

LE LYNX



ÉDITORIAL

La presse spécialisée s'est faite dernièrement l'écho des difficultés rencontrées par la société-mère du Lynx en Grande-Bretagne au cours du premier semestre 84. Ces difficultés réelles, dues à des problèmes de gestion interne que le succès du Lynx n'a pas réussi à conjurer, se sont traduites, au mois de juin dernier, par le dépôt de bilan de cette société.

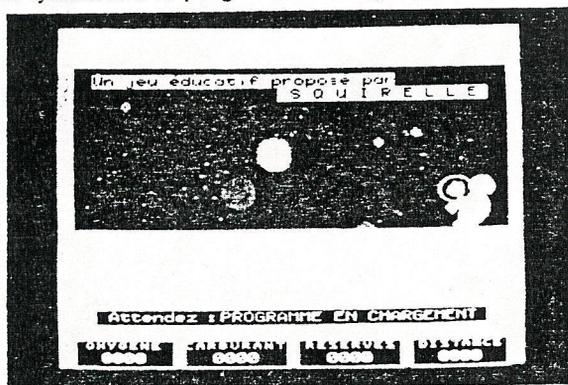
Cette nouvelle alarmante ne doit cependant pas inquiéter outre-mesure les Lynxistes français. L'importateur, Segimex, dispose encore de stocks de pièces détachées, et assure le service après-vente pour toutes les machines déjà vendues.

Ce dépôt de bilan ne signifie pas pour autant la disparition du Lynx du paysage informatique. Les succès déjà remportés et les prototypes actuellement à l'étude ont suscité l'intérêt de plusieurs sociétés, et le rachat de Computers est actuellement en cours de négociation. Rien n'est encore décidé, ni n'assure que la nouvelle équipe reconduira les contrats des importateurs actuels.

Si l'avenir est donc encore imprévisible, le Lynx n'est donc pas mort. En l'attente de nouvelles plus précises, Segimex n'est cependant pas resté inactif : la distribution de périphériques peu coûteux, de livres anglais et l'importation d'un micro-ordinateur japonais révolutionnaire, le Yashica MSX en sont les meilleures preuves.

DES DIDACTIQUES DE QUALITÉ CHEZ SQUIRELLE

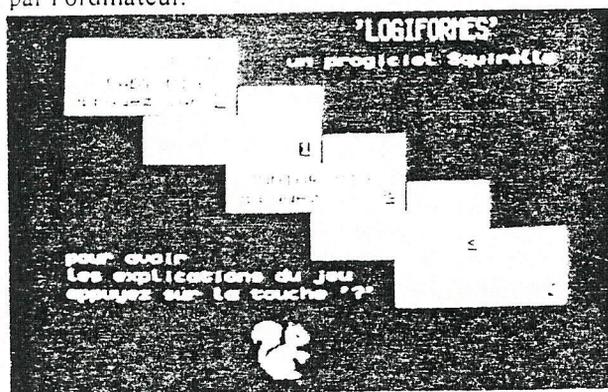
Un éditeur français de logiciels (denrée hélas trop rare...) s'est attaqué au marché prometteur des didacticiels, c'est-à-dire des programmes d'enseignement. Destiné principalement aux jeunes enfants dans leurs premières années de l'école primaire (encore que bien des adultes seront surpris de leur manque d'érudition...), les logiciels Squirelle existent sur plusieurs machines, dont l'Oric Atmos, le PC Junior et, bien sûr, le Lynx 96 Ko. Un grand bravo à cet éditeur performant, dont les programmes ne manqueront pas d'intéresser les lynxistes à la progéniture en âge scolaire...



JE SAIS COMPTER

Avec mon ami l'Écureuil porte comme titre ce programme faisant référence à l'animal fétiche de cet éditeur. Ce programme en deux parties (initiation et entraînement) et comportant plusieurs niveaux (de 0 à 9) permet aux jeunes enfants de se familiariser avec les quatre opérations, tout en s'amusant grâce à l'utilisation des possibilités graphiques et sonores des ordinateurs. L'initiation utilise une représentation visuelle de l'opération à effectuer (5 oiseaux plus 4 oiseaux égal...) et une animation graphique à base... d'écureuil pour indiquer les résultats.

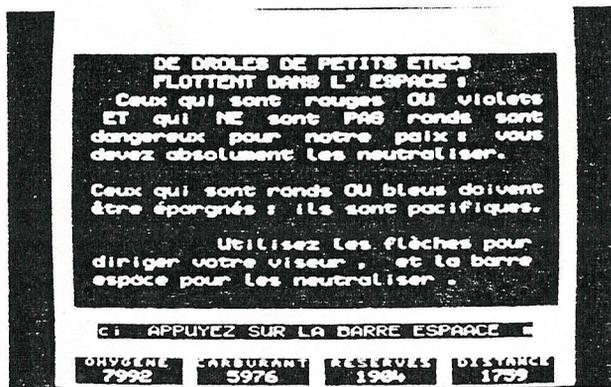
Dans un deuxième temps, FILE AUX TRÉSORS propose un jeu d'entraînement au calcul mental : votre équipier, qui saute de rocher à chaque bonne réponse parviendra-t-il à temps à l'Île au Trésor, avant que le trésor n'ait sombré dans l'océan ? Un seul moyen, répondre vite – et bien ! – aux opérations posées par l'ordinateur.



EN ROUTE VERS LA GALAXIE CE2

Parions que plus d'un parent, réfractaire aux beautés des maths modernes, ne se prenne au jeu de cette « Galaxie CE2 ». Squirelle ne propose pas moins, en effet, d'une initiation aux lois et règles de la théorie des ensembles, les fonctions logiques ET, OU et ensemblistes, INCLUSIONS, INTERSECTION, RÉUNION, APPARTENANCE sont expliquées simplement et sous forme ludique. Aux commandes d'un vaisseau spatial, l'utilisateur est en route vers la galaxie CE2. Aux questions de l'ordinateur, il faut répondre parfaitement pour se rapprocher du but. A chaque mauvaise réponse, une petite explication apparaît, et permet de repartir vers la galaxie.

Les qualités des réponses sont jugées en fonction de quatre compteurs du vaisseau : l'oxygène, qui mesure la vitesse de compréhension, le carburant, qui diminue à chaque hésitation, les réserves, utilisées à chaque mauvaise réponse et la distance qui diminue à chaque bonne réponse. Avec ses 6 niveaux de jeu, ce logiciel permet à chacun de se familiariser avec les notions abstraites des mathématiques modernes.



LOGIFORME

Ce programme teste la logique visuelle de l'enfant selon une méthode simple et efficace : l'ordinateur choisit et dispose aléatoirement de 3 à 6 animaux et l'enfant doit les retrouver et les remplacer dans le bon ordre. Un bon entraînement de la mémoire visuelle, qui convient à tous, à partir des classes maternelles.

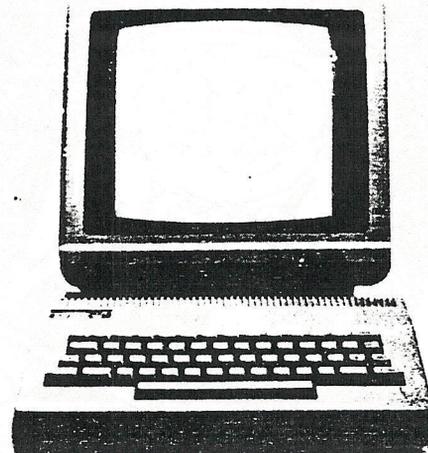
NEWS

MONITEUR VERT PHILIPS TP2000, 1.100 F T.T.C. (12")

Comme de nombreux utilisateurs de micro-ordinateurs familiaux, vous utilisez votre téléviseur comme écran : image parfaite (grâce à la prise Peritel) et en couleur, rien à redire... ou presque, du moins tant qu'une émission télévisée passionnante est programmée juste au moment où le "bug" récalcitrant vient d'être détecté.

Pour éviter un drame familial, vous rendez le téléviseur à sa destination première, et vous perdez complètement le fil de vos brillantes déductions (dans le meilleur des cas, sinon, vous ne cédez pas le drame éclate, etc...).

Pour un investissement réduit (1.100 F T.T.C.), vous auriez pu éviter le pire ! C'est en effet le premier prix pour un moniteur monochrome de marque Philips. Dommage de s'en priver ! D'autant plus que, si votre micro-ordinateur est un Lynx (heureux homme), un cordon d'interface vous permet d'obtenir sept nuances de vert à l'affichage.



DEUX NOUVEAUX LOGICIELS CHEZ SEGIMEX

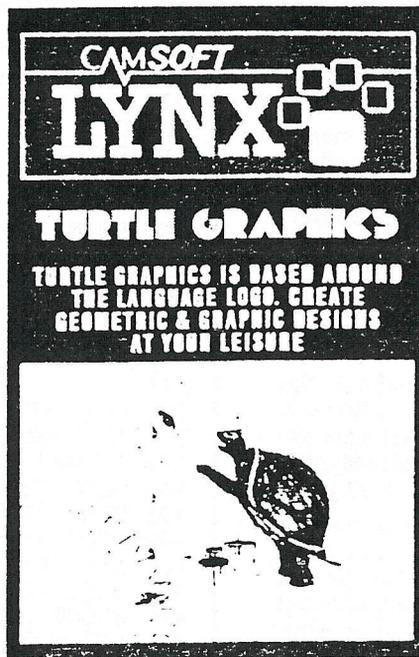
PANIC

Parmi les derniers arrivages d'Angleterre, deux logiciels ont retenu notre attention. Le premier, PANIC, est un classique des jeux d'arcades sur micro-ordinateur, célèbre sur Apple, Commodore, Spectrum, etc. Le Lynx ne pouvait pas ne pas l'inscrire dans sa ludothèque. Il s'agit bien entendu de PANIC. Le scénario du jeu est très simple, et tient en un seul mot : "survivre".

Le joueur se déplace en effet dans un labyrinthe d'échafaudages, de passerelles et d'échelles. Poursuivi par des monstres visiblement animés de mauvaises intentions, il doit les éviter aussi longtemps que possible. Pour se défendre, une seule arme, une pelle, avec laquelle il est possible de creuser des trous dans lesquels les monstres vont tomber, puis de les reboucher pour les ensevelir. Mais attention, pas question de creuser si le monstre est trop près. PANIC se révèle donc à la fois un jeu de réflexes et de stratégie, qui vous promet de bons moments.

TURTLE GRAPHICS

Dans un esprit beaucoup plus sérieux, TURTLE GRAPHICS permet de réaliser des dessins à l'écran, en pilotant une tortue (le curseur à l'écran) par des ordres simples : FORWARD 80 par exemple, la fera avancer de 80 pixels dans sa direction, alors que LEFT 90 lui fera effectuer un quart de tour vers la gauche. Inspiré du célèbre langage Logo du docteur Seymour Papert, TURTLE GRAPHICS constitue une excellente initiation au dialogue avec l'ordinateur, et convient plus particulièrement aux jeunes.



GRANADA & COLLINS CHEZ OBJECTIF MICRO : LA MICRO-INFORMATIQUE EN V.O.

Les anglophones distingués et microphiles avertis de tout bord peuvent se réjouir. Le catalogue complet des ouvrages consacrés à la micro-informatique des éditions britanniques Granada & Collins est enfin disponible en France !

Ce catalogue recèle plusieurs dizaines de titres dont de nombreux "petits bijoux" jusqu'à présent introuvables en France : par exemple, le "Lynx Computing" de Ian Sinclair, qui s'avérera vite un compagnon indispensable à tous les Lynxistes, ou le "Z80 Machine code for Humans" de Alan Teotill et David Barrow, l'un des meilleurs ouvrages d'initiation à la programmation en langage machine.

Toutes nos félicitations à Segimex et à Objectif Micro pour cette initiative qu'apprécieront tous les "hobbyistes de la région parisienne".

Objectif Micro
101, Bd Haussmann
75008 Paris

PROGRAMME DU MOIS



ENTRAÎNEMENT AU CALCUL

En ce début d'année scolaire, le programme du mois, signé D. Chaugne trouvera son application dans de nombreux foyers. Il s'agit en effet d'un didacticiel (programme d'enseignement) destiné aux jeunes enfants, pour les guider dans les premiers pas d'apprentissage du calcul arithmétique.

Le principe en est très simple : l'enfant choisi entre deux opérations, l'addition et la multiplication. Le Lynx choisit deux nombres au hasard, entre 0 et 9 (le programme est destiné aux très jeunes enfants, mais rien n'empêche de le modifier pour compliquer un peu le jeu) et demande le résultat. Le résultat entré au clavier par l'enfant est comparé avec le bon résultat. L'enfant peut jouer autant de fois qu'il le désire. A la fin d'une partie, l'ordinateur calcule le ratio de bonnes réponses, et propose une note comprise entre 0 et 20, assortie d'un commentaire, comme "te félicite", ou "l'encourage à persévérer".

L'intérêt pédagogique de ce jeu est multiple, malgré la simplicité du problème. D'une part, l'attrait de la machine et l'aspect jeu font que l'enfant n'a pas l'impression de travailler. De plus, le fait que la machine tienne compte du résultat et le commente confère une vertu éducative supplémentaire.

La programmation de ce didacticiel n'est pas très difficile, la plus grande part du listing consistant en la gestion de l'affichage. Certaines modifications très simples peuvent être envisagées, comme changer la valeur maximale des nombres à calculer (lignes 1020 et 1030, 1520 et 1530) ou la nature des commentaires (lignes 475, 480, 482). La ligne 490 affiche le logo Lynx, dont les caractères graphiques figurent en mémoire morte.

Sur le même principe, des opérations plus complexes peuvent être envisagées, comme des additions à trois chiffres ou des soustractions. Attention cependant à ce que le résultat ne soit pas négatif, pour ne pas poser de problèmes trop difficiles. De même, les divisions sont à éviter puisque les résultats risquent fort de ne pas tomber juste.

```

1 REM *****
2 REM *** ENTRAINEMENT AU CALCUL ***
3 REM *****
4 REM *VERSION ADAPTEE AU LYNX 48 Ko *
5 REM *****
6 REM * PAR D. CHAUGNE *
7 REM *****
8 REM -----INTRODUCTION -----
10 TEXT
20 PROTECT 0
30 PAPER RED
40 INK BLUE
50 LET a=0
60 LET b=0
70 CLS
80 PRINT @ 50,70:CHR$(243):CHR$(244):CH
R$(245):CHR$(246):CHR$(247):CHR$(248):CH
R$(249)
90 PRINT @ 40,85:"VA T'AIDER"
100 PRINT @ 39,100:"A APPRENDRE"
110 PRINT @ 40,115:"A CALCULER"
120 PRINT @ 25,160:"TU ES PRET ? (Oui o
u Non )"
130 PRINT @ 25,180:"Appuie sur la touch
e D ou N "
140 LET L=GETN
150 IF L=79 THEN GOTO 170
160 GOTO 120
162 REM **** FIN DE L'INTRODUCTION ***
165 REM ---- CHOIX DE L'OPERATION ----
170 CLS
180 PRINT @ 38,70:"TU PEUX CHOISIR"
190 PRINT @ 34,85:"ENTRE DEUX OPERATION
S : "
200 PRINT @ 33,100; "* LA MULTIPLICATION
(M) "
210 PRINT @ 33,115; "* L'ADDITION .....
(A) "
220 PRINT @ 25,180:"Appuie sur la touch
e M ou A "
230 LET M=GETN
240 IF M=77 THEN GOTO 290
250 IF M=65 THEN GOTO 270
260 GOTO 230
270 LET X$="+",Y=1,Z=0
280 GOTO 300
290 LET X$="*",Y=2,Z=1
300 CLS
310 PRINT @ 38,70:"LORSQUE LE LYNX"
320 PRINT @ 38,85:"TE DEMANDERA : "
330 PRINT @ 40,100;"3":X$;"7 = ? "
340 PRINT @ 38,115;"ALORS TAPES : "
350 PRINT @ 32,130;"SUR ";Y;" PUIS ";Z;
" SUR LE CLAVIER"
360 PRINT @ 36,145;"PUIS SUR [ RETURN ]
"
370 PRINT @ 24,200:"TU ES PRET ? (O ou
N) "
380 LET N=GETN
385 IF M=65 AND N=79 THEN GOTO 420
390 IF M=77 AND N=79 THEN GOTO 405
400 GOTO 380
401 REM ****FIN DU CHOIX DE L'OP ****
402 REM -----MULTIPLICATION-----
405 LET X$="*"
408 GOSUB 500
410 GOSUB 1000
411 LET M=77

```

```

415 GOTO 435
416 REM ****FIN DE MULTIPLICATION****
417 REM ----- ADDITION -----
420 LET X$="+"
422 GOSUB 500
425 GOSUB 1500
426 LET M=65
430 REM ***** FIN D'ADDITION *****
432 REM --- VARIANTES D'OPERATION ---
435 PRINT @ 40,240;"VEUX TU CONTINUER ?
(O ou N) "
440 LET O=GETN
442 IF O=78 THEN GOTO 457
445 IF M=65 AND O=79 THEN GOTO 425
450 IF M=77 AND O=79 THEN GOTO 410
455 GOTO 440
457 PRINT @ 10,240;"VEUX TU CHANGER D'O
PERATION ? (O ou N)"
460 LET P=GETN
462 IF M=65 AND P=79 THEN GOTO 405
465 IF M=77 AND P=79 THEN GOTO 420
467 IF P=78 THEN GOTO 472
469 GOTO 460
470 REM ***** FIN DE VARIANTES *****
471 REM ----- CONCLUSION -----
472 LET Q=INT((20*b)/a)
475 IF Q<14 THEN LET R$="T'ENCOURAGE",S
$="A PERSEVERER "
480 IF Q<7 THEN LET R$="TE CONSEILLE",S
$="DE TRAVAILLER "
482 IF Q>=14 THEN LET R$="TE FELICITE",
S$=" "
485 TEXT
486 PROTECT 0
487 PAPER RED
488 INK BLUE
489 CLS
490 PRINT @ 50,70;CHR$(243);CHR$(244);C
HR$(245);CHR$(246);CHR$(247);CHR$(248);C
HR$(249)
492 PRINT @ 40,85;R$
495 PRINT @ 40,100;S$
497 PRINT @ 31,200;"TA MOYENNE EST DE "
:Q;" SUR 20"
499 END
500 REM --- MISE EN PLACE DU TABLEAU ---
505 INK GREEN
510 TEXT
515 PROTECT 0
518 CLS
519 PRINT @ 17,15;X$
520 LET C=17
530 FOR I=1 TO 9
540 LET C=C+10
550 PRINT @ C,15:I
560 NEXT I
570 LET D=15
580 FOR I=1 TO 9
590 LET D=D+20
600 PRINT @ 17,D:I
610 NEXT I
620 MOVE 28,30
630 DRAW 28,210
640 LET A=28
650 FOR I=1 TO 10
660 LET A=A+20
670 MOVE A,210

```

```

680 DRAW A,10
690 NEXT I
700 DRAW 48,10
710 LET B=10
720 FOR I=1 TO 10
730 LET B=B+20
740 MOVE 228,B
750 DRAW 28,B
760 NEXT I
765 PROTECT 4
767 INK 2
770 RETURN
780 REM ***** FIN DE TABLEAU *****
1000 REM --- CALCUL DE MULTIPLICATION ---
1010 FOR I=1 TO 5
1020 LET E=RAND(8)+1
1030 LET F=RAND(8)+1
1040 LET G=E*F
1045 PRINT @ 40,240;"
"
1050 PRINT @ 40,240;E;"*";F;"=" ";
1060 INPUT H
1065 LET a=a+1
1070 IF G=H THEN GOTO 1110
1080 BEEP 3000,30,63
1090 PRINT @ 40,225;"TU T'ES TROMPE ,
RECOMMANCE !"
1100 GOTO 1045
1110 LET b=b+1
1120 PRINT @ 40,225;"
"
1130 PRINT @ 40,240;"
"
1140 GOSUB 2000
1150 NEXT I
1160 RETURN
1170 REM * FIN CALCUL MULTIPLICATION *
1500 REM --- CALCUL D'ADDITION ---
1510 FOR I=1 TO 5
1520 LET E=RAND(8)+1
1530 LET F=RAND(8)+1
1540 LET G=E+F
1545 PRINT @ 40,240;"
"
1550 PRINT @ 40,240;E;"+";F;"=" ";
1560 INPUT H
1570 LET a=a+1
1580 IF G=H THEN GOTO 1620
1590 BEEP 3000,30,63
1600 PRINT @ 40,225;"TU T'ES TROMPE ,
RECOMMANCE !"
1610 GOTO 1545
1620 LET b=b+1
1630 PRINT @ 40,225;"
"
1640 PRINT @ 40,240;"
"
1650 GOSUB 2000
1660 NEXT I
1670 RETURN
1680 REM *** FIN CALCUL ADDITION ***
2000 REM --- AFFICHAGE RESULTAT ---
2010 LET J=17+(E*10)
2020 LET K=15+(F*20)
2030 PRINT @ J,K;G
2040 RETURN
2050 REM ** FIN AFFICHAGE RESULTAT **

```



CALENDRIER PERPÉTUEL

Dans le dernier numéro de l'Œil de Lynx, nous vous avons présenté un programme de calendrier perpétuel. Si vous maîtrisez un peu le Basic, voici une formule qui vous permettra de calculer un facteur F relatif à une date donnée sous la forme de trois variables : J (Jour), M (Mois) et A (Année complète). La différence entre deux facteurs relatifs à deux dates F (Date 1) - F (Date 2) permet de calculer le nombre de jours écoulés entre deux dates. De plus, la valeur $F \text{ MOD } 7$ (reste de la division de F par 7) permet de connaître le jour de la semaine correspondant à une date donnée :

0	Samedi
1	Dimanche
2	Lundi
3	Mardi
4	Mercredi
5	Jeudi
6	Vendredi

Cette formule, d'apparence assez complexe, tient compte des années bisextiles et de tous les cas particuliers du calendrier (mois de 28, 30 ou 31 jours) :

```
1000 DEFPROC FACTEUR
1010 LET F = 35*A + J + 31*(M-1)
+ M = 2*(INT(-3/4*(INT((A-1)
/100)))+(A-1) DIV 4) - (M 2)*(
(4*M + 23) DIV 10 - A DIV
4 + INT(3/4 * 5A DIV 100 + 1))
```

Attention aux parenthèses !!!

Cette formule peut servir de base à la plupart des programmes utilisant des dates, calendriers, biorythmes, etc... Le petit programme Basic suivant est un exemple simple d'application :

```
1 CLS
100 DIM A$(8) (7)
110 LET A$(0) = "SAMEDI",
A$(1) = "DIMANCHE",
A$(2) = "LUNDI",
A$(3) = "MARDI",
A$(4) = "MERCREDI",
A$(5) = "JEUDI",
A$(6) = "VENDREDI".
200 INPUT "DATE DE NAIS-
SANCE (JJ, MM, AAAA)" ;
J, M, A.
210 PROC FACTEUR
220 PRINT "VOUS ETES NE
UN" : A$(F MOD 7) ;
225 VDU 31, 31
230 LET F = F
240 INPUT "DATE ACTUELLE
(JJ, MM, AAAA)" ; J, M, A.
250 PROC FACTEUR
255 VDU 31
260 PRINT "AUJOURD'HUI" ;
A$(F MOD 7) ; " ; J ; "/" ; M ; "/" ; A
270 PRINT "VOUS AVEZ EXAC-
TEMENT" ; F-f ; "JOURS"
280 END
1000 .....
1010 .....
```

Le B.A. BA de l'ABC !

Les lynxistes lecteurs de la revue « encyclopédiste » l'ABC de l'informatique auront eu la désagréable surprise de lire dans le numéro 7 (page 137) que le programme « variante de Basic » ne « fonctionnera pas sur Lynx. »

Amis Lynxistes, ne vous découragez pas ! Ce programme de tri de noms par ordre alphabétique peut facilement s'adapter à votre ordinateur préféré. Il suffit de réaliser les quelques modifications suivantes :

A la place de :

```
310 DIM A$(N),
520 CH$ = MID$(A$, (X), P, 1)
523 IF CH$ = " "
THEN LET $ = P : LET P = L
S = P, P = L
550 LET F$ = F$ + " , "
```

Ecrire :

```
310 DIM A$(40), N
520 H$ = MID$(A$, (X), P, 1)
523 IF H$ = " " THEN LET
550 LET F$ = F$ + " , "
Et ajouter les deux lignes :
527 DIM F$(25)
1105 DIM T$(40)
```

Et signalons pour finir que la ligne :
1010 LET S = D
comporte certainement une erreur, la seule réponse cohérente étant :
1010 LET S = P

J. Stasny

PROGRAMME DE CONVERSION BINAIRE/HEXADECIMAL/DECIMAL

Cette routine Basic permet de donner automatiquement la conversion d'un nombre décimal quelconque entré au clavier sous une forme quelconque (directement ou comme résultat d'une opération) dans les trois systèmes numériques employés en informatique. L'intérêt de cette routine se révèle surtout pour suivre en langage machine le résultat de calculs, et pour traiter aisément tous les problèmes mettant en jeu des codes ASCII.

```
100 REM CONVERSION EN
BINAIRE 16 BITS
```

```
110 CLS
120 CODE E5 06 10 CB 25 CB 14
3005 3E 31 CF 18 03 3E 30 CF 10 FO
3E 20 CF E1 CD 34 22 C9
130 INPUT A
140 CALL LCTN (120) , A
150 PRINT " " ; A
160 GOTO 130
```

GÉNÉRATEUR DE CARACTÈRES :

Pour les Lynxistes déjà chevronnés, voici un petit programme mixant à la fois Basic et langage machine, permettant de se familiariser avec les fonctions de la ROM génératrice de caractères.

```

100 CLS
110 PRINT 3, 120 ;
    "CARACTÈRE"
130 FOR I = LETTER (32)
    TO LETTER (127) STEP 10
140 LET C = (I-LETTER (32) )
    10 + 32
150 PRINT 36, 120 ; C ;
160 VDU 23
170 FOR J = 0 TO 9
180 CALL LCTN (230),
    PEEK (I+J)
190 VDU 31
200 NEXT J
210 LET A=GETN
220 NEXT I
230 CODE 06 08 CB 25 30 05 3E
    F2 CF 18 03 3E 2E CF 10
    F2 C9

```

Le programme machine de la ligne 230 est en fait un convertisseur décimal/binaire (sur 8 bits) où O (ASCII 30h) est remplacé par un point (ASCII 2Eh) et I (ASCII 31h) par un gris (ASCII F2h). Le nombre à convertir est transféré dans le registre HL lors de l'appel de la routine de la ligne 180.

Il est possible d'obtenir des résultats amusants en apportant les modifications suivantes :

- dans la ligne 230, remplacer 3E 2E (point) par 3E 20 (blanc)
- ajouter la ligne suivante :
145 POKE LCTN (230) + 7,C

Le gris est alors remplacé par le code ASCII du caractère en cours.

A LA CHASSE AUX "BUGS"

Comme tous les micro-ordinateurs, le lynx recèle, bien caché au fond de sa ROM, quelques "parasites" parfois gênants... En particulier, le clavier révèle parfois un comportement des plus fantaisistes. Pour le Lynxistes, O.D.L est parti à la chasse aux bugs, et à découvert un certain nombre d'originalités dans la routine de scrutation du clavier...

Un des problèmes majeurs rencontré par les lynxistes provient de l'action de la touche SHIFT LOCK, et son action sur l'affichage de certains caractères, comme le « @ » et le « / ».

La routine de scrutation du clavier est située à l'adresse 2907, pointée par la variable système 25092/93. Elle utilise trois tables qui donnent les codes des différents caractères, selon la position de la touche SHIFT LOCK et l'emploi de la touche CTRL. Les bugs proviennent de la matière dont est détectée la position de la touche SHIFT LOCK. La routine de scrutation du clavier considère en mode normal que cette touche est enfoncée. Dans le cas contraire (détecté en regardant la variable système en 25145, qui contient 0 lorsque la touche SHIFT LOCK est enfoncée, et 1 sinon), la routine ignore théoriquement tous les caractères dont le code est inférieur à 64 et altère l'état du bit 5 du code des autres caractères, ce qui revient à ajouter ou à soustraire 32 au code ASCII correspondant. Or le symbole « / » inscrit sur la touche « @ » ne peut être obtenue de cette manière, qui provoque en fait l'obtention du symbole « L », la différence entre les deux codes étant justement 32 ! La raison de l'altération de l'ordre d'entrée semble provenir en fait d'un bug dans la routine, qui ignore en fait tous les caractères de codes inférieurs à 65 (au lieu de 64!), alors que le code ASCII de « @ » est justement 64 !

Il est possible de changer cette routine du Lynx de façon à ce que cette erreur

soit corrigée, et qu'il soit possible d'obtenir les caractères correspondant aux codes ASCII standards directement à partir du clavier, sans passer par l'instruction CHRS du Basic...

La méthode utilisée consiste à « copier » la routine de scrutation du clavier dans la mémoire vive du Lynx, en changeant certaines adresses, puis à réécrire une nouvelle routine tenant mieux compte du cas où la touche SHIFT LOCK n'est pas enfoncée. Certaines tables de la routine initiale doivent également être modifiées. L'ordinateur sera capable de détecter la routine modifiée en changeant l'adresse contenue en 25092/93. Un petit programme en langage machine a été ajouté pour pousser la zone réservée aux programmes Basic à ce niveau ? Pour ce faire, il est nécessaire de changer les adresses de trois variables systèmes :

```

25082/83 Début de la zone Basic
25084/85 Fin de cette zone
25119/20 Fin de la zone des variables
/ Basic.

```

Le programme est bien évidemment entré grâce aux facilités du Moniteur intégré du Basic. Pour ceux qui ne sont pas familiers avec son utilisation, reportez vous au manuel, ou au cours d'initiation à l'assembleur.

H 6AA0 5C	SHIFT @ = \ (ASCII 92)
H 6AA1 5E	SHIFT [= ASCII 94
H 6AA6 7F	SHIFT] = ASCII 127
H 6AA5 5F	SHIFT DEL = _ (ASCII 95)
H 6AA7 7C	SHIFT ← = ← (ASCII 124)
H 6AAA 7B	SHIFT → = → (ASCII 123)
.B 6A4C	This will set the keyboard routine into operation.
D 694D 6AEB 6A4C "<Program name>"	This will save the routine on tape. Using HL,DAD "<program name>" will load the program into the correct area of memory and being it was saved with an auto entry address, it will automatically set up the new routine.
N.B. The ASCII codes 94 & 127 combined, save the copyright symbol.	
C 09BD 694D 019E	Copy the keyboard routine to the RAM.
C 0B5B 6A42 0A	
H 6985 06 6A	6A06 - Address of SHIFT LOCK table
H 6990 0F 6A	6A0F - Address of SHIFT LOCK Shift Table
H 6AB2 4D 69	Start of the new routine in memory
H 6A46 AB 6A	Call address modification
H 69AB CD D5 69 18 03	Alteration in call to SHIFT LOCK-off routine
H 69D5 EB 21 39 62 CB 46 CB FE 41	New SHIFT LOCK-off routine
D8 01 69 20 FE 5B 38 0C FE	
61 D8 FE 7B D0 EB AF ED 42	
18 02 EB D9 7E EB C9	
H 6A4C 21 42 6A 22 04 62 21 EB 6A	Routine to bring the new keyboard routine
34 0D 23 34 80 22 FA 61 22	into operation and set up the BASIC program
FC 61 23 22 1F 62 C9	above this.
H 6A38 5B	Set up left hand bracket key [(ASCII 91)
H 6A3D 5D	Set up right hand bracket key] (ASCII 93)
H 6A73 7D	SHIFT f = f (ASCII 125)
H 6A74 7E	SHIFT b = b (ASCII 126)
H 6A99 60	SHIFT o = o (ASCII 96)

TABLE DE CONVERSION

DEC	BINAIRE	HEX	DEC	BINAIRE	HEX
0	00000000	00	64	01000000	40
1	00000001	01	65	01000001	41
2	00000010	02	66	01000010	42
3	00000011	03	67	01000011	43
4	00000100	04	68	01000100	44
5	00000101	05	69	01000101	45
6	00000110	06	70	01000110	46
7	00000111	07	71	01000111	47
8	00001000	08	72	01001000	48
9	00001001	09	73	01001001	49
10	00001010	0A	74	01001010	4A
11	00001011	0B	75	01001011	4B
12	00001100	0C	76	01001100	4C
13	00001101	0D	77	01001101	4D
14	00001110	0E	78	01001110	4E
15	00001111	0F	79	01001111	4F
16	00010000	10	80	01010000	50
17	00010001	11	81	01010001	51
18	00010010	12	82	01010010	52
19	00010011	13	83	01010011	53
20	00010100	14	84	01010100	54
21	00010101	15	85	01010101	55
22	00010110	16	86	01010110	56
23	00010111	17	87	01010111	57
24	00011000	18	88	01011000	58
25	00011001	19	89	01011001	59
26	00011010	1A	90	01011010	5A
27	00011011	1B	91	01011011	5B
28	00011100	1C	92	01011100	5C
29	00011101	1D	93	01011101	5D
30	00011110	1E	94	01011110	5E
31	00011111	1F	95	01011111	5F
32	00100000	20	96	01100000	60
33	00100001	21	97	01100001	61
34	00100010	22	98	01100010	62
35	00100011	23	99	01100011	63
36	00100100	24	100	01100100	64
37	00100101	25	101	01100101	65
38	00100110	26	102	01100110	66
39	00100111	27	103	01100111	67
40	00101000	28	104	01101000	68
41	00101001	29	105	01101001	69
42	00101010	2A	106	01101010	6A
43	00101011	2B	107	01101011	6B
44	00101100	2C	108	01101100	6C
45	00101101	2D	109	01101101	6D
46	00101110	2E	110	01101110	6E
47	00101111	2F	111	01101111	6F
48	00110000	30	112	01110000	70
49	00110001	31	113	01110001	71
50	00110010	32	114	01110010	72
51	00110011	33	115	01110011	73
52	00110100	34	116	01110100	74
53	00110101	35	117	01110101	75
54	00110110	36	118	01110110	76
55	00110111	37	119	01110111	77
56	00111000	38	120	01111000	78
57	00111001	39	121	01111001	79
58	00111010	3A	122	01111010	7A
59	00111011	3B	123	01111011	7B
60	00111100	3C	124	01111100	7C
61	00111101	3D	125	01111101	7D
62	00111110	3E	126	01111110	7E
63	00111111	3F	127	01111111	7F

IMPORDINATEUR

En cette période hivernale, pourquoi ne pas se remonter le moral en songeant à sa prochaine déclaration d'impôts? Dans sa grande campagne de lutte contre la morosité, Œil de Lynx vous offre un programme de calcul automatique. Conçu avec les réglementations de 1983, il demandera sans doute quelques modifications (lignes 430 à 560 et 1040 à 1165) pour s'adapter aux nouvelles réglementations. En particulier, le 1 % exceptionnel au titre exceptionnel va peut-être être supprimé ! Enfin une bonne nouvelle...

```

100 REM CALCUL DE L'IMPOT 1983.
101 CLS
110 REM -----
115 PAPER 0
116 INK 2
117 VDU 25,18
120 PRINT " PRECISER VOTRE NOM "
125 INPUT N$
130 PRINT " PRECISER VOTRE SITUATION DE
FAMILLE"
140 PRINT " M pour MARIE, C pour CELIBAT
AIRE"
150 INPUT S$
170 PRINT " COMBIEN D'ENFANTS A CHARGE "
175 IF S$="M" THEN GOTO 198
176 INPUT A
180 IF A=0 THEN LET Q=1
190 IF A>0 THEN LET Q=0.5+A
195 GOTO 220
198 INPUT A
200 IF A=0 THEN LET Q=2
210 IF A>0 THEN LET Q=2+A/2
220 PRINT " PRECISER LE MONTANT DES SALA
IRES "
230 PRINT " SALAIRE No 1 ? "
240 INPUT S
250 PRINT " SALAIRE No 2 ?"
260 INPUT F
270 PRINT " AUTRES SALAIRES SANS OUBLIER
LES AVANTAGES EN NATURE "
275 INPUT Z
280 LET X=S+F+Z
281 INPUT " BENEFICE NON COMMERCIAUX APR
ES ABATTEMENTS ";R
285 CLS
290 PRINT "VOS REVENUS BRUT SONT :";X+R
295 PAUSE 100000
296 CLS
300 LET I=(X*0.9*0.8)+R
310 PRINT "PRECISER VOS REVENUS MOBILIER
S"
320 PRINT "DIVIDENDES NETS "
325 INPUT D
330 PRINT "AVOIRS FISCAUX "
335 INPUT G
    
```

```

340 LET M=(D+G)-3000
350 IF M<0 THEN LET M=0
360 LET H=I+M
365 LET Y=H/100
370 PRINT "VOUS AVEZ LA POSSIBILITE DE D
EDUIRE :      ";Y;"F. EN DONNS AUX DEUV
RES"
375 LET K=H-Y
376 VDU 24,18
380 PRINT "VOTRE REVENU NET EST ALORS : "
;K;"F.      "
385 PAUSE 50000
386 VDU 18,25
400 REM CALCUL DE L'IMPOT
410 REM -----
420 LET I=K/Q
430 IF I<13770 THEN GOTO 1030
440 IF I<=14390 THEN GOTO 1040
450 IF I<=17070 THEN GOTO 1050
460 IF I<=26990 THEN GOTO 1060
470 IF I<=34700 THEN GOTO 1070
480 IF I<=43610 THEN GOTO 1080
490 IF I<=52760 THEN GOTO 1090
500 IF I<=60870 THEN GOTO 1100
510 IF I<=101430 THEN GOTO 1110
520 IF I<=139500 THEN GOTO 1120
530 IF I<=165010 THEN GOTO 1130
540 IF I<=187700 THEN GOTO 1140
550 IF I<=212750 THEN GOTO 1150
560 IF I>212750 THEN GOTO 1160
1030 LET J=0
1035 PRINT
1036 PRINT " VOUS N'ALLEZ DONC PAS PAYER
D'IMPOT SUR LE REVENU ."
1037 PRINT " OU BIEN VOUS ETES TRES PAUV
RE ET VOUS ETES A PLAINDRE ."
1038 PRINT "OU VOUS TROMPEZ LE FISC, ATT
ENTION VOUS RISQUEZ DES POURSUITES !!"
1039 END
1040 LET J=(0.05*I)-689
1045 GOTO 2000
1050 LET J=(0.1*I)-1408
1055 GOTO 2000
1060 LET J=(0.15*I)-2262
1065 GOTO 2000
1070 LET J=(0.2*I)-3611
1075 GOTO 2000
1080 LET J=(0.25*I)-5346
1085 GOTO 2000
1090 LET J=(0.3*I)-7527
1095 GOTO 2000
1100 LET J=(0.35*I)-10165
1105 GOTO 2000
1110 LET J=(0.4*I)-13208
1115 GOTO 2000
1120 LET J=(0.45*I)-18279
1125 GOTO 2000
1130 LET J=(0.5*I)-25254
1135 GOTO 2000
1140 LET J=(0.55*I)-33505
1145 GOTO 2000
1150 LET J=(0.6*I)-42890
1155 GOTO 2000
1160 LET J=(0.65*I)-53527
1165 GOTO 2000
2000 REM CALCUL DES DEDUCTIONS
2010 CLS
2020 PRINT "PRCISER VOS DEDUCTIONS"
2030 INPUT "MONTANT DES INVESTISSEMENTS
EN ACTIONS DU C.E.A ";B
2040 LET C=0.25*B
2050 IF S$="M" AND B>=14000 THEN LET C=
14000*0.25
2060 IF S$="C" AND B>=7000 THEN LET C=7
000*0.25
2065 IF S$="M" THEN LET c=0.25*14000
2066 IF S$="C" THEN LET c=0.25*7000
2070 INPUT "MONTANT DES INTERETS ET DEPE
NSES DE RAVALEMENT ";i
2080 LET j=i*0.2
2090 LET m=(9000+(A*1500))*0.2
2100 IF j>=m THEN LET j=m
2110 INPUT "DEPENSES POUR L'ECONOMIE D'E
NERGIE";a
2120 LET n=(8000+(A*1000))*0.25
2130 LET b=a*0.25
2140 IF b>=n THEN LET b=n
2150 INPUT " MONTANT DE L'ASSURANCE-VIE
";v
2160 LET w=v*0.2
2170 LET p=(4000+(A*1000))*0.2
2180 IF w>=p THEN LET w=p
2182 INPUT "AUTRES DEDUCTIONS ";l
2190 LET d=w+b+j+C+l
3000 LET x=J*Q
3010 IF x<20000 THEN GOTO 4000
3020 IF x<=25000 THEN GOTO 4100
3030 IF x<=30000 THEN GOTO 4200
3040 IF x>30000 THEN GOTO 4300
4000 LET g=x-d
4010 GOTO 5000
4100 LET g=(x*1.25)-5000-d
4110 GOTO 5000
4200 LET g=(1.05*x)-d
4210 GOTO 5000
4300 LET g=(x*1.08)-d
4310 GOTO 5000
5000 PRINT "DEDUCTIONS ENREGISTREES AU T
ITRE DE : "
5001 PRINT "ACTIONS :";C;" MAXI :";c
5002 PRINT "INTERETS :";j;" MAXI : ";m
5003 PRINT "ECONOMIE D'ENERGIE :";b;" MA
XI :";n
5004 PRINT "ASSURANCE-VIE :";w;" MAXI : "
;p
5004.5 PRINT " AUTRES DEDUCTIONS ";l
5005 PRINT
5006 PRINT " DEDUCTIONS TOTALE : ";d
5006.5 PRINT " DEDUCTIONS MAXIMUM POSSIB
LE ";c+p+n+m
5007 PRINT "NOMBRE DE PARTS :";Q
5007.5 INPUT " appuyez sur C pour contin
uer";C$
5007.6 CLS
5008 VDU 24,18
5008.5 PRINT "-----"
"
5009 PRINT " ";N$
5010 PRINT " VOTRE IMPOT EST : "
5015 PRINT
5100 PRINT " ";g-G+(K/100);" "
5105 PRINT " CE CHIFFRE COMPREND LE 1 %
AU TITRE DE CONTRIBUTION SOCIALE SOIT : "
;K/100;" "
5110 PRINT
5120 VDU 25,7,7,7

```

INITIATION

INITIATION A L'ASSEMBLEUR II

Dans la première partie de cet article nous avons vu les différents modes de dialogue direct avec l'ordinateur, en binaire et en hexadécimal. Le petit programme hexa suivant, dont nous ne savons d'ailleurs pas la signification, ne demande plus qu'à entrer dans l'ordinateur. Mais où ? La plupart des programmeurs en langage machine sont confrontés avec cette difficulté. Théoriquement, ce programme peut commencer n'importe où dans la mémoire vive de l'ordinateur. Mais comment être sûr qu'il y a de la place ? Avant de voir en détail ce problème, essayons d'abord de le mettre quelque part, en 9000H par exemple.

En 9000H par exemple :

Pour ce faire, tapons MON (suivi de RETURN) pour appeler le moniteur du Lynx. Ensuite tapons M9000 (RETURN) pour commencer à charger (commande M du moniteur) à l'adresse choisie. Sans tenir compte des bizarreries de l'affichage, entrons notre programme à commencer par le premier code, c'est-à-dire 21. N'oublions pas de taper RETURN après chaque code. Lorsque le dernier code est chargé dans l'ordinateur, tapons sur le STOP pour sortir de la commande M du moniteur. Nous sommes maintenant prêts pour lancer notre programme, comme par un RUN en Basic. Pour faire cela, il est nécessaire que le PC (program counter) soit pointé sur le premier octet de notre programme. Par chance, le moniteur a une commande qui réalise cela automatiquement. Il s'agit de la commande G. Dès que l'on entre au clavier G9000, le programme fonctionne. Amusant n'est-ce pas. Essayez plusieurs fois. Mais qu'à donc compris notre ordinateur ?

I LA PROGRAMMATION HEXA DÉCIMALE

Le programme commence par l'octet 21. Ce code indique à l'ordinateur de prendre les deux octets suivants, c'est-à-dire 5B et 62 et de les ranger dans les registres H et L. Hélas, ce n'est pas tout à fait aussi simple, car les octets sont appelés en ordre inverse : le registre contient 62 alors que le registre L contient 5B. La paire HL contient donc, dans l'ordre, 625B. HL sait (il a bien de la chance) que 625B est l'adresse ou est enregistrée la couleur INK. Le code suivant, 7E place dans le registre A le contenu de l'adresse contenue dans HL, c'est-à-dire 625B. Le registre A contient donc la couleur INK. Le code 23 incrémente (ajoute I) au contenu de HL, qui devient donc 625C. Or 625C est justement l'adresse de la couleur PAPER. Le code 46 indique au microprocesseur de charger le registre B avec le contenu de l'adresse 625C. Maintenant, le registre A contient donc la couleur INK du texte et le registre B la couleur PAPER du fond. Quand au registre HL, il contient désormais l'adresse 625C. Mais le code 77 charge le registre A avec le contenu de l'adresse 625C, c'est-à-dire la couleur PAPER, qui remplace la couleur INK. Le code suivant décrémente (retire I) le

contenu du registre HL, qui redevient 625B. Le code 70 effectue la même opération avec le registre B, qui pointe maintenant la couleur INK. Le dernier code C9 est une instruction qui renvoie au moniteur. Ce programme en langage machine est en fait l'équivalent de la routine VDU 18 d'inversion vidéo.

II LES INSTRUCTIONS CODE, CALL, LCTN.

Une autre manière de charger un tel programme en langage machine est d'utiliser les instructions Basic CODE, CALL, LCTN. La méthode est encore plus simple :

```
10 CODE 21 5B 62 7E 23 46 77 2B 70  
C9 20 CALL LTCN (10)
```

Un RUN classique lancera le programme en langage machine, et le dernier code renverra dans ce cas à la suite du programme Basic. Il est ainsi possible d'intégrer des routines en langage machine dans un programme Basic.

Mais vous vous posez sans doute quelques questions. En particulier, vous vous demandez peut-être comment savoir quels sont les registres à utiliser ? La seule réponse est l'expérience... ce qui n'est pas d'un grand secours quand on

ASSEMBLEUR (II)

hésite devant 22 situations ! En fait, lorsque l'on a compris que le Z80 lit les codes dans la mémoire, et exécute les instructions qu'ils contiennent, vous avez gagné la moitié de la bataille. Après avoir essayé de comprendre comment fonctionnent quelques programmes simples, lancez-vous sans crainte. Mais, avant d'acheter un livre vous donnant la signification de tous les codes, patientez un peu : certains de ces codes sont dangereux pour la machine lorsqu'ils sont mal utilisés, et il vaut mieux les ignorer au début.

La deuxième question la plus souvent posée est comment savoir où envoyer le programme. Au moins, la réponse vous apportera un peu plus de réconfort que la précédente. La mémoire du Lynx est disposée d'une manière immuable, et connue. Il suffit donc de consulter la carte mémoire (memory map en anglais) du Lynx pour savoir où il y a des places de libres.

Lorsque vous saurez où se trouve le Basic, où sont rangées les variables et vos programmes, il ne vous restera plus qu'à choisir un coin de libre.

Pratiquement, la mémoire interne du Lynx est ainsi répartie :

```

FFFF .....
  Extention
A000 .....
9FFF RAM utilisateur
6000 .....
5FFF EPROM libre
4000 .....
3FFF Basic
0000 .....
    
```

La RAM utilisateur correspond à la place en mémoire vive accessible à l'utilisateur, ce qui signifie que la place où l'on va « ranger » les programmes en langage machine se situera forcément entre l'adresse 6000 et l'adresse 00 9FFF.

Cependant, toutes les places ne sont pas bonnes à prendre. Le Lynx utilise en effet une grande partie de cette mémoire vive pour stocker un certain nombre de renseignements (la liste des variables Basic, par exemple, et un programme machine mal placé risquerait d'effacer des données importantes !

Regardons donc de plus près la « memory map » de la zone comprise entre les adresses 6000 et 9FFF :

```

9FFF .....
  HIMEM
  .....
  Place occupée par la pile (variable)
9000
  Stockage des variables chaînes
  et vecteurs, débutant après la fin
  du programme Basic
  .....
  Le programme Basic finit quel-
  que par par là...
694D Le programme Basic commence
  ici
  .....
68E4 Pointeurs des variables $
6814 Pointeurs des variables tableaux
670C Pointeurs des variables scalaires
    
```

Cette carte montre un certain nombre d'endroits réservés de la mémoire vive du Lynx. On voit que quelque chose appelée la « pile » (nous y reviendrons) utilise des adresses inférieures à 9FFF et qu'un programme Basic débute en 694D. Le reste de la mémoire est utilisée comme « zone de travail » pour le Basic, sous une forme ou sous une autres.

Si on utilise que du langage machine, le programme peut-être chargé n'importe où dans cette zone, et, en tout état de cause, l'adresse 9000 est une aussi bonne adresse... que n'importe où ailleurs !

Par contre, si l'on souhaite utiliser des routines en langage machine conjointement avec un programme Basic, il est nécessaire de s'assurer qu'aucune interférence ne va poser de problèmes : si l'adresse à laquelle vous stockez vos routines machines est 68E4, vous aurez sans doute quelques difficultés à retrouver vos variables chaînes de caractères...

La manière la plus sûre est d'utiliser l'instruction RESERVE qui déplace le marqueur HIMEM vers le bas. Or cette marque indique au Basic combien de place disponible il lui reste. En déplaçant artificiellement cette marque, on peut réserver (d'où le nom) la place nécessaire au programme en langage machine.

Par exemple,
RESERVE HIMEM -300
 déplace la marque HIMEM de 300 octets. La carte mémoire apparaîtra comme ceci

```

9FFF .....
  300 octets que le Basic ne « voit » pas
9ECC HIMEM
  .....
  Début de la pile
  .....
    
```

Le programme en langage machine peut désormais être stocké en 9ECC, sans que le Basic ne puisse s'y installer, puisque ces 300 octets lui sont invisibles !

Nous allons mettre notre nouveau savoir immédiatement en pratique. Le programme suivant à pour but - modeste - d'afficher sur l'écran le message suivant
DISPLAY YOUR NAME

Et nous mettons le doigt sur un point important : il est inutile de ré-inventer la roue à chaque nouveau programme ! Il y a dans le langage Basic une routine qui affiche des caractères à l'écran, et qui est en particulier utilisée par l'instruction PRINT. Cette routine, située à l'adresse 06A4, à pour effet d'afficher à l'écran le caractère dont le code ASCII est stocké dans le registre a. Cette adresse peut être trouvée en 6200H et peut être appelée par l'instruction

PRINT / DPEEK (&6200)

Le déroulement logique du programme est donc :

1. Charger HL avec le premier octet à lire moins un
2. Incrémenter HL pour pointer sur l'octet suivant
3. Charger A avec ce que pointe HL
4. Appeler la routine d'affichage
5. Tester la fin du message, et retourner au point 2. si l'on n'a pas fini.
6. RETURN.

Nous savons maintenant réaliser l'étape N° 1 : pour charger HL avec une adresse on utilise le code machine 21 suivi de l'adresse choisie. La deuxième étape est aussi simple en utilisant le code 23. La troisième étape a déjà été employée dans le programme précédent, et correspond au code 7E.

Par contre, nous n'avons jamais essayé d'appeler une routine déjà existante. Ce n'est pas très compliqué. Le code correspondant à CALL est CD, et il doit être suivi de l'adresse sur 2 octets à laquelle se situe la routine. En hexadécimal, celle qui nous intéresse est située en 06A4. Attention, le code CD utilise les adresses « à l'envers » et l'instruction correcte est : CDA406. Rappelons nous aussi que l'étape N° 6 utilise simplement le code C9. Il ne nous reste plus que l'étape N° 5, qui nous demandera l'apprentissage d'un nouveau code ! que nous apprendrons la prochaine fois...

SPÉCIAL CALENDRIER

CALENDRIER

Les opérations de calcul de nombre de jours entre deux dates font partie des applications classiques sur ordinateur. Mais le programme présenté ce mois-ci va plus loin. Sous une présentation très "pro", avec menu, Calendrier permet de réaliser automatiquement les principales opérations sur les dates auxquelles nous sommes pratiquement confrontés : combien de jours se sont écoulés entre deux dates, quelle date seront-nous dans tant de jours, etc... et même l'affichage du calendrier d'un mois quelconque.

Ce programme est très simple à utiliser, puisque l'auteur a prévu toutes les fausses manœuvres possibles. Quand aux programmeurs néophytes, ils trouveront ici un exemple parfait de "programmation structurée", avec un listing très clair, et quelques astuces Basic intéressantes à connaître, et faciles à adapter à d'autres programmes, comme la manière de transformer une chaîne de caractères en trois valeurs numériques (lignes 1980 et suivantes).

```
100 REM "CALENDRIER"
1000 CLS
1010 PROC INI
1020 PROC MENU
1030 REPEAT
1040   IF Q=1 THEN PROC MENU
1050   IF Q=2 THEN PROC JSEM
1060   IF Q=3 THEN PROC DIFDATE
1070   IF Q=4 THEN PROC ADDNDATE
1080   IF Q=5 THEN PROC MOIS
1090   IF Q<>2 AND Q<>0 AND Q<>3 AND Q<>
4 AND Q<>5 THEN LET Q=1
1100 UNTIL Q=0
1110 VDU 25
1120 CLS
1130 END
1140 DEFPROC MENU
1150 VDU 24
1160 CLS
1170 PRINT "          * * M E N U * *"
1180 PRINT
1190 PRINT " Liste des operations .....
.....>1"
1200 PRINT " Jour de la semaine .....
.....>2"
1210 PRINT " Nombre de jours entre 2 da
tes...>3"
1220 PRINT " Date plus un nombre de jou
rs...>4"
1230 PRINT " Calendrier d'un mois.....
.....>5"
1240 PRINT " Fin d'execution.....
.....>0"
1250 PRINT
1260 INPUT " Choisissez une operation..
....>":Q
1270 ENDPROC
1280 DEFPROC INI
1290 DIM J$(8)(7),P$(2)(42),M$(9)(12)
1300 FOR K=1 TO 7
1310   READ J$(K)
1320 NEXT K
1330 FOR K=1 TO 12
1340   READ M$(K)
1350 NEXT K
1360 DATA Lundi,Mardi,Mercredi,Jeudi,Ven
dredi,Samedi,Dimanche
1370 DATA Janvier,Fevrier,Mars,Avril,Mai
,Juin,Juillet,Aout,Septembre,Octobre,Nov
embre,Decembre
1380 ENDPROC
1390 DEFPROC NJOURS
1400 WHILE M<=2
1410   LET M=M+12
1420   LET A=A-1
1430 WEND
1440 LET M=M+1
1450 LET N=INT(365.25*A)+INT(M*30.6001)+
J
1460 ENDPROC
1470 DEFPROC CDATE
1480 LET A=INT((N-122.1)/365.25)
1490 LET M=INT((N-INT(365.25*A))/30.6001
)
1500 LET J=N-INT(365.25*A)-INT(M*30.6001
)
1510 IF M>13 THEN LET M=M-12
1520 LET M=M-1
1530 IF M<=2 THEN LET A=A+1
1540 ENDPROC
1550 DEFPROC JSEM
1560 CLS
1570 PRINT "          =====
===="
1580 PRINT "          JOUR DE LA SEMAI
NE"
1590 PRINT "          =====
===="
1600 PRINT
1610 PRINT "          Donnez une date"
1620 PRINT
1630 INPUT " JJ/MM/AAAA >>>>":Q$
1640 GOSUB LABEL L
1650 LET j=J,m=M,a=A
1660 PROC NJOURS
```

```

1670 LET S=((N+4) MOD 7)+1
1680 PRINT "-----"
":
1690 PRINT "      Le ":j:" ";M$(m);" ";a
:" est un ";J$(S)
1700 PRINT "-----"
":
1710 PRINT
1720 INPUT "  Choisissez une operation..
.....>";Q
1730 ENDPROC
1740 DEFPROC DIFDATE
1750 CLS
1760 PRINT "      =====
=====
1770 PRINT "      Nombre de jours entre
2 dates"
1780 PRINT "      =====
=====
1790 PRINT
1800 INPUT "  Date1 JJ/MM/AAAA >>>>";Q$
1810 GOSUB LABEL L
1820 LET j=J,m=M,a=A
1830 PROC NJOURS
1840 LET n=N,R$=Q$
1850 INPUT "  Date2 JJ/MM/AAAA >>>>";Q$
1860 PRINT "-----"
":
1870 GOSUB LABEL L
1880 PROC NJOURS
1890 LET n=N-n
1900 PROC CDATE
1910 PRINT "  Entre le ":j:" ";M$(m);" "
;a
1920 PRINT "      et le ":J:" ";M$(M):
" ";A
1930 PRINT "      il y a ":n:"
jours"
1940 PRINT "-----"
":
1950 INPUT "  Choisissez une operation..
.....>";Q
1960 ENDPROC
1970 LABEL L
1980 LET J=VAL(LEFT$(Q$,2))
1990 LET M=VAL(MID$(Q$,3,2))
2000 LET A=VAL(RIGHT$(Q$,4))
2010 RETURN
2020 DEFPROC ADDNDATE
2030 CLS
2040 PRINT "      =====
=====
2050 PRINT "      Date plus un nombre d
e jours"
2060 PRINT "      =====
=====
2070 PRINT
2080 INPUT "  Date JJ/MM/AAAA >>>>";Q$
2090 INPUT "  Nombre de jours >>>>";n
2100 GOSUB LABEL L
2110 LET j=J,m=M,a=A
2120 PROC NJOURS
2130 LET N=N+n
2140 PROC CDATE
2150 PRINT "-----"
":
2160 VDU 24

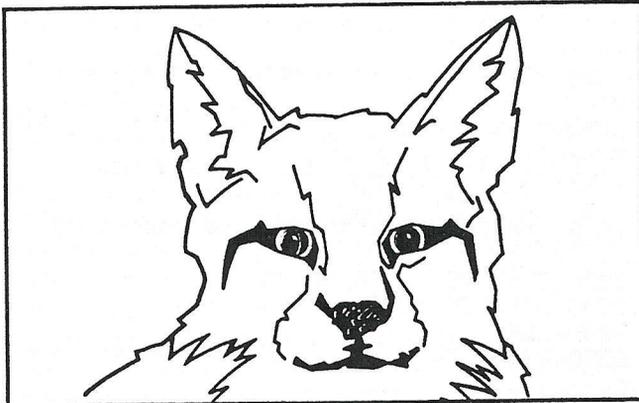
```

```

2170 PRINT "      La date du ":n;" ieme jou
rs apres le"
2180 PRINT "      ":j;" ";M$(m);" ";a;" est l
e:"
2190 PRINT "      ":J;" ";M$(
M);" ";A
2200 PRINT "-----"
":
2210 INPUT "  Choisissez une operation..
.....>";Q
2220 ENDPROC
2230 DEFPROC MOIS
2240 CLS
2250 PRINT "      =====
=====
2260 PRINT "      CALENDRIER DU MOIS
"
2270 PRINT "      =====
=====
2280 PRINT
2290 PRINT "      Donnez le mois et l'annee
"
2300 PRINT
2310 INPUT "  MM/AAAA >>>>";Q$
2320 LET M=VAL(LEFT$(Q$,2))
2330 LET A=VAL(RIGHT$(Q$,4))
2340 LET J=1
2350 FOR K=1 TO 42
2360   LET P$(K)=" "
2370 NEXT K
2380 LET j=J,m=M,a=A
2390 PROC NJOURS
2400 LET S=((N+4) MOD 7)+1
2410 FOR K=S TO 42
2420   LET J=j,M=m,A=a
2430   GOSUB LABEL V
2440   IF V<>0 THEN LET P$(K)=STR$(j)
2450   LET j=j+1
2460 NEXT K
2470 GOTO 2540
2480 LABEL V
2490 LET V=1
2500 PROC NJOURS
2510 PROC CDATE
2520 IF J<>j OR M<>m OR A<>a THEN LET V
=0
2530 RETURN
2540 CLS
2550 VDU 24
2560 CLS
2570 PRINT "      Mois de ":M$(m):"
";a
2580 PRINT "-----"
":
2590 FOR K=1 TO 7
2600   PRINT J$(K):
2610   FOR k=1 TO 6
2620     PRINT TAB (k*5+8);P$(K+(k-1)*7
):
2630   NEXT k
2640   VDU 19
2650 NEXT K
2660 PRINT "-----"
":
2670 INPUT "  Choisissez une operation..
.....>";Q
2680 ENDPROC

```

BIORYTHMES



BIORYTHMES

D'après les spécialistes, notre vie est gouvernée par trois cycles, correspondant à notre potentiel affectif, physique et intellectuel. Ces trois cycles varient régulièrement entre un maximum et un minimum, selon des périodes variables, mais communes à tous les individus. A partir de votre date de naissance, ce programme peut tracer, à compter d'un jour précis, les évolutions de vos trois cycles de base, et vous indiquer les périodes fastes ou néfastes à venir...

```

1 CLS
2 DIM z(3)
3 PRINT TAB 10;"*** BIORYTHME ***"
4 PRINT ""
5 PRINT ""
6 PRINT "Date de naissance:"
7 PROC EQNB
8 LET N=D
9 PRINT "Vous etes ne(e) un ";
10 PROC JOUR(N)
11 PRINT
12 PRINT "Date du jour:"
13 PROC EQNB
14 PROC AFF(D,N)
15 LET D=D+21
16 LET Q$=KEY$
17 IF Q$="" THEN GOTO 16
18 IF Q$="C" THEN GOTO 14
19 CLS
20 GOTO 12
21 DEFPROC EQNB
22 PRINT
23 INPUT "JOUR=";J
24 INPUT " MOIS(1/12)=";M
25 INPUT " ANNEE=";A
26 LET D=A*365+J+31*(M-1)
27 IF M>2 THEN LET D=D-INT(0.4*M+2.3),H=A
28 ELSE LET H=A-1
29 LET D=D+INT(H/4)-INT(0.75+INT(H/100)*0.75)
30 ENDPROC
31 DEFPROC JOUR(V)
32 LET V=V MOD 7*2+34
33 GOTO V
34 PRINT "Samedi ";
35 GOTO 47

```

```

36 PRINT "Dimanche ";
37 GOTO 47
38 PRINT "LUNDI ";
39 GOTO 47
40 PRINT "Mardi ";
41 GOTO 47
42 PRINT "Mercredi ";
43 GOTO 47
44 PRINT "Jeudi ";
45 GOTO 47
46 PRINT "Vendredi ";
47 ENDPROC
48 DEFPROC AFF(d,n)
49 CLS
50 LET R$="PIEM"
51 FOR a=1 TO 4
52 INK a*2-1
53 PRINT " ";MID$(R$,a,1);
54 NEXT a
55 PRINT TAB 24;"-1 0 1"
56 LET f=d-n
57 FOR a=0 TO 20
58 PROC JOUR(d+a)
59 PROC JS
60 FOR b=0 TO 9
61 LET g=14+a*10+b
62 LET z(0)=f/23,z(1)=f/33,z(2)=f/28
63 FOR c=0 TO 2
64 LET z(c)=SIN(FRAC(z(c))*2*PI)
65 NEXT c
66 LET z(3)=(z(1)+z(2)+z(0))/3
67 DOT 150,g
68 DOT 200,g
69 DOT 250,g
70 FOR c=0 TO 3
71 INK c*2+1
72 DOT 200+INT(z(c)*50),g
73 NEXT c
74 LET f=f+0.1
75 NEXT b
76 MOVE 0,(a+2)*10
77 NEXT a
78 PRINT "** C:CONTINUER D:AUTRE D
ATE **"
79 ENDPROC
80 DEFPROC JS
81 IF J<29 OR (J<31 AND M<>2) THEN GOTO
97
82 IF M<>2 THEN GOTO 92
83 LET H=A MOD 100
84 IF H=0 THEN LET H=A DIV 100
85 ELSE LET H=A
86 IF H MOD 4=0 THEN LET H=0
87 IF J=29 AND H=0 THEN GOTO 97
88 LET J=1,M=M+1
89 IF M<13 THEN GOTO 97
90 LET M=1,A=A+1
91 GOTO 97
92 LET H=M MOD 2
93 IF M>7 THEN LET H=H+1
94 IF H=2 THEN GOTO 97
95 IF J>H+30 THEN GOTO 88
96 ELSE GOTO 97
97 PRINT STR$(J)+"-"+STR$(M)+"-"+STR$(A)
98 LET J=J+1
99 ENDPROC

```