT ou un MOVE par exemple) comportant directement le texte ou les coordonnées des points inscrits à l'écran. Ces lignes sont numérotées à partir de 1000, avec un incrément de 10. Ce programme autogénéré efface progressivement le générateur, et le résultat peut être sauvegardé

sur cassette comme n'importe quel programme Basic, et relancé par un simple RUN. Bien entendu, les fonctions d'édition du Lynx permettent de modifier aisément le programme.

```

1 CODE 00 00 00 20 10 09 05 03 0F 00 00
  3C 30 28 24 22 01 00 00 00
2 CODE 19 00 7D 7E 7F 2C 7D 7E 7F 0D
3 CODE 10 00 7D 7E 7F 0D
4 CODE 00 00 00 00 00 00 1D 09 7D 7E 00
  00 00 00 00 7F 2C 7D 7E 00 00 00 00
  7F 3B 22
5 CODE 7F 2C 00 3D 7D 7E
6 CODE 11 0F 49 3D 7D 27 7F DD 7D 0D 7F
  0D 0F 22 7D 09 7F 2C 7D 28 7F 0D 09 05
  49 0D 00
7 DATA PAUSE.MOVE.DRAW.PRINT. INK.STOP
8 DATA FILL.CADRE, DOT.SIZE.PAPER, CLS
9 DIM C$(40)
10 PROTECT /
11 VDU 25
12 CCHAR &20/21
13 LET Z=DPEEK(&51FC)+100
14 LET L=1000
15 DPOKE GRAPHIC,LCTN(1)
16 PAPER BLUE
17 CLS
18 INK /
19 FOR D=2 TO 200 STEP 36
20 MOVE D.235
21 DRAW D.248
22 DRAW D+34.248
23 DRAW D+34.235
24 DRAW D.235
25 READ L$
26 PRINT @ D/2+1.237:L$:
27 NEXT D
28 MOVE 218.248
29 DRAW 252.248

```

```

30 DRAW 252.225
31 DRAW 218.225
32 DRAW 218.248
33 PRINT @ 110.227:"I P S":
34 LET A#=CHR$(2)+CHR$(1)+CHR$(1)+CHR$(7)
  )
35 LET A#=CHR$(128),B#=CHR$(129)
36 MOVE 0,223
37 DRAW 255.223
38 MOVE /,203
39 LET H=/,V=203,h=1,v=203,M=/,N=203,t=1
40 LET X=2,x=3,C=15,F=TRUE,K=7,P=1,Y=/,M
  =K,B=P
41 INK K
42 PRINT @ 110.237:K:
43 PRINT @ 116.237:P:
44 PRINT @ 122.236:CHR$(125+t):
45 PROC PR
46 REPEAT
47 PAPER BLUE
48 INK 7
49 LET X=X+4*((INP(&0980)=223)-(INP(&0
  980)=251))
50 IF ((INP(&0080)=223) OR (INP(&0080)
  =239)) THEN PROC FONC
51 IF X>105 THEN LET X=105
52 IF X<2 THEN LET X=2
53 IF x=X THEN GOTO LABEL A
54 PRINT @ x.224:" ":
55 PRINT @ X.224:A$:
56 LET x=X
57 LABEL A
58 IF (INP(&0980)<>247) THEN GOTO LA
  BEL B
59 LET C=INT(X/18+0.05)+Y
60 IF C=/ THEN PROC PS
61 IF C=1 OR C=2 OR C=3 OR C=8 THEN P
  ROC MV
62 IF C=4 OR C=10 THEN PROC IK
63 IF C=6 OR C=7 THEN PROC FF
64 IF C=9 THEN PROC SZ
65 IF C=11 THEN PROC CL
66 IF C=5 OR Z>&9800 THEN PROC FIN
67 LABEL B
68 IF (INP(&0280)=223) THEN PROC HP
69 UNTIL F=FALSE
71 END
72 DEFPROC FONC
73 INK /
74 LET Y=6-Y
75 RESTORE 7+Y/6
76 FOR I=2 TO 200 STEP 36
77 READ L$
78 PRINT @ I/2+1.237:" ":
79 PRINT @ I/2+1.237:L$:
80 NEXT I
81 ENDPROC
82 DEFPROC FIN
83 CLS
84 PRINT @ 2.20:"Attendez...":
85 POKE Z,&0080
86 FOR I=DPEEK(&51FC) TO Z-100
87 POKE I,PEEK(I+100)
88 NEXT I
89 DPOKE (&51FC),Z-100
90 PRINT L-1000:" Lignes crees : 1000-":
  L-14

```

```

91 LET F=FALSE
92 ENDPROC
93 DEFPROC FF
94 PROC MV
95 LET m=M.n=N
96 PAUSE 2000
97 PROC MV
98 IF m>M THEN SWAP m,M
99 IF C=7 THEN GOTO LABEL E
100 FOR I=m TO M
101 MOVE I,n
102 DRAW I,N
103 NEXT I
104 PROC FC
105 LET C=15
106 ENDPROC
107 LABEL E
108 LET S=M.U=N
109 LET C=1
110 PROC MVC
111 LET M=m.C=2
112 PROC MVC
113 LET N=n
114 PROC MVC
115 LET M=5
116 PROC MVC
117 LET N=U
118 PROC MVC
119 LET C=15
120 ENDPROC
121 DEFPROC MV
122 PAUSE 1000
123 PAPER B
124 IF K=B THEN INK INK+1
125 ELSE INK K
126 REPEAT
127 LET H=H+2*((INP(&0980)=223)-(INP(&
0980)=251))
128 LET V=V+4*((INP(&0080)=223)-(INP(&
0080)=239))
129 IF H>123 THEN LET H=123
130 IF H<0 THEN LET H=0
131 IF V>203 THEN LET V=203
132 IF V<0 THEN LET V=0
133 PRINT @ h,v:" ":
134 PRINT @ H,V:B$:
135 LET h=H.v=V
136 LET M=H*2.N=V
137 UNTIL (INP(&0980)=247)
138 PRINT @ H,V:" ":
139 IF C=1 OR C=8 OR C=2 THEN PROC MVC
140 IF C=3 THEN PROC PT
141 IF C=6 OR C=7 THEN ENDPROC
142 PAUSE 1000
143 LET C=15
144 ENDPROC
145 DEFPROC IK
146 PAUSE 1000
147 IF C=4 THEN LET d=0
148 ELSE LET d=6
149 PRINT @ 110+d.237:"?":
150 REPEAT
151 LET G=GETN-48
152 IF G<0 OR G>7 THEN GOTO LABEL F
153 LET M=G
154 INK M
155 IF M=1 THEN INK 0
156 PRINT @ 110+d.237:M:"?":
157 LABEL F
158 UNTIL G=-35
159 IF C=4 THEN LET K=M
160 ELSE LET P=M
161 PRINT @ 113+d.237:" ":
162 PROC IKC
163 LET C=15
164 PAUSE 1000
165 ENDPROC
166 DEFPROC PS
167 LET M=20000
168 PROC IKC
169 ENDPROC
170 DEFPROC CL
171 PRINT @ 104.236:"*":
172 PAPER P
173 FOR I=0 TO 21
174 PRINT @ 0,I*10;CHR$(30):
175 PRINT @ 123,I*10:" ":
176 PRINT @ 125,I*10:" ":
177 NEXT I
178 INK P
179 FOR I=0 TO 2
180 MOVE 0.220+I
181 DRAW 255.220+I
182 NEXT I
183 PAPER BLUE
184 PRINT @ 104.236:" ":
185 PROC CLC
186 LET C=15.B=P
187 LET H=0,V=203,h=1,v=203
188 ENDPROC
189 DEFPROC PT
190 PAPER P
191 INK K
192 VDU 24+t
193 LET C$=" ",l=0 l?
194 IF t=0 THEN LET V=V*0.5
195 LET M=H.N=V
196 PAUSE 2000
197 REPEAT
198 PRINT @ H,V:CHR$(62):
199 LET z=GETN
200 IF z<32 OR z>122 THEN GOTO LABEL
D
201 PRINT @ H,V:CHR$(z):
202 LET C$=C$+CHR$(z)
203 LABEL D
204 LET l=LEN(C$) l=LEN(C$)
205 IF z=8 THEN PRINT CHR$(8):
206 IF l>0 AND z=8 THEN LET C$=LEFT$(
C$,l-1) l?
207 IF l=1 AND z=8 THEN LET C$=""
208 LET H=H+3*((z>31 AND z<123)-(z=8))
209 UNTIL z=13 OR l=40
210 PRINT @ H,V:" ":
211 LET h=H.v=V
212 PROC PTC
213 VDU 25
214 IF t=0 THEN LET V=2*V
215 PAPER BLUE
216 PAUSE 1000
217 ENDPROC
218 DEFPROC SZ
219 LET t=1-t.M=24+t
220 PROC IKC

```

```

221 LET C=15
222 PRINT @ 122,236:CHR$(125+t):
223 ENDFPROC
224 DEFPROC MVC
225 FOR I=0 TO 4
226   POKE Z+I,PEEK(&6743+I)
227   POKE Z+9+I,PEEK(&6748+I)
228   POKE Z+18+I,PEEK(&674D+I)
229 NEXT I
230 LET T=LCTN(2)
231 FOR I=0 TO 3
232   POKE Z+5+I,PEEK(T+I)
233   POKE Z+14+I,PEEK(T+4+I)
234 NEXT I
235 DPOKE Z+23,DPEEK(T+8)
236 IF C=1 THEN POKE Z+6,&0022
237 IF C=1 THEN MOVE M,N
238 IF C=8 THEN POKE Z+6,&000B
239 IF C=8 THEN DOT M,N
240 IF C=2 THEN POKE Z+6,&0023
241 IF C=2 THEN DRAW M,N
242 LET Z=Z+&0019
243 LET L=L+1
244 ENDFPROC
245 DEFPROC IKC
246 FOR I=0 TO 4
247   POKE Z+I,PEEK(&6743+I)
248   POKE Z+9+I,PEEK(&6748+I)
249 NEXT I
250 LET T=LCTN(3)
251 FOR I=0 TO 3
252   POKE Z+5+I,PEEK(T+I)
253 NEXT I
254 DPOKE Z+14,DPEEK(T+4)
255 IF C=0 THEN POKE Z+6,&002A
256 IF C=4 THEN POKE Z+6,&0006
257 IF C=9 THEN POKE Z+6,&0001
258 IF C=10 THEN POKE Z+6,&0007
259 LET Z=Z+&0010,L=L+1
260 ENDFPROC
261 DEFPROC CLC
262 FOR I=0 TO 4
263   POKE Z+I,PEEK(&6743+I)
264 NEXT I
265 POKE (Z+5),&0008
266 POKE (Z+6),&0013
267 POKE (Z+7),&000D
268 LET Z=Z+&0008,L=L+1
269 ENDFPROC
270 DEFPROC PTC
271 LET G=DPEEK(&6213)
272 LET G=DPEEK(G+8),G=G+1
273 LET T=LCTN(4)
274 FOR I=0 TO 26
275   POKE Z+I,PEEK(T+I)
276 NEXT I
277 FOR I=0 TO 4
278   POKE Z+I,PEEK(&6743+I)
279   POKE Z+10+I,PEEK(&6748+I)
280   POKE Z+19+I,PEEK(&674D+I)
281 NEXT I
282 POKE Z+5,30+1 → l
283 FOR I=0 TO 1-1 ← l-1
284   POKE Z+27+I,PEEK(G+I)
285 NEXT I
286 DPOKE Z+27+1,&3B22
287 POKE Z+29+1,&000D

```

```

288 LET Z=Z+30+1,L=L+1
289 ENDFPROC
290 DEFPROC FC
291 LET T=LCTN(5)
292 FOR I=0 TO 4
293   POKE Z+I,PEEK(&6743+I)
294   POKE Z+11+I,PEEK(&6748+I)
295   POKE Z+22+I,PEEK(&674D+I)
296   POKE Z+33+I,PEEK(&67CA+I)
297   POKE Z+44+I,PEEK(&67CF+I)
298   FOR J=0 TO 5
299     POKE Z+5+11*I+J,PEEK(T+J)
300   NEXT J
301 NEXT I
302 DPOKE Z+5,&0033
303 POKE Z+7,&004D
304 POKE Z+18,&004E
305 POKE Z+29,&006D
306 POKE Z+40,&006E
307 DPOKE Z+49,&0D7F
308 LET Z=Z+51,L=L+1 ← l 1
309 FOR I=0 TO 4
310   POKE Z+I,PEEK(&6743+I)
311 NEXT I
312 LET T=LCTN(6)
313 FOR I=0 TO 11
314   POKE Z+5+I,PEEK(T+I)
315 NEXT I
316 LET L=L+1,Z=Z+17
317 FOR J=0 TO 1
318   FOR I=0 TO 4
319     POKE Z+I,PEEK(&6743+I)
320   NEXT I
321   FOR I=0 TO 9
322     POKE Z+5+I,PEEK(T+12+I)
323   NEXT I
324   LET Z=Z+15, I=I+1 ? → L=L+1
325 NEXT J
326 POKE Z-3,&000E
327 POKE Z-9,&0023
328 FOR I=0 TO 4
329   POKE Z+I,PEEK(&6743+I)
330   POKE Z+5+I,PEEK(T+22+I)
331 NEXT I
332 LET Z=Z+9,L=L+1
333 ENDFPROC
334 DEFPROC HP
335 MOVE 6,0
336 DRAW 244,0
337 DRAW 244,15
338 DRAW 6,15
339 DRAW 6,0
340 PRINT @ 4.3:"AIDE":
341 LET C=INT(X/18+0.05)+Y
342 IF C=0 OR C=5 OR C=9 OR C=11 THEN R
  ESTORE 359
343 IF C=1 OR C=2 OR C=8 THEN RESTORE 3
  60
344 IF C=6 OR C=7 THEN RESTORE 361
345 IF C=4 OR C=10 THEN RESTORE 362
346 IF C=3 THEN RESTORE 363
347 READ U
348 FOR I=1 TO U
349   LET S=14+23*I
350   MOVE S,2
351   DRAW S+21,2
352   DRAW S+21,13

```

```

353 DRAW 5,13
354 DRAW 5,2
355 READ A,G,T
356 LET C#=CHR$(A)+CHR$(G)+CHR$(T)
357 PRINT @ 5/2+1,3;C#;
358 NEXT I
359 DATA 1,82,101,116
360 DATA 3,82,101,116,124,32,125,82,101,
116
361 DATA 5,82,101,116,124,32,125,82,101,
116,124,32,125,82,101,116
362 DATA 3,82,101,116,78,117,109,82,101,
116
363 DATA 9,82,101,116,124,32,125,82,101,
116,32,65,32,32,66,32,68,101,108,69,116,
99,46,46,46,82,101,116
364 LET C=15
365 MOVE M,N
366 PAUSE 20000
367 PRINT @ X,0;CHR$(30);
368 PRINT @ X,6;CHR$(30);
369 ENDPROC
370 DEFPROC PR
371 PRINT @ 4,6;" GENERATEUR BASIC V1.0
";CHR$(94);CHR$(127);"1984 C.Aussage";
372 PRINT @ 3,224;CHR$(125);CHR$(126);C
HR$(124);CHR$(123);"boude ";CHR$(128);"
. Ret:valide . A:aide";
373 PAUSE 50000
374 PRINT @ X,6;CHR$(30);
375 FOR X=102 TO 2 STEP -2
376 PRINT @ X,224;CHR$(128);" ";
377 NEXT X
378 ENDPROC

```

Générateur Basic : mode d'emploi

Voici la liste des instructions Basic générées par le programme, ainsi que les séquences de touches correspondantes :

Fonction	Effet	Touches
PAUSE	Arrêt de 2 secondes	RETURN
STOP	Génération Basic	RETURN
SIZE	Taille des caractères	RETURN
CLS	Effacement d'écran	RETURN
MOVE		
DOT	Déplacement du point	RETURN et flèches
DRAW		
INK	Changement de couleur	RETURN numéro RETURN
PAPER	Changement de couleur	RETURN numéro RETURN
PRINT	Affichage texte	RETURN flèches RETURN
FILL	Dessin de rectangle	RETURN flèches RETURN flèches RETURN
CADRE	Dessin de rectangle	RETURN flèches RETURN flèches RETURN

REDÉFINITION DE CARACTÈRES

Le club des utilisateurs de Lynx de la région de Pau (un grand bonjour à tous!) nous envoie ce petit programme permettant de redéfinir un caractère sans détruire tous les autres :

```

10 RESERVE 39700
20 DPOKE ALPHA,HIMEM
30 LET J=HIMEM
40 FOR I=148 TO 3357
50 DPOKE J,PEEK(I)
60 J=J-1
70 NEXT I

```

Le programme se charge par un RUN, suivi de NEW lorsque le curseur réapparaît. Pour l'utiliser, il suffit d'écrire un programme de redéfinition de caractère semblable à celui-ci :

```

10 RESERVE HIMEM -10
20 FOR I=0 TO 9
30 READ A
40 POKE LETTER(90)+I,A
50 NEXT I
60 DATA 0,3,2,3,2,26,62,60,24,0

```

En ligne 40, on indique entre parenthèse le code ASCII correspondant à la touche à modifier (ici 90 c'est-à-dire le Z). Lorsque l'on appuie sur cette touche, on obtient le caractère dessiné par les valeurs incluses dans les DATA de la ligne 60 (ici une double croche).

REDÉFINITION DE CARACTÈRES (Bis)

Lorsque l'on cherche à redéfinir un caractère ASCII sur le Lynx, la méthode préconisée dans le manuel oblige à introduire en DATA une longue liste de 0 et de 1, coûteuse en mémoire, fastidieuse et souvent génératrice d'erreur.

La solution la plus simple consiste à transformer les données binaires en autant de puissances de 2. Par exemple, le codage de la double croche peut se faire comme suit, en inscrivant au dessus de chaque colonne les puissances de deux dans l'ordre croissant en partant de la droite (1,2,4,8,16,32). Il suffit alors de faire la somme ligne par ligne des cases

noircies (dans lesquelles il y aura un point) :

	32	16	8	4	2	1	
0							0
3							3
2							2
3							3
2							2
26							26
62							62
60							60
24							24
0							0

Le programme suivant permet alors de redéfinir la double croche comme correspondant au code ASCII 128 :

```

10 RESERVE HIMEM -10
20 DPOKE GRAPHIC,HIMEM
30 FOR I=0 TO 9
40 READ R
50 POKE LETTER(128)+I,R
60 NEXT I
70 DATA 0,3,2,3,2,26,62,60,24,0
80 PRINT CHR$(128)

```

JEU DE L'OIE

Considéré par les spécialistes comme l'un des plus anciens jeux du monde, le Jeu de l'Oie connaît actuellement un regain d'intérêt. Peut-être que cette mise au goût du jour sur ordinateur va susciter de nouvelles vocations. Conçu pour jouer à plusieurs (de 1 à 6 joueurs), le Jeu de l'Oie programmé par Didier Desjardin tire le meilleur parti des possibilités graphiques du Lynx.

```

100 REM "JEU DE L'OIE"
110 SPEED 0
120 WINDOW 3,123,5,245
130 CLS
140 VDU 1.2.2.1.7.24
150 PRINT @ 40.60:"STOP THE TAPE":
160 PAUSE 20000
170 VDU 7.25
180 CLS
190 FOR J=0 TO 7
200 PAPER J
210 FOR I=0 TO 2
220 FOR K=0 TO 7
230 INK K
240 PRINT CHR$(128)+CHR$(128)+CHR$(
(128)+CHR$(128)+CHR$(128):
250 BEEP 50+80*J.50-8*J.63
260 NEXT K
270 NEXT I
280 NEXT J
290 VDU 1.7.2.0.24
300 PRINT @ 42.57.5:"JEU DE L'OIE":
310 VDU 1.2.2.0.25
320 PRINT @ 12.140:"POUR COMMENCER PR
ESSEZ UNE TOUCHE":
330 LET X=GETN
340 CLS
350 VDU 2.0.24
360 CLS
370 PRINT @ 42.5:"JEU DE L'OIE":
380 RESERVE HIMEM=20
390 DPOKE GRAPHIC.HIMEM LETN(100)
400 FOR J=0 TO 17
410 READ A
420 POKE LETTER(128)+J,BIN(A) AF 0E 04 0C
430 NEXT J
440 DATA 010101,101010,010101,101010,010
101,101010,010101,101010,010101,101010
450 DATA 010000,110000,010000,010000,010
110,011111,011111,001110,000100,001100
460 VDU 1.6.2.0.25.20
470 PRINT @ 12.40:"CE JEU DE L'OIE PEUT
SE JOUER A SIX , CHAQUE JOUEUR A UN PLO
N D'UNE COULEUR .POUR JOUER IL SUFFIT DE
TAPER SUR LE NUMERO DE LA COULEUR":
480 PRINT @ 21.100:"VOICI LES PIONS ET
LEUR CODE":
490 VDU 24
500 FOR I=1 TO 6
510 INK I
520 PRINT @ 20*I-5.70:CHR$(129):
530 PRINT @ 20*I-5.80:I:
540 NEXT I
550 VDU 1.6.2.0.25
560 PRINT @ 21.180:"COMBIEN ETES-VOUS D
E JOUEURS"
570 VDU 1,2,24
580 PRINT @ 1,100;" 1 2 3
4 5 6 ";
590 LET B=GETN-48
600 IF B>6 THEN GOTO 580
610 VDU 1.2,25
620 PRINT @ 21.230:"POUR JOUER TAPE SUR
UNE TOUCHE"
630 LET X=GETN
640 DIM V(62),A(6),A$(6)(10)
650 WINDOW 3,123,5,245
660 SPEED 0
670 FOR I=0 TO 62
680 READ V(I)
690 NEXT I
700 DATA 26.007,39.007,52.007,65.007,78.
007,91.007,104.007,104.029,104.051,104.0
73,104.095,104.117,104.139,104.161,104.1
83,104.205,104.227,91.227,78.227,65.227.
52.227,39.227,26.227,13.227
710 DATA 13.205,13.183,13.161,13.139,13.
117,13.095,13.073,13.051,13.029,26.029,3
9.029,52.029,65.029,78.029,91.029,91.051
,91.073,91.095,91.117,91.139,91.161,91.1
83,91.205
720 DATA 78.205,65.205,52.205,39.205,26.
205,26.183,26.161,26.139,26.117,26.095,2
6.073,26.051,39.051,52.051,65.051,78.051
730 FOR I=0 TO B
740 LET A(I)=0
750 NEXT I
760 VDU 1.7,2,0
770 CLS
780 FOR K=0 TO 10
790 PAPER K
800 FOR I=0 TO 15
810 INK I
820 PRINT @ 13+13*(I MOD 8),7+(22*K
)+10*(I DIV 8):CHR$(128)+CHR$(128)+CHR$(
128)+CHR$(128):
830 BEEP 100+10*K,100-5*K,63
840 NEXT I
850 NEXT K
860 FOR I=0 TO 8
870 VDU 1.7,2,0
880 MOVE 24+26*I,7
890 DRAW 24+26*I,247
900 BEEP 100+10*I,100-5*I,63
910 NEXT I
920 FOR I=0 TO 11
930 MOVE 25,6+22*I
940 DRAW 232,6+22*I
950 BEEP 200+10*I,50-2*I,63
960 NEXT I
970 PROC DEL (0)
980 PRINT @ 23,7:CHR$(32)+CHR$(32)+CHR$(
32)+CHR$(32)+CHR$(32):
990 PRINT @ 23,17:CHR$(32)+CHR$(32)+CHR
$(32)+CHR$(32)+CHR$(32):
1000 VDU 1,7,2,0
1010 PRINT @ 13,7:"START":
1020 FOR I=1 TO 62
1030 PRINT @ INT(V(I)),(1000*FRAC(V(I
))):INT(I/10):
1040 PRINT @ INT(V(I)),(1000*FRAC(V(I
)))+10:I-10*INT(I/10):

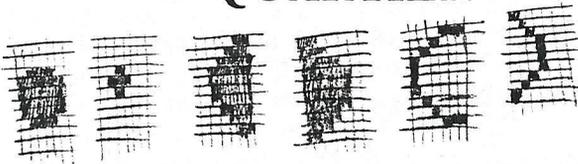
```

```
1050 BEEP 100+I,160-2*I,63
1060 NEXT I
1070 FOR I=1 TO B
1080 INK I
1090 LET X=3+INT(V(0))+3*((I-1) MOD 3)
1100 LET Y=1000*FRAC(V(0))+10*((I-1) DIV
IV 3)
1110 PRINT @ X,Y:CHR$(129):
1120 NEXT I
1130 WINDOW 40,88,78,189
1140 VDU 23
1150 FOR I=1 TO B
1160 INK I
1170 INPUT "QUEL EST TON NOM":A$
1180 BEEP 200-10*I,50+5*I,63
1190 LET A$(I)=A$
1200 PRINT "TON PION C'EST ":CHR$(129)
1210 NEXT I
1220 WINDOW 3,123,5,245
1230 PROC DEL (0)
1240 LET I=-1
1250 LET I=I+1,J=1+(I MOD B)
1260 INK J
1270 PRINT @ 54,73:A$(J):" ";
1280 PRINT @ 39,93:"TU TAPES TON CODE":
1290 BEEP 200,200,63
1300 BEEP 100,100,63
1310 BEEP 300,50,63
1320 LET G=BETN-48
1330 IF G<>J THEN GOTO 1290
1340 PRINT @ 39,113:"JE LANCE LES DES":
1350 LET D=RAND(6)+1,d=RAND(6)+1
1360 SPEED 20
1370 PRINT @ 39,133:" ET TU AS OBTENU":
1380 PRINT @ 51,153:D:" ET ":d:
1390 PROC JEU(J)
1400 IF D<>d THEN GOTO 1250
1410 PRINT @ 39,93:" TU AS UN DOUBLE ":
1420 PRINT @ 39,113:" JE RELANCE "
:
1430 GOTO 1350
1440 DEFPROC JEU(J)
1450 INK 0
1460 LET K=A(J),X=3+INT(V(K))+3*((J-1) MOD 3)
1470 LET Y=1000*FRAC(V(K))+10*((J-1) DIV
3)
1480 PRINT @ X,Y:CHR$(32):
1490 BEEP 100,50,63
1500 LET A(J)=A(J)+D+d
1510 PROC COMP
1520 IF A(J)>63 THEN LET A(J)=126-A(J)
1530 INK J
1540 LET K=A(J),X=3+INT(V(K))+3*((J-1) MOD 3)
1550 LET Y=1000*FRAC(V(K))+10*((J-1) DIV
3)
1560 PRINT @ X,Y:CHR$(129):
1570 BEEP 200,50,63
1580 ENDPROC
1590 DEFPROC COMP
1600 IF A(J)=63 THEN PROC FIN
1610 IF A(J)=3 OR A(J)=21 OR A(J)=34 OR
A(J)=45 OR A(J)=50 THEN PROC AV
1620 IF A(J)=10 OR A(J)=19 OR A(J)=31 OR
A(J)=37 OR A(J)=41 OR A(J)=53 THEN PRO
C REC
```

```
1630 IF A(J)=7 OR A(J)=13 OR A(J)=25 OR
A(J)=32 OR A(J)=39 OR A(J)=48 THEN PROC
REJ
1640 IF A(J)=14 OR A(J)=27 OR A(J)=43 OR
A(J)=57 THEN PROC PAS
1650 ENDPROC
1660 DEFPROC AV
1670 INK J
1680 PRINT @ 39,93:" TU AVANCES DE ":
1690 PRINT @ 39,113:" DEUX CASES. ":
1700 BEEP 100,100,63
1710 PAUSE 10000
1720 LET A(J)=A(J)+2
1730 ENDPROC
1740 DEFPROC REC
1750 INK J
1760 PRINT @ 39,93:" TU RECULES DE ":
1770 PRINT @ 39,113:" DEUX CASES. ":
1780 BEEP 110,90,63
1790 PAUSE 10000
1800 LET A(J)=A(J)-2
1810 ENDPROC
1820 DEFPROC REJ
1830 INK J
1840 PRINT @ 39,93:" TU PEUX REJOUER ":
1850 BEEP 120,80,63
1860 PAUSE 10000
1870 LET D=d
1880 ENDPROC
1890 DEFPROC PAS
1900 INK J
1910 PRINT @ 39,93:" TU PASSES TON ":
1920 PRINT @ 39,113:" TOUR. ":
1930 BEEP 130,70,63
1940 PAUSE 10000
1950 LET A(J)=A(J)-(D+d)
1960 ENDPROC
1970 DEFPROC FIN
1980 SPEED 0
1990 PROC DEL (J)
2000 INK J+3
2010 PRINT @ 39,103:"*****"
:
2020 PRINT @ 39,113:"*":
2030 PRINT @ 87,113:"*":
2040 PRINT @ 39,123:"*":
2050 PRINT @ 87,123:"*":
2060 PRINT @ 39,133:"*****"
:
2070 PRINT @ 52,113:A$(J):
2080 PRINT @ 52,123:"A GAGNE":
2090 PRINT @ 44,153:" FOUR REJOUER ":
2100 PRINT @ 44,173:"TAPE SUR UNE ":
2110 PRINT @ 56,193:"TOUCHE":
2120 LET X=BETN
2130 RUN 350
2140 DEFPROC DEL (J)
2150 PAPER J
2160 FOR I=0 TO 12
2170 FOR K=0 TO 16
2180 PRINT @ 39+3*K,73+10*I:CHR$(32
):
2190 NEXT K
2200 BEEP 100+10*I,100-5*I,63
2210 NEXT I
2220 ENDPROC
```

par Bernard TORGUE
et Laurent COURTIN

PAS DE QUARTIER



Ce jeu est une réponse parfaite à une opinion communément répandue en micro-informatique : "pour réaliser un programme du niveau de ceux qui sont commercialisés, il est nécessaire de maîtriser la programmation en langage assembleur".

C'est peut-être vrai pour les Basic courants (Microsoft ou Apple soft par exemple), mais le Lynx Basic est suffisamment puissant (programmation structurée, grand choix d'instructions graphiques, etc.) pour permettre d'écrire des programmes d'action rapide sans connaître le langage machine. "Pas de quartier" en est une parfaite illustration. Jeu à la fois d'actions rapides et de stratégie basée sur le scénario classique d'une guerre entre le joueur et les méchants, il n'a rien à envier aux programmes du commerce, tant par la qualité du scénario que par celles des graphismes.

A vous de jouer ! Tentez de gagner bataille sur bataille (car la guerre ne se gagne — ni ne se perd ! — pas sur une seule victoire...), en tuant le plus d'ennemis possibles, mais en épargnant (si possible) les innocents touristes...

```
100 CODE 00 00 1E 3F 3F 3F 3F 1E 00 00 00 00 00
    0B 1C 0B 00 00 00 00 03 07 0F 1F
    1F 1F 1F 0F 07 03 30 38 3C 3E 3E
    3E 3E 3C 38 30 07 08 10 20 20 20
    20 10 0B 07 20 10 0B 04 04 04 0B
10 20
```

```
80 CLS
81 PRINT @ 40.120:" B.L. 1984":
100 RESERVE HIMEM=60
110 PPOKE GRAPHIC.HIMEM LOTN(100)
120 FOR J=0 TO 59
130 READ A
140 PPOKE LETTER(128)+JTBIN(A)
150 NEXT J
160 DATA 000000.000000.011110.111111.111
170 DATA 111111.111111.011110.000000.000000
180 DATA 000000.000000.000000.001000.011
190 DATA 001000.000000.000000.000000.000000
200 DATA 000011.000111.001111.011111.011
210 DATA 011111.011111.001111.000111.000011
220 DATA 110000.111000.111100.111110.111
230 DATA 111110.111110.111100.111000.110000
240 DATA 000111.001000.010000.100000.100
250 DATA 100000.100000.010000.001000.000111
```

```
210 DATA 100000.010000.001000.000100.000
100.000100.000100.001000.010000.100000
220 DIM W(9).V(2)
230 LET N=0.M=0.D=0.P=0.R=0.Q=0.a=0.h=0.
B=0.w=TRUE.l=30
240 PROTECT BLACK
250 CLS
260 RANDOM
270 FOR N=1 TO 9
280 LET W(N)=TRUE
290 NEXT N
300 INK BLUE
310 MOVE 40.0
320 DRAW 40.248
330 MOVE 128.0
340 DRAW 128.248
350 MOVE 210.0
360 DRAW 210.248
370 MOVE 0.20
380 DRAW 255.20
390 MOVE 0.122
400 DRAW 255.122
410 MOVE 0.220
420 DRAW 255.220
430 PRINT @ 22.235:"MORTS:";a;"000":
440 PRINT @ 66.235:"POINTS:";h;
450 PRINT @ 34.7:"MISSILES":
460 PRINT @ 66.7:"RESTANTS:";1;
470 LET x=137.v=22
480 MOVE x.v
490 RESTORE 2490
500 READ X.Y
510 LET x=x+(X*.8)
520 LET v=v+Y
530 DRAW x.v
540 IF X=0 AND Y=0 THEN GOTO 560
550 ELSE GOTO 500
560 INK MAGENTA
570 VDU 21
580 PRINT @ 68.35:CHR$(128):
590 PRINT @ 86.188:CHR$(128):
600 PRINT @ 48.185:CHR$(128):
610 PRINT @ 97.65:CHR$(128):
620 PRINT @ 67.65:CHR$(128):
630 PRINT @ 57.50:CHR$(128):
640 PRINT @ 30.75:CHR$(128):
650 PRINT @ 80.137:CHR$(128):
660 PRINT @ 63.118:CHR$(128):
670 VDU 20
680 PROTECT MAGENTA
690 LET i=128.j=123.l=0.j=0
700 IF RAND(2)+1=2 THEN LET V(1)=TRUE.V
(2)=TRUE
710 ELSE LET V(1)=TRUE.V(2)=FALSE
720 IF B>10 THEN LET V(1)=TRUE.V(2)=TRU
E
730 IF NOTV(2) THEN GOTO 1010
740 LET C=INT(RND*4)+1
750 LET D=INT(RND*9)+1
760 IF B>19 THEN LET D=9
770 IF B>22 THEN LET C=1
780 GOTO (C*20)+780
790 REM
800 LET a=INT(RND*256).o=0
810 GOTO 870
820 LET a=0.b=INT(RND*248)
```

PAS DE QUARTIER

```
830 GOTO 870
840 LET a=INT(RND*256).b=247
850 GOTO 870
860 LET a=255.b=INT(RND*248)
870 IF D=1 AND W(D) THEN LET E=138.F=40
880 IF D=2 AND W(D) THEN LET E=178.F=19
3
890 IF D=3 AND W(D) THEN LET E=98.F=190
900 IF D=4 AND W(D) THEN LET E=196.F=70
910 IF D=5 AND W(D) THEN LET E=136.F=70
920 IF D=6 AND W(D) THEN LET E=116.F=55
930 IF D=7 AND W(D) THEN LET E=62.F=80
940 IF D=8 AND W(D) THEN LET E=162.F=14
2
950 IF D=9 AND W(D) THEN LET E=128.F=12
3
960 IF NOTW(D) THEN GOTO 750
970 LET e=E-a.f=F-b
980 IF ABS(e)<ABS(f) THEN LET O=e/(ABS(f)/2).P=f/(ABS(f)/2)
990 ELSE LET O=e/(ABS(e)/2).P=f/(ABS(e)/2)
1000 LET N=a.M=b.O=E.R=F.T=D
1010 REM
1020 LET C=INT(RND*4)+1
1030 LET D=INT(RND*9)+1
1040 IF B>22 THEN LET C=3
1050 GOTO (C*20)+1050
1060 REM
1070 LET a=INT(RND*256).b=0
1080 GOTO 1140
1090 LET a=0.b=INT(RND*248)
1100 GOTO 1140
1110 LET a=INT(RND*256).b=247
1120 GOTO 1140
1130 LET a=255.b=INT(RND*248)
1140 IF D=1 AND W(D) THEN LET E=138.F=4
0
1150 IF D=2 AND W(D) THEN LET E=178.F=1
93
1160 IF D=3 AND W(D) THEN LET E=98.F=19
0
1170 IF D=4 AND W(D) THEN LET E=196.F=7
0
1180 IF D=5 AND W(D) THEN LET E=136.F=7
0
1190 IF D=6 AND W(D) THEN LET E=116.F=5
5
1200 IF D=7 AND W(D) THEN LET E=62.F=80
1210 IF D=8 AND W(D) THEN LET E=162.F=1
42
1220 IF D=9 AND W(D) THEN LET E=128.F=1
23
1230 IF NOTW(D) THEN GOTO 1030
1240 LET e=E-a.f=F-b
1250 IF ABS(e)<ABS(f) THEN LET G=e/(ABS(f)/2).H=f/(ABS(f)/2)
1260 ELSE LET G=e/(ABS(e)/2).H=f/(ABS(e)/2)
1270 INK GREEN
1280 PRINT @ (1 DIV 2)-1.j-5:CHR*(129):
1290 IF V(1) THEN DOT a.b
1300 IF V(2) THEN DOT N.M
1310 INK RED
1320 IF V(1) AND (a>E-1.5 AND b>F-1.5) A
ND (a<E+1.5 AND b<F+1.5) THEN GOTO 1590
1330 IF V(2) AND (N>Q-1.5 AND M>R-1.5) A
ND (N<Q+1.5 AND M<R+1.5) THEN GOTO 1610
1340 IF V(1) AND (i>a-1.5 AND j>b-1.5) A
ND (i<a+1.5 AND j<b+1.5) THEN GOTO 1510
1350 IF V(2) AND (i>N-1.5 AND j>M-1.5) A
ND (i<N+1.5 AND j<M+1.5) THEN GOTO 1530
1360 BEEP 100,5,63
1370 INK BLACK
1380 PRINT @ (1 DIV 2)-1.j-5:CHR*(129):
1390 IF V(1) THEN DOT a.b
1400 IF V(2) THEN DOT N.M
1410 IF V(1) THEN LET a=a+G
1420 IF V(2) THEN LET b=b+H
1430 IF V(2) THEN LET N=N+O.M=M+P
1440 LET A=INP(&0980)
1450 LET a=((A-223)-(A-251)) MOD 248
1460 LET A=INP(&0080)
1470 LET r=((A-223)-(A-239)) MOD 198
1480 IF INP(&00FA)<>255 THEN GOSUB 2390
1490 GOSUB 2260
1500 GOTO 1270
1510 LET z=a.x=b.V(1)=FALSE.c=FALSE
1520 GOTO 1540
1530 LET z=N.x=M.V=1.V(2)=FALSE.c=FALSE
1540 LET B=B+1.l=1-1
1550 LET a=a+175
1560 LET h=h+700
1570 LET w=TRUE
1580 GOTO 1650
1590 LET z=E.x=F.W(D)=FALSE.V(1)=FALSE.c
=TRUE
1600 GOTO 1620
1610 LET z=0.x=R.V=1.W(T)=FALSE.V(2)=FAL
SE.c=TRUE
1620 LET a=a+6519
1630 LET h=h-1000
1640 LET w=FALSE
1650 INK BLUE
1660 PROTECT BLACK
1670 PRINT @ 40,235:a:"000":
1680 PRINT @ 87,235:" "":
1690 PRINT @ 93,7:" "":
1700 IF w AND (B=10 OR B=20 OR B=30) THE
N LET h=h+5000
1710 ELSE GOTO 1800
1720 PROTECT MAGENTA
1730 INK GREEN
1740 VDU 18
1750 PRINT @ z DIV 2-6.x-5:"5000":
1760 VDU 18,7
1770 PRINT @ z DIV 2-6.x-5:" "":
1780 PROTECT BLACK
1790 IF B=30 THEN GOTO 2170
1800 INK BLUE
1810 IF h<0 THEN LET h=0
1820 PRINT @ 93,7;l:
1830 PRINT @ 87,235:h:
1840 IF c THEN PROTECT BLUE
1850 ELSE PROTECT MAGENTA
1860 INK GREEN
1870 PRINT @ (z DIV 2)-2.x-5:CHR*(130):
CHR*(131):
1880 FOR X=0 TO 2
1890 BEEP 100,5,63
1900 BEEP 50,10,63
1910 NEXT X
```


INITIATION

ASSEMBLEUR

Lors de notre dernière rencontre, nous nous étions arrêtés à la cinquième étape de notre premier programme. Notre problème était double, vous vous en souvenez certainement. Il fallait à la fois diminuer de 1 (les informaticiens disent "décrémenter") le contenu du registre B, et tester ce contenu pour ne pas dépasser la valeur 0. Pour réaliser cette opération, nous avons besoin de connaître un nouveau code, qui est 10nn. Ce code réalise les deux opérations. Premièrement, il décrémente le registre B. ensuite, il teste le contenu de ce registre. S'il est différent de 0, il passe à l'instruction suivante. Dans le cas contraire, c'est-à-dire si B contient 0, il effectue un saut en aval de nn octets. Si B contient le nombre d'octets qui doivent être affichés, ce code va nous permettre d'arrêter l'affichage dès que tous les caractères ont été affichés à l'écran.

DIS-MOI COMMENT TU T'APPELLES !

Notre programme a maintenant l'allure suivante :

1^{ère} étape : 21 11 hh (charge le registre HL avec l'adresse de DISPLAY YOUR NAME, moins 1)

06 nn (charge B avec le nombre d'octets à afficher)

2^{ème} étape : 23 (incrémente HL)

3^{ème} étape : 7E (charge A avec l'adresse pointée par HL)

4^{ème} étape : CD B8 62 (appelle la routine d'affichage d'un caractère)

5^{ème} étape : 10 nn (décrémente B et teste le résultat, avec retour à la deuxième étape)

6^{ème} étape : C9 (Retour)

Maintenant, il ne reste plus (délicat euphémisme) qu'à remplir les nombres manquants. Premièrement, nous allons décider de l'endroit où nous allons ranger les caractères qui forment DISPLAY YOUR NAME. La place la plus

simple est la fin du programme. Ensuite, il faut aussi trouver l'adresse où l'on va mettre le programme lui-même ! Pourquoi pas 9000 (la suite dans les idées...). On peut donc en déduire que l'adresse où seront stockés les caractères de DISPLAY YOUR NAME sera 9000 plus le nombre d'octets du programme. Or celui-ci en contient 13, c'est-à-dire D en hexadécimal. L'adresse en question sera donc 900D. Pour commencer la boucle qui incrémente le contenu du registre HL, il nous faut partir d'un octet avant, soit à l'adresse 900C.

La première ligne de notre programme est donc :
21 0C 90

Calculons maintenant le nombre d'octets contenu dans DISPLAY YOUR NAME. On arrive à 17. Or 17 en hexadécimal font 11. Maintenant, la ligne 12 prend sa forme définitive :
06 11

Il ne nous reste plus qu'à trouver le nombre associé au code 10 de l'étape 5. Il s'agit du nombre relatif d'octet correspondant à la longueur du "saut" dans le programme, effectué si le contenu de B est 0. Le nombre est relatif, puisqu'il correspond en fait à la longueur du déplacement comptée à partir de notre adresse actuelle. Or tous les sauts sont effectués en changeant le contenu du registre PC, appelé aussi le Compteur du Programme. Vous vous rappelez sans doute que ce registre indique au microprocesseur Z 80 où il peut trouver l'instruction suivante.

Donc, pour connaître le nombre d'octets correspondant à la longueur du saut, il faut auparavant savoir quel était le contenu du registre PC lorsque cette instruction a été réalisée... Or, lorsque le Z80 retourne à la routine d'affichage, le compteur du programme pointait le code 10. Les nn octets ne sont pas pris en compte (puisque le registre B ne

ASSEMBLEUR (III)

contient pas encore 0) et, à l'étape suivante, le registre PC pointe sur l'instruction immédiatement après, c'est-à-dire le code C9. En conclusion, lorsque le saut va être effectué, le registre PC pointe sur l'instruction C9 !

UN PAS EN AVANT UN PAS EN ARRIÈRE

Le Z80 a une façon particulière de gérer les sauts. La valeur nn indiquant le nombre d'octets correspondant à la longueur du déplacement ne peut prendre que les valeurs hexadécimales comprises entre 00 et FF. Or, ces sauts peuvent être effectués dans les deux sens, en aval (plus loin dans le programme) ou en amont (plus tôt dans le programme) ! Par convention, le microprocesseur va décider que les sauts "positifs", c'est-à-dire en aval, seront représentés par les valeurs hexadécimales 01, 02, en comptant à partir de 0, alors que les sauts "négatifs" (en remontant dans le programme) seront comptés en reculant à partir de FF : Fe pour un saut d'un octet vers l'arrière, Fd pour deux octets, etc.

Or, le contenu de notre registre PC lorsque le saut va être effectué est C9. Un saut en arrière de l'octet nous ramènerait à l'instruction 10 et serait représenté par la valeur hexadécimale FE (-1 pour le microprocesseur Z 80 !). Ainsi de suite, FD ramènerait à l'octet 06, FC à l'octet A4, FB au code CD, FA au code 7E et F9 au code 23...

Puisque le saut doit nous conduire à l'affichage du caractère suivant, il faut qu'il provoque l'incrément du contenu du registre HL, correspondant à la position des caractères de DISPLAY YOUR NAME ! La valeur recherchée est donc F9. Sous sa forme définitive, le programme prend donc l'allure suivante :

9000	21	OC	90
9003	06	11	
9005	23		
9006	7E		
9007	CD	A4	06
900A	10	F9	
900C	C9		
900D	44	D	
900E	49	I	
900F	53	S	
9010	50	P	
9011	4C	L	
9012	41	A	
9013	59	Y	
9014	20		
9015	59	Y	
9016	4F	O	
9017	55	U	
9018	52	R	
9019	20		
901A	4E	N	
901B	41	A	
901C	4D	M	
901D	45	E	

La liste des nombres à partir de l'adresse 900D correspond bien évidemment à la liste des codes ASCII des caractères à afficher (44 pour la lettre D, etc.).

PRENEZ LES COMMANDES

Maintenant, pour bien maîtriser ces commandes, il ne vous reste plus qu'à modifier ce programme. Commencer par changer le message (pourquoi pas votre vrai nom) en recommençant tous les calculs pour les adresses. N'oubliez pas que les codes ASCII sont ici exprimés en hexadécimal et non en décimal, comme dans un programme Basic. Vous pouvez aussi essayer les effets des caractères suivants :

04 (effacement d'écran)

07 (BEEP)

Et pourquoi ne pas réécrire entièrement le programme pour qu'il affiche le message à l'envers, en décrémentant le registre HL ? Le code correspondant est 2B. Amusez-vous bien !

FIN DE LA
PREMIÈRE PARTIE
DE CE COURS !

LA SUITE VOUS INTÉRESSE PEUT-ÊTRE ?

TOURNEZ VITE LA PAGE !

COUPON-RÉPONSE

NOM _____
ADRESSE _____
C.P. _____ VILLE _____
TÉL. (facultatif) _____

PRÉNOM _____

DATE : _____

SIGNATURE : _____

Je souhaite que l'ŒIL de LYNX continue et joins à la présente, pour le prochain numéro, un chèque de 50,00 F. En cas d'arrêt définitif de l'ŒIL de LYNX, mon chèque me sera retourné.

SOMMAIRE

Pages 2 et 3 – Gestion de données

Avec six instructions, transportez des données d'un programme dans un autre.

Pages 4 à 9 – Jeux de Mots

Deux programmes pour les amateurs du Petit Larousse Illustré.

Pages 10 à 15 – La boîte à outils

Une super boîte à outils avec un générateur basic hors classe.

Pages 16 et 17 – Jeu de foie

Jouez en famille avec le plus vieux jeu du monde.

Pages 18 et 19 – Initiation assembleur III

Fin de la première partie de ce cours.

Pages 21 à 23 – Pas de quartier

Prouvez votre habileté et votre courage tout en révélant vos qualités de stratège.

DERNIÈRE MINUTE

Le 8 juin dernier COMPUTERS déposait son bilan en Angleterre. Le 14 décembre 1984, ANSTON TECHNOLOGIE Ltd, une firme spécialisée en électronique, rachetait COMPUTERS.

LYNX continue !

LYNX présentera au cours des tous prochains mois ses nouveaux produits !

LYNX continue le développement soft !

Si vous êtes nombreux à nous retourner le coupon-réponse, imprimé ci-dessus, la parution de l'ŒIL de LYNX sera prolongée !