

SUPER GÉNÉRATEUR DE CARACTÈRES SUR AMSTRAD



JEAN-FRANÇOIS SEHAN

***SUPER GÉNÉRATEUR
DE CARACTÈRES
SUR AMSTRAD***

Autres ouvrages relatifs à l'Amstrad CPC 464, 664 ou 6128

- Amstrad en famille - Jean-François Sehan
- 102 programmes pour Amstrad - Jacques Deconchat
- Super jeux Amstrad - Jean-François Sehan
- La découverte de l'Amstrad - Daniel-Jean David
- Exercices en BASIC pour Amstrad - Maurice Charbit
- BASIC Amstrad, 1. Méthodes pratiques - Jacques Boisgontier et Bruno Césard
- BASIC Amstrad, 2. Programmes et fichiers - Jacques Boisgontier
- BASIC plus 80 routines sur Amstrad - Michel Martin
- Clefs pour Amstrad, 1. Système de base - Daniel Martin
- Assembleur de l'Amstrad - Marcel Henrot

A paraître :

- Périphériques et gestion de fichiers sur Amstrad - Daniel-Jean David
- Simulation et intelligence artificielle sur Amstrad - René Descamps
- Clefs pour Amstrad, 2. Système disque - Daniel Martin
- Turbo Pascal sur Amstrad - Frédéric Blanc et Pierre Brandeis

Pour tout problème rencontré dans les ouvrages P.S.I.
vous pouvez nous contacter au numéro ci-dessous :

Numéro Vert/Appel Gratuit en France

05 21 22 01

(Composer tous les chiffres, même en région parisienne)

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les «copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective», et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite» (alinéa 1^{er} de l'article 40)

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal

© Éditions du P.S.I. - B.P. 86 - 77402 Lagny cedex
1986

ISBN: 2-86595-300-9

MICRO ET TECHNIQUES

SUPER GÉNÉRATEUR DE CARACTÈRES SUR AMSTRAD

JEAN-FRANÇOIS SEHAN



**ÉDITIONS DU P.S.I.
1985**

Sommaire

Introduction	7
Pourquoi des caractères graphiques ?	8
Qu'est-ce qu'un caractère graphique ?	8
Un peu de mathématique	8
Convention	9

Première partie

1. – Mode d'emploi du générateur	13
Les déplacements du curseur	14
Les modes de tracé	14
Les autres commandes du générateur	15
2. – Le programme du générateur	23
Précautions d'emploi	23
Programme	24
Commentaires du programme	25
Liste des variables	39
Comment modifier le générateur	40
3. – Exemple d'utilisation	43
Programme	44

Commentaires du programme	45
Liste des variables	48
4. – Graphiques et langage machine	49
Seconde partie	
Graphiques prêts à l'emploi	63
Musique	65
Electronique	73
Maître cartier	81
Echec et mat	89
Fruit-machine	97
Western	103
Oasis	107
Ski	111
Guerre des étoiles	115
Humanoïdes	121
Home, sweet home	129
Code de la route	133
Trafic	141
Géographie	147
Paname	153
Bric-à-brac	157
Ménagerie	169
Nourritures terrestres	179
Signes zodiacaux	187
Trames	195
Index des graphiques	201
Conseils de lecture	205

Introduction

Ce livre se veut avant tout un ouvrage d'approfondissement du BASIC Amstrad par l'utilisation et la compréhension d'un programme complexe qui va vous permettre de créer vos propres graphiques.

Le programme est écrit pour les ordinateurs Amstrad mais il s'adapte facilement aux autres micro-ordinateurs du marché.

Les explications, données ligne par ligne, vous aideront à transférer facilement les listings sur une autre machine.

Cependant, quelques limites sont à observer. Il faut bien sûr que votre ordinateur ait un jeu de caractères redéfinissable. Généralement, des instructions spécifiques permettent de redéfinir la forme des caractères, mais pour certaines machines, c'est l'instruction POKE qui modifie directement la mémoire : vous devez vous référer à votre manuel d'instructions pour savoir dans quel cas vous vous trouvez. La deuxième contrainte technique concerne la sauvegarde des fichiers. Votre ordinateur doit posséder des instructions de sauvegarde des données sur cassette ou sur disquette pour conserver tous les graphiques que vous aurez créés.

Pourquoi des caractères graphiques ?

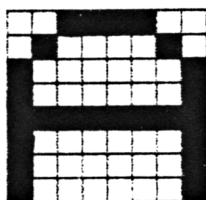
Pour animer vos programmes de jeux ou d'éducation, il est toujours plus parlant de "matérialiser" les personnages ou les objets par de vrais dessins. Imaginez vos "envahisseurs" représentés par de simples étoiles ou les pièces d'un échiquier désignées par leurs initiales !

D'autre part, les instructions graphiques qui tracent point par point des lignes, des rectangles ou des cercles sont généralement très lentes. Elles sont très pratiques quand il s'agit de "planter le décor", mais seuls les caractères graphiques permettent de déplacer rapidement des objets sur l'écran.

Qu'est-ce qu'un caractère graphique ?

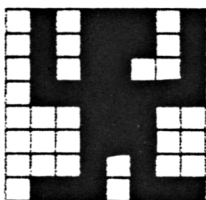
C'est tout simplement un ensemble de points qui forme un dessin. Il peut représenter une lettre de l'alphabet, un chiffre ou tout autre signe.

Exemple : la lettre A.



Mais si vous le redessinez, vous pouvez obtenir une balle pour votre jeu de tennis ou un robot pour vos "envahisseurs". Un caractère graphique est donc un groupement de 8 fois 8 points qui donne une forme.

Exemple : caractère inventé.



Un peu de mathématique

8 fois 8 points égalent 64 points. Chaque point peut être allumé ou éteint (deux états différents) ; il y a donc 2^{64} solutions dans une matrice 8×8

($2^{64} = 18E18$). Bien sûr, toutes ces possibilités ne donnent pas forcément quelque chose de cohérent mais, dans ces 18 milliards de milliards de combinaisons, il en reste beaucoup d'exploitables !

Convention

Dans l'ensemble du livre, nous appelons "caractère" cet ensemble de 8×8 , et "graphique" l'association de quatre caractères (donc une matrice de 16 sur 16). Dans la plupart des cas, un caractère seul offre peu de possibilités du fait de sa taille très réduite sur l'écran. Mais rien ne vous empêche de dessiner un seul caractère dans un graphique.

Première partie

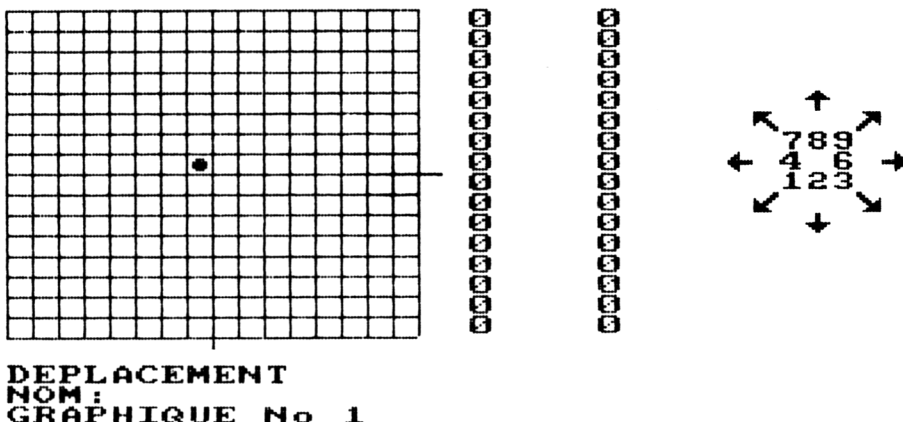
1

Mode d'emploi du générateur

Avant d'utiliser le générateur, vous devez vous assurer que vous êtes en majuscule : ceci est obligatoire pour le bon déroulement du programme.

Dès le lancement par la commande RUN, suivie de ENTER, l'ordinateur vous demande s'il s'agit d'une première utilisation. En effet, il est nécessaire de relire le fichier créé antérieurement. La machine n'ayant aucune possibilité de vérifier l'existence de données sur une cassette, cette question vous sera posée systématiquement à chaque utilisation.

Si c'est la première utilisation, répondez par la lettre O suivie de ENTER. L'écran 1 s'affiche alors. Sinon, répondez par la lettre N suivie de ENTER ou appuyez simplement sur ENTER. Un message vous indique alors le moment où vous devez mettre le lecteur de cassettes ou de disquettes en lecture. Quand le fichier a été relu, le programme en affiche toujours le premier graphique.



Ecran 1

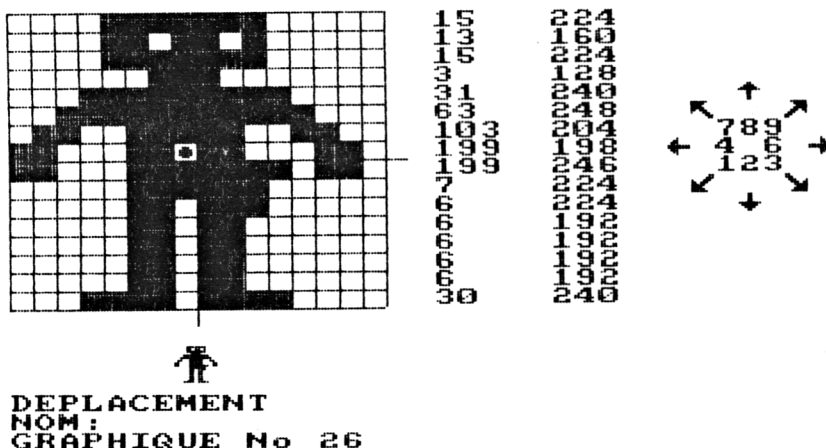
Les déplacements du curseur

Toutes les touches de déplacement vous sont indiquées à droite de l'écran. Elles permettent de déplacer le curseur dans tous les sens, y compris les quatre diagonales. Si le curseur arrive sur le bord de la grille, il revient automatiquement sur le bord opposé (s'il arrive à gauche, il réapparaît à droite, s'il arrive en bas, il réapparaît en haut). Cela évite d'avoir à traverser tout l'écran si vous désirez aller sur un bord opposé. Faites des essais pour bien comprendre le principe de déplacement du curseur.

Les modes de tracé

Si vous avez déplacé le curseur sur la grille, vous avez pu remarquer qu'il ne se passe rien. En effet, le programme se place toujours en mode **DEPLACEMENT** en début de session (ce mot est inscrit en bas de l'écran). Le curseur se déplace mais ne modifie pas la grille. Pour commencer à créer un graphique, il faut changer de mode. C'est le rôle de la touche **M**. Chaque fois que vous appuyez sur cette touche, un mode nouveau apparaît en bas de l'écran.

Le mode **NOIR** permet de déplacer le curseur en laissant une trace noire derrière lui. Le mode **BLANC** fait exactement l'inverse en laissant une trace blanche qui efface éventuellement les carrés noirs déjà affichés. Le mode **INVERSE** est un mélange des deux précédents. Il place des carrés noirs sur les cases blanches et des carrés blancs sur les cases noires. En plus des différents modes, la touche **5** permet de changer la couleur d'un point. Comme précédemment, faites des essais en déplaçant le curseur et en changeant de mode.

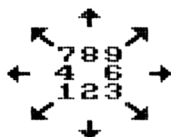


Ecran 2

Les autres commandes du générateur

Comme il en existe beaucoup, la touche ? va nous donner un menu d'aide dans lequel figurent toutes les commandes accompagnées d'une explication. Dès que vous l'aurez consulté et aurez trouvé la commande dont vous avez besoin, il vous suffira d'appuyer sur la touche ENTER pour revenir au graphique en cours.

A- RECOPIER UN AUTRE GRAPHIQUE
C- NUMERO DU GRAPHIQUE
D- ENTRER LES VALEURS DECIMALES
E- EFFACER UN GRAPHIQUE
F- FIN: SAUVEGARDE SUR CASSETTE
H- HORIZON. (AJOUT., RETIRER, MIROIR, SCROLL
I- INVERSER UN GRAPHIQUE
L- LISTE DES GRAPHIQUES
M- MODE (DEPL. NOIR, BLANC, INVERSE)
N- DONNER UN N°M
R- ROTATION 90 DEGRES
U- VERTICA. (AJOUT., RETIRER, MIROIR, SCROLL
S- CHANGER LA COULEUR D'UN POINT



Ecran 3

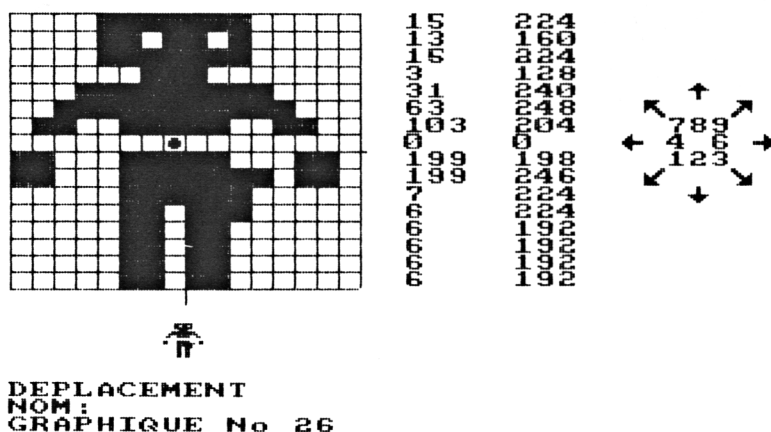
□ La commande I

Nous avons vu que le mode INVERSE permet d'inverser les carrés sur lesquels le curseur se déplace. La commande I permet la même chose, mais sur la totalité de l'écran en une seule opération.

□ La commande H

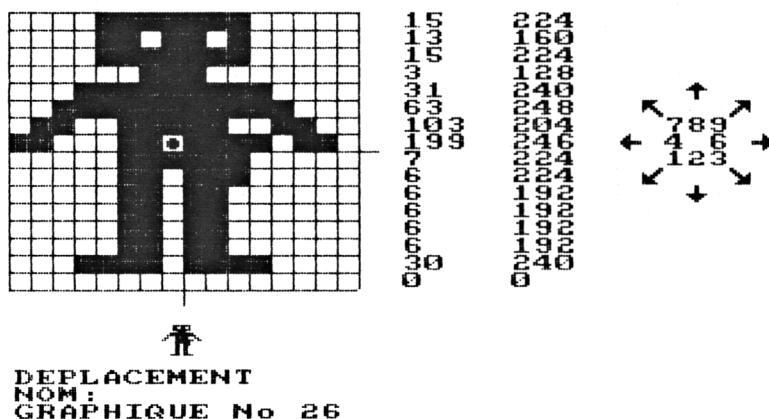
Cette commande est multiple. Après la frappe de la touche H, le programme demande si vous désirez ajouter ou retirer une ligne, obtenir un miroir ou faire défiler le graphique (scroll) selon un axe horizontal. Vous devez alors taper une des touches A, R, M ou S.

- A pour Ajouter : le programme ajoute une ligne de carrés blancs à la position du curseur. La dernière ligne de la grille est perdue.



Ecran 6

- R pour Retirer : le programme supprime la ligne à la position du curseur. Cette ligne est donc perdue, une ligne blanche apparaît alors en bas de la grille.

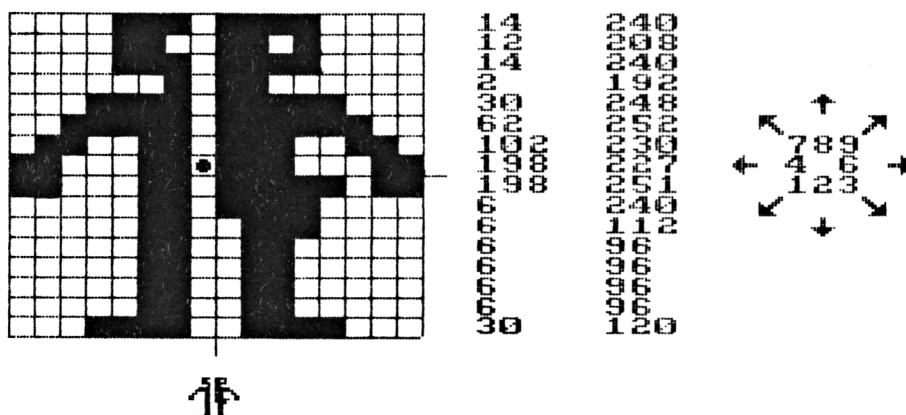


Ecran 7

□ *La commande V*

Cette commande réalise la même chose que la commande H mais à la verticale. A et R ajoutent et retirent des colonnes, M et S donnent un miroir ou font défiler la grille selon un axe vertical.

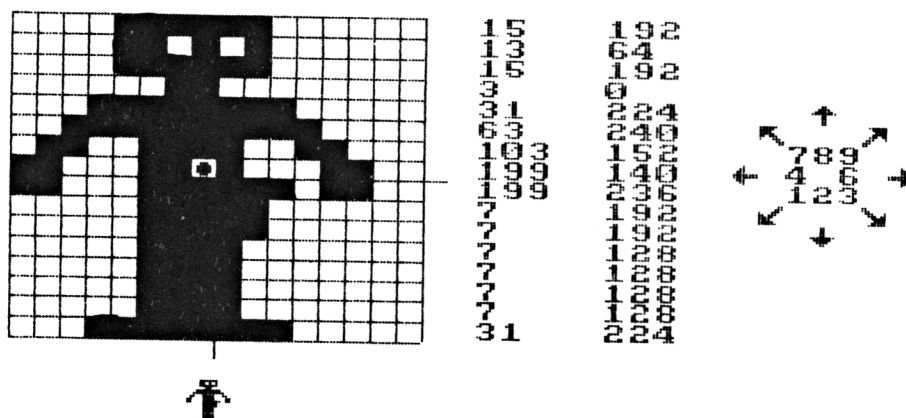
- A pour Ajouter :



INVERSE
NOM:
GRAPHIQUE No 26

Ecran 10

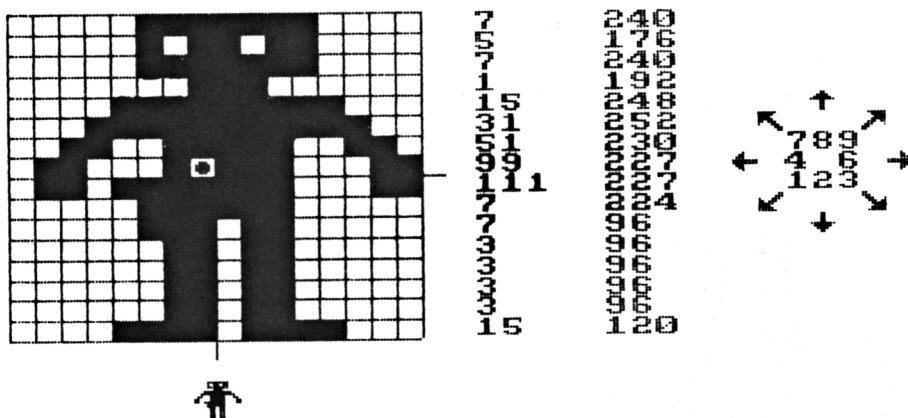
- R pour Retirer :



DEPLACEMENT
NOM:
GRAPHIQUE No 26

Ecran 11

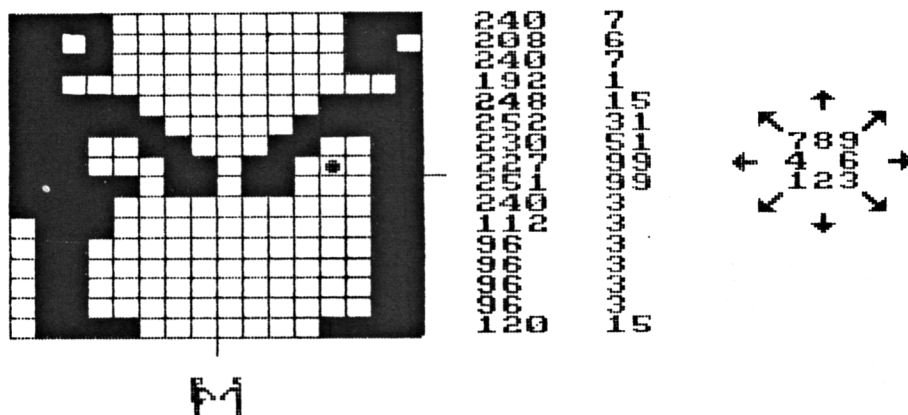
- M pour Miroir :



DEPLACEMENT
NOM:
GRAPHIQUE No 26

Ecran 12

- S pour Scroll :

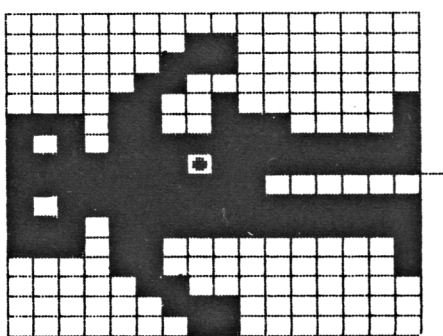


DEPLACEMENT
NOM:
GRAPHIQUE No 26

Ecran 13

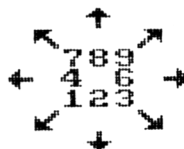
☐ La commande R

Cette commande exécute une rotation d'un quart de tour de la grille, dans le sens inverse des aiguilles d'une montre. Quatre rotations successives replacent le graphique dans sa position originale.



0
1
3
6
12
20
27
34
41
48
55
62
69
76
83
90
97
104
111
118
125
132
139
146
153
160
167
174
181
188
195
202

0
1
1
2
2
3
3
4
4
5
5
6
6
7
7
8
8
9
9
10
10
11
11
12
12
13
13
14
14
15
15
16
16
17
17
18
18
19
19
20
20
21
21
22
22
23
23
24
24
25
25
26
26
27
27
28
28
29
29
30
30
31
31
32
32
33
33
34
34
35
35
36
36
37
37
38
38
39
39
40
40
41
41
42
42
43
43
44
44
45
45
46
46
47
47
48
48
49
49
50
50
51
51
52
52
53
53
54
54
55
55
56
56
57
57
58
58
59
59
60
60
61
61
62
62
63
63
64
64
65
65
66
66
67
67
68
68
69
69
70
70
71
71
72
72
73
73
74
74
75
75
76
76
77
77
78
78
79
79
80
80
81
81
82
82
83
83
84
84
85
85
86
86
87
87
88
88
89
89
90
90
91
91
92
92
93
93
94
94
95
95
96
96
97
97
98
98
99
99



DEPLACEMENT
NOM:
GRAPHIQUE No 26

Ecran 14

☐ La commande E

Elle efface complètement le caractère en cours. Cette commande est à utiliser avec précaution.

☐ La commande N

Chaque graphique doit porter un nom si l'on veut le retrouver. Le programme saisit, en bas de l'écran, le nom du graphique en cours. Entrez le nom, suivi de ENTER au clavier.

☐ La commande L

Cette commande liste à l'écran la totalité des noms entrés par l'option N. Il est possible d'arrêter et de reprendre le listage en appuyant sur n'importe quelle touche, et ceci à tout moment.

1	-	BATEAU
2	-	JE
3	-	PERSONNAGE
4	-	ESCARGOT
5	-	TORTUE
6	-	ANANAS
7	-	HAMBURGER
8	-	CHANDELIER
9	-	PISTOLET
10	-	HOMME
11	-	BOUGIE
12	-	LOUPE
13	-	PIPE

Ecran 15

☐ *La commande A*

Si vous voulez tracer un dessin en prenant comme base un graphique déjà existant, vous devez utiliser cette commande. Le programme demande en bas de l'écran le numéro du graphique que vous désirez recopier dans le graphique en cours. En multipliant un graphique de base, on peut obtenir un dessin animé après avoir modifié chaque fois la position du personnage.

☐ *La commande D*

Si vous désirez conserver dans le fichier des graphiques déjà utilisés dans vos programmes, vous pouvez avec cette commande entrer directement les valeurs décimales. Le programme demande en premier lieu le numéro du caractère (entre 1 et 4), puis il saisit les huit valeurs correspondantes.

	1	2	
	3	4	
NUMERO	DU	CARACTERE?	4
VALEUR	1	?	246
VALEUR	2	?	224
VALEUR	3	?	224
VALEUR	4	?	192
VALEUR	5	?	192
VALEUR	6	?	192
VALEUR	7	?	192
VALEUR	8	?	246

Ecran 16

☐ *La commande F*

Cette commande entraîne la sauvegarde du fichier sur cassette ou disquette. Le programme attend que le lecteur de cassettes ou de disquettes soit en position d'enregistrement. Il suffit alors de taper sur la touche ENTER pour commencer l'écriture.

2

Le programme du générateur

Précautions d'emploi

Le programme étant très long, il est vivement conseillé de faire des sauvegardes sur cassette ou disquette le plus souvent possible. N'attendez pas qu'il soit entièrement tapé et testé, personne n'est à l'abri d'une coupure de courant. Par précaution, il est indispensable d'avoir un double du programme sur une autre cassette ou disquette. Il en est de même pour la sauvegarde des graphiques.

Avant de faire tourner le programme, vous devez initialiser la machine en appuyant simultanément sur les touches SHIFT CTRL et ESC. Vérifiez bien toutes les lignes sans oublier un seul signe, et sans confondre le chiffre Ø avec la lettre O ou le chiffre 1 avec la lettre l.

Programme

```
10 REM GENERATEUR DE CARACTERES
20 REM -----
30 DEFINT A-Z
40 MODE 1
50 DIM S(100,4,8),N$(100),B$(16),BB$(16)
60 N=1
70 INPUT"PREMIERE UTILISATION (O/N)";A$
80 IF A$="O" THEN 190
90 CLS:PRINT"METTEZ LE DATACORDER EN LECTURE"
100 OPENIN "!CARACTERES"
110 INPUT#9,N
120 FOR I=1 TO N
130 INPUT#9,N$(I)
140 FOR J=1 TO 4
150 FOR K=1 TO 8
160 INPUT#9,S(I,J,K)
170 NEXT K,J,I
180 CLOSEIN
190 SYMBOL AFTER 128
200 SYMBOL 231,255,128,156,190,190,190,156,128
210 SYMBOL 232,255,255,255,255,255,255,255
220 SYMBOL 233,255,128,128,128,128,128,128
230 SYMBOL 234,128,128,128,128,128,128,128
240 SYMBOL 235,255,0,0,0,0,0,0
250 SYMBOL 236,252,248,240,248,220,142,7,3
260 SYMBOL 237,63,31,15,31,59,113,224,192
270 SYMBOL 238,3,7,142,220,248,240,248,252
280 SYMBOL 239,192,224,113,59,31,15,31,63
290 FOR I=128 TO 192:SYMBOL I,0,0,0,0,0,0,0:
NEXT I
300 S=1:X=8:Y=8:M=1
310 M$(1)="DEPLACEMENT":M$(2)="NOIR":M$(3)="BL
ANC":M$(4)="INVERSE"
320 X(1)=18:X(2)=23:X(3)=18:X(4)=23:Y(1)=0:Y(2
)=0:Y(3)=8:Y(4)=8
330 F$(1)=" "+CHR$(240):F$(2)=" "+CHR$(236)+
" "+CHR$(237):F$(3)=" 789"
340 F$(4)=CHR$(242)+" 4 6 "+CHR$(243)
350 F$(5)=" 123":F$(6)=" "+CHR$(238)+" "+CH
R$(239):F$(7)=" "+CHR$(241)
380 REM -----
390 REM DESSIN DU GENERATEUR
400 MODE 1
410 FOR I=1 TO 16
420 FOR J=1 TO 16:PRINT CHR$(233);:NEXT J
430 IF I=9 THEN PRINT CHR$(233) ELSE PRINT CHR
$(234)
```

Commentaires du programme

30 Toutes les variables numériques contiendront des nombres entiers
pour une question de rapidité.

40 L'écran est en mode texte 40×25.

50 Dimensionnement des tableaux (voir liste des variables).

60 Au début du programme, il n'y a qu'un seul graphique.

70 On demande si c'est la première utilisation.

80 Si c'est la première utilisation, on ne lit pas le fichier.

90 On indique à l'utilisateur de mettre la cassette en route.

100 On ouvre le fichier CARACTERES.

110 La variable N donne le nombre de graphiques dans le fichier.

120 On recherche les N graphiques.

130 Lecture du nom dans N\$().

140 Lecture des quatre caractères d'un graphique.

150 Lecture des huit nombres qui composent un caractère.

160 Les valeurs sont placées dans le tableau S(,,).

170 NEXT des trois boucles.

180 Fermeture du fichier CARACTERES.

190 Les caractères définis commencent au code ASCII 128.

200 Définition du caractère 231.

210 Définition du caractère 232.

220 Définition du caractère 233.

230 Définition du caractère 234.

240 Définition du caractère 235.

250 Définition du caractère 236.

260 Définition du caractère 237.

270 Définition du caractère 238.

280 Définition du caractère 239.

290 Tous les caractères de 128 à 192 sont remis à zéro.

300 S = numéro du graphique en cours, X et Y = coordonnées du
curseur, M = mode.

310 Nom des modes possibles.

320 Positions des valeurs décimales des quatre caractères sur l'écran.

330 }
350 } Dessin de la définition des touches de déplacement.

400 On efface l'écran.

410 Il y a seize lignes à afficher.

420 Il y a seize colonnes par ligne.

430 On ferme la ligne. On indique si c'est la neuvième ligne.


```

440 NEXT I
450 FOR I=1 TO 16
460 IF I=9 THEN PRINT CHR$(233); ELSE PRINT CH
R$(235);
470 NEXT I
480 LOCATE 1,25:PRINT"GRAPHIQUE No";S:LOCATE 1
,24:PRINT "NOM:";N$(S):LOCATE 1,23:PRINT M$(M)
490 FOR I=1 TO 7:LOCATE 29,I+4:PRINT F$(I):NEX
T I
580 REM -----
590 REM AFFICHER UN GRAPHIQUE
600 FOR P=1 TO 4
610 IF P>2 THEN YY=8 ELSE YY=0
620 IF P/2=INT(P/2) THEN XX=8 ELSE XX=0
630 FOR I=1 TO 8:FOR J=7 TO 0 STEP -1
640 LOCATE XX+8-J,YY+I
650 IF S(S,P,I) AND 2^J THEN PRINT CHR$(232) E
LSE PRINT CHR$(233)
660 NEXT J,I,P
670 FOR I=1 TO 4
680 SYMBOL 127+I,S(S,I,1),S(S,I,2),S(S,I,3),S(
S,I,4),S(S,I,5),S(S,I,6),S(S,I,7),S(S,I,8)
690 NEXT I
700 LOCATE 8,19:PRINT CHR$(128);CHR$(129)
710 LOCATE 8,20:PRINT CHR$(130);CHR$(131)
720 GOSUB 1500
730 LOCATE 1,24:PRINT CHR$(18);"NOM:";N$(S)
740 GOTO 1180
780 REM -----
790 REM DEPLACEMENTS
800 IF X<9 THEN P=1:XX=X-1 ELSE P=2:XX=X-9
810 IF Y<9 THEN YY=Y ELSE P=P+2:YY=Y-8
820 A$=INKEY$:IF A$="" THEN 820
830 IF A$="C" THEN GOSUB 1300:GOTO 600
840 IF A$="M" THEN GOSUB 1400
850 IF A$="N" THEN GOSUB 3500
860 IF A$="F" THEN 4200
870 IF A$="L" THEN GOSUB 3600:GOTO 400
880 IF A$="?" THEN GOSUB 1800:GOTO 400
890 IF A$="I" THEN GOSUB 1600:GOTO 600
900 IF A$="E" THEN GOSUB 1700:GOTO 600
910 IF A$="H" THEN GOSUB 2100:GOTO 600
920 IF A$="V" THEN GOSUB 2500:GOTO 600

```

440 NEXT de la boucle.
 450 Il y a seize colonnes par ligne.
 460 On ferme la grille. On indique si c'est la neuvième colonne.
 470 NEXT de la boucle.
 480 On affiche le nom et le numéro du graphique en cours ainsi que le mode.
 490 On affiche la définition des touches de déplacement.
 600 Boucle pour les quatre caractères qui composent un graphique.
 610 On initialise la position verticale des caractères.
 620 On initialise la position horizontale des caractères.
 630 Huit nombres composent un caractère, chaque nombre comporte huit bits.
 640 On positionne le curseur.
 650 Si le bit est à 1, on affiche un carré noir, sinon un carré blanc.
 660 NEXT des trois boucles.
 670 Définition des quatre caractères dans leur taille réelle.
 680 Définition avec SYMBOL des caractères à partir du code ASCII 128.
 690 NEXT de la boucle.
 700 On affiche les deux premiers caractères dans leur taille réelle.
 710 Puis les deux suivants.
 720 Le sous-programme en 1500 affiche les valeurs décimales.
 730 On affiche le nom du graphique.
 740 On va en 1180 dans la partie DEPLACEMENT pour afficher le curseur.
 800 } En fonction de la position du curseur (X et Y), on recherche dans
 810 } quel caractère on se trouve (initialisation de P, XX et YY).
 820 On saisit dans A\$ la dernière touche enfoncée.
 830 Si A\$="C", on change de graphique. On retourne en 600 pour l'affichage du nouveau graphique.
 840 Si A\$="M", on change de mode (noir, blanc, déplacement et inverse).
 850 Si A\$="N", le programme saisit le nom du graphique en cours.
 860 Si A\$="F", le programme sauve le fichier sur cassette.
 870 Si A\$="L", on affiche le nom des graphiques déjà définis.
 880 Si A\$="?", le programme affiche un menu d'aide.
 890 Si A\$="I", on inverse tous les pixels des quatre caractères.
 900 Si A\$="E", le programme efface complètement le graphique en cours.
 910 Si A\$="H", on effectue une modification horizontale (voir sous-programme).
 920 Si A\$="V", on effectue une modification verticale (voir sous-programme).

```

930 IF A$="R" THEN GOSUB 2900:GOTO 600
940 IF A$="D" THEN GOSUB 3800:GOTO 400
950 IF A$="A" THEN GOSUB 4000:GOTO 600
960 IF A$<"1" OR A$>"9" THEN 820
970 IF M=4 OR A$="5" THEN S(S,P,YY)=S(S,P,YY)X
OR(2^(7-XX)):GOTO 1010
980 IF M=1 THEN 1050
990 IF M=3 THEN S(S,P,YY)=S(S,P,YY)AND 255-2^(
7-XX)
1000 IF M=2 THEN S(S,P,YY)=S(S,P,YY)OR(2^(7-XX
))
1010 SYMBOL 127+P,S(S,P,1),S(S,P,2),S(S,P,3),S
(S,P,4),S(S,P,5),S(S,P,6),S(S,P,7),S(S,P,8)
1020 LOCATE 8,19:PRINT CHR$(128);CHR$(129)
1030 LOCATE 8,20:PRINT CHR$(130);CHR$(131)
1040 LOCATE X(P),Y(P)+YY:PRINT S(S,P,YY);" "
1050 LOCATE X,Y:IF S(S,P,YY) AND 2^(7-XX) THEN
PRINT CHR$(232) ELSE PRINT CHR$(233)
1060 IF A$="4" THEN X=X-1
1070 IF A$="6" THEN X=X+1
1080 IF A$="8" THEN Y=Y-1
1090 IF A$="2" THEN Y=Y+1
1100 IF A$="7" THEN X=X-1:Y=Y-1
1110 IF A$="9" THEN X=X+1:Y=Y-1
1120 IF A$="1" THEN X=X-1:Y=Y+1
1130 IF A$="3" THEN X=X+1:Y=Y+1
1140 IF X=0 THEN X=16
1150 IF X=17 THEN X=1
1160 IF Y=0 THEN Y=16
1170 IF Y=17 THEN Y=1
1180 LOCATE X,Y:PRINT CHR$(231)
1190 GOTO 800
1280 REM -----
1290 REM NUMERO DU GRAPHIQUE
1300 LOCATE 1,25:PRINT CHR$(18);"No GRAPHIQUE
(MAX: ";:IF N>99 THEN PRINT " 100 )"; ELSE PRIN
T N+1;";";
1310 INPUT;" ";A$:S=VAL(A$):IF S<1 OR S>N+1 OR
S>100 THEN 1300
1320 LOCATE 1,25:PRINT CHR$(18);"GRAPHIQUE No"
;S;
1330 IF S>N THEN N=S
1340 RETURN
1380 REM -----
1390 REM MODE
1400 M=M+1:IF M>4 THEN M=1

```

930 Si A\$="R", le programme exécute une rotation de 90 degrés.
 940 Si A\$="D", on saisit directement les valeurs décimales d'un caractère.
 950 Si A\$="A", le programme recopie les valeurs d'un graphique dans un autre.
 960 Si A\$ ne correspond pas à une touche de déplacement, on revient à la saisie de A\$.
 970 Si l'on est en mode inverse ou A\$="5", on change la valeur du pixel sous le curseur.
 980 Si l'on est en mode déplacement, on saute les lignes 990 à 1040.
 990 Si l'on est en mode blanc, on met à zéro le pixel sous le curseur.
 1000 Si l'on est en mode noir, on met à un le pixel sous le curseur.
 1010 On redéfinit avec SYMBOL le caractère concerné.
 1020 On affiche en taille réelle les deux premiers caractères.
 1030 Puis les deux autres.
 1040 On affiche la nouvelle valeur décimale.
 1050 Suivant la valeur du pixel sous le curseur, on affiche un carré noir ou blanc.
 1060 Si A\$="4", on déplace le curseur à gauche.
 1070 Si A\$="6", on déplace le curseur à droite.
 1080 Si A\$="8", on déplace le curseur en haut.
 1090 Si A\$="2", on déplace le curseur en bas.
 1100 Si A\$="7", on déplace le curseur en haut à gauche.
 1110 Si A\$="9", on déplace le curseur en haut à droite.
 1120 Si A\$="1", on déplace le curseur en bas à gauche.
 1130 Si A\$="3", on déplace le curseur en bas à droite.
 1140 Si le curseur déborde à gauche du graphique, on le positionne à droite.
 1150 Si le curseur déborde à droite du graphique, on le positionne à gauche.
 1160 Si le curseur déborde en haut du graphique, on le positionne en bas.
 1170 Si le curseur déborde en bas du graphique, on le positionne en haut.
 1180 On affiche le curseur.
 1190 On revient à la saisie de A\$.
 1300 Le programme demande le numéro du graphique dans la limite de N+1.
 1310 On saisit et l'on vérifie la réponse de l'utilisateur.
 1320 On affiche le nouveau graphique.
 1330 On modifie N si le nouveau numéro de graphique lui est supérieur.
 1340 Retour du sous-programme.
 1400 On change le mode M.

```

1410 LOCATE 1,23:PRINT CHR$(18);M$(M)
1420 RETURN
1480 REM -----
1490 REM VALEURS DECIMALES
1500 FOR I=1 TO 4
1510 FOR J=1 TO 8
1520 LOCATE X(I),Y(I)+J:PRINT S(S,I,J); " "
1530 NEXT J,I
1540 RETURN
1580 REM -----
1590 REM INVERSION
1600 FOR I=1 TO 4
1610 FOR J=1 TO 8
1620 S(S,I,J)=S(S,I,J) XOR 255
1630 NEXT J,I
1640 RETURN
1680 REM -----
1690 REM EFFACER UN GRAPHIQUE
1700 FOR I=1 TO 4
1710 FOR J=1 TO 8
1720 S(S,I,J)=0
1730 NEXT J,I
1740 RETURN
1780 REM -----
1790 REM MENU D'AIDE
1800 CLS
1810 PRINT"A- RECOPIER UN AUTRE GRAPHIQUE"
1820 PRINT"C- NUMERO DU GRAPHIQUE"
1830 PRINT"D- ENTRER LES VALEURS DECIMALES"
1840 PRINT"E- EFFACER UN GRAPHIQUE"
1850 PRINT"F- FIN: SAUVEGARDE SUR CASSETTE"
1860 PRINT"H- HORIZON. (AJOUT., RETIRER, MIROIR, S
CROLL ";
1870 PRINT"I- INVERSER UN GRAPHIQUE"
1880 PRINT"L- LISTE DES GRAPHIQUES"
1890 PRINT"M- MODE (DEPL., NOIR, BLANC, INVERS
E)"
1900 PRINT"N- DONNER UN NOM"
1910 PRINT"R- ROTATION 90 DEGRES"
1920 PRINT"V- VERTICA. (AJOUT., RETIRER, MIROIR, S
CROLL ";
1930 PRINT"5- CHANGER LA COULEUR D'UN POINT"
1940 PRINT
1950 FOR I=1 TO 7:PRINT TAB(15);F$(I):NEXT I
1960 IF INKEY$<>CHR$(13) THEN 1960
1970 RETURN
2080 REM -----
2090 REM HORIZONTALE

```

1410 On affiche ce mode.
 1420 Retour du sous-programme.
 1500 Il y a quatre caractères.
 1510 Chaque caractère est composé de huit valeurs décimales.
 1520 On affiche la valeur décimale.
 1530 NEXT des boucles.
 1540 Retour du sous-programme.
 1600 Il y a quatre caractères.
 1610 Chaque caractère comporte huit lignes.
 1620 On inverse chaque valeur en effectuant un OU exclusif
 (fonction XOR).
 1630 NEXT des boucles.
 1640 Retour du sous-programme.
 1700 Il y a quatre caractères.
 1710 Chaque caractère contient huit lignes.
 1720 On remet à zéro chaque ligne.
 1730 NEXT des boucles.
 1740 Retour du sous-programme.
 1800 On efface l'écran.
 1810 }
 1940 } On affiche la définition de toutes les commandes du générateur.
 1950 Cette boucle affiche les touches de déplacement.
 1960 On attend que la touche ENTER soit enfoncée pour continuer.
 1970 Retour du sous-programme.

```

2100 LOCATE 1,22:PRINT CHR$(18); "AJOUTER, RETIR
ER, MIROIR, SCROLL (A/R/M/S)";
2110 A$=INKEY$: IF A$="" THEN 2110
2120 IF A$<>"A" AND A$<>"R" AND A$<>"M" AND A$
<>"S" THEN 2370
2130 GOSUB 3100
2140 IF A$="S" THEN 2310
2150 IF A$="M" THEN 2270
2160 IF A$="R" THEN 2220
2170 FOR I=16 TO Y+1 STEP -1
2180 B$(I)=B$(I-1)
2190 NEXT I
2200 B$(Y)="0000000000000000"
2210 GOTO 2360
2220 FOR I=Y TO 15
2230 B$(I)=B$(I+1)
2240 NEXT I
2250 B$(16)="0000000000000000"
2260 GOTO 2360
2270 FOR I=1 TO 8
2280 B$=B$(I):B$(I)=B$(17-I):B$(17-I)=B$
2290 NEXT I
2300 GOTO 2360
2310 B$=B$(1)
2320 FOR I=1 TO 15
2330 B$(I)=B$(I+1)
2340 NEXT I
2350 B$(16)=B$
2360 GOSUB 3300
2370 LOCATE 1,22:PRINT CHR$(18)
2380 RETURN
2480 REM -----
2490 REM VERTICALE
2500 LOCATE 1,22:PRINT CHR$(18); "AJOUTER, RETIR
ER, MIROIR, SCROLL (A/R/M/S)";
2510 A$=INKEY$: IF A$="" THEN 2510
2520 IF A$<>"A" AND A$<>"R" AND A$<>"M" AND A$
<>"S" THEN 2760
2530 GOSUB 3100

```

2100 On demande une des quatre options AJOUTER, RETIRER, MIROIR ou SCROLL.
 2110 On saisit dans A\$ la réponse de l'utilisateur.
 2120 Le programme vérifie cette réponse.
 2130 Le sous-programme en 3100 transfère les valeurs numériques du tableau S dans le tableau alphanumérique B\$.
 2140 Si A\$="S", on va en 2310 pour le SCROLL.
 2150 Si A\$="M", on va en 2270 pour le MIROIR.
 2160 Si A\$="R", on va en 2220 pour retirer une ligne de pixels.
 2170 Il y a seize lignes par graphique. On décale le tableau B\$ vers le bas jusqu'au curseur Y.
 2180 On décale le tableau vers le bas.
 2190 NEXT de la boucle.
 2200 On ajoute une ligne de blanc à la position du curseur.
 2210 On saute le reste du sous-programme.
 2220 Si l'on supprime une ligne du graphique, il n'en restera plus que quinze. On décale le tableau B\$ vers le haut à partir du curseur.
 2230 On décale le tableau vers le haut.
 2240 NEXT de la boucle.
 2250 On met la seizième ligne à zéro.
 2260 On saute le reste du sous-programme.
 2270 Il y a deux fois huit lignes dans un graphique. On échange la première avec la seizième, la deuxième avec la quinzième, etc.
 2280 On échange le contenu des seize lignes du tableau B\$.
 2290 NEXT de la boucle.
 2300 On saute le reste du sous-programme.
 2310 On garde dans la variable B\$ le contenu de la première ligne.
 2320 Il y a quinze lignes à monter d'un cran.
 2330 On monte chaque ligne d'un cran.
 2340 NEXT de la boucle.
 2350 On remplace à la seizième ligne le contenu de la variable B\$.
 2360 Le sous-programme en 3300 transfère les valeurs alphanumériques du tableau B\$ dans le tableau numérique S.
 2370 On efface de l'écran le message affiché au début du sous-programme.
 2380 Retour du sous-programme.
 2500 On demande une des quatre options AJOUTER, RETIRER, MIROIR ou SCROLL.
 2510 On saisit dans A\$ la réponse de l'utilisateur.
 2520 Le programme vérifie cette réponse.
 2530 Le sous-programme en 3100 transfère les valeurs numériques du tableau S dans le tableau alphanumérique B\$.


```

2540 IF A$="S" THEN 2720
2550 IF A$="M" THEN 2650
2560 IF A$="R" THEN 2610
2570 FOR I=1 TO 16
2580 B$(I)=LEFT$(B$(I),X-1)+"0"+MID$(B$(I),X,1
6-X)
2590 NEXT I
2600 GOTO 2750
2610 FOR I=1 TO 16
2620 B$(I)=LEFT$(B$(I),X-1)+RIGHT$(B$(I),16-X)
+"0"
2630 NEXT I
2640 GOTO 2750
2650 FOR I=1 TO 16
2660 FOR J=1 TO 16
2670 BB$(I)=LEFT$(BB$(I),J-1)+MID$(B$(I),17-J,
1)+RIGHT$(BB$(I),16-J)
2680 NEXT J
2690 B$(I)=BB$(I)
2700 NEXT I
2710 GOTO 2750
2720 FOR I=1 TO 16
2730 B$(I)=RIGHT$(B$(I),15)+LEFT$(B$(I),1)
2740 NEXT I
2750 GOSUB 3300
2760 LOCATE 1,22:PRINT CHR$(18)
2770 RETURN
2880 REM -----
2890 REM ROTATION DE 90 DEGRES
2900 GOSUB 3100
2910 FOR I=1 TO 16:FOR J=1 TO 16
2920 BB$(17-J)=LEFT$(BB$(17-J),I-1)+MID$(B$(I)
,J,1)+RIGHT$(BB$(17-J),16-I)
2930 NEXT J,I
2940 FOR I=1 TO 16:B$(I)=BB$(I):NEXT I
2950 GOSUB 3300
2960 RETURN
3080 REM -----
3090 REM S(,,) DANS B$( )
3100 FOR I=1 TO 16

```

2540 Si A\$="S", on va en 2720 pour le SCROLL.
 2550 Si A\$="M", on va en 2650 pour le MIROIR.
 2560 Si A\$="R", on va en 2610 pour retirer une colonne de pixels.
 2570 Il y a seize lignes par graphique.
 2580 On décale les valeurs de chaque ligne du tableau vers la droite à partir du curseur X.
 2590 NEXT de la boucle.
 2600 On saute le reste du sous-programme.
 2610 Il y a seize lignes par graphique.
 2620 On décale le tableau vers la gauche jusqu'au curseur X.
 2630 NEXT de la boucle.
 2640 On saute le reste du sous-programme.
 2650 Il y a seize lignes par graphique.
 2660 Il y a seize colonnes par graphique.
 2670 On échange le contenu de la première colonne avec celui de la seizième, le contenu de la deuxième avec celui de la quinzième, etc. Le résultat est conservé dans le tableau BB\$.
 2680 NEXT de la deuxième boucle.
 2690 On place le résultat BB\$ dans le tableau B\$.
 2700 NEXT de la première boucle.
 2710 On saute le reste du sous-programme.
 2720 Il y a seize lignes par graphique.
 2730 On décale toutes les colonnes vers la gauche ; la colonne un prend la place de la colonne seize.
 2740 NEXT de la boucle.
 2750 Le sous-programme en 3300 transfère les valeurs alphanumériques du tableau B\$ dans le tableau numérique S.
 2760 On efface de l'écran le message affiché au début du sous-programme.
 2770 Retour du sous-programme.
 2900 Le sous-programme en 3100 transfère les valeurs numériques du tableau S dans le tableau alphanumérique B\$.
 2910 Il y a seize lignes de seize colonnes par graphique.
 2920 On échange les valeurs par ligne et colonne pour obtenir une rotation de 90 degrés. Le résultat est conservé dans le tableau BB\$.
 2930 NEXT des deux boucles.
 2940 On remplace dans le tableau B\$ le résultat contenu dans le tableau BB\$.
 2950 Le sous-programme en 3300 transfère les valeurs alphanumériques du tableau B\$ dans le tableau numérique S.
 2960 Retour du sous-programme.
 3100 Il y a seize valeurs à transférer.

```

3110 IF I<9 THEN Z=1:ZZ=I ELSE Z=3:ZZ=I-8
3120 B$(I)=BIN$(S(S,Z,ZZ),8)+BIN$(S(S,Z+1,ZZ),
8)
3130 NEXT I
3140 RETURN
3280 REM -----
3290 REM B$( ) DANS S( , , )
3300 FOR I=1 TO 16
3310 IF I<9 THEN Z=1:ZZ=I ELSE Z=3:ZZ=I-8
3320 S(S,Z,ZZ)=VAL("&X"+LEFT$(B$(I),8)):S(S,Z+
1,ZZ)=VAL("&X"+RIGHT$(B$(I),8))
3330 NEXT I
3340 RETURN
3480 REM -----
3490 REM DONNER UN NOM
3500 LOCATE 1,24:PRINT CHR$(18);:INPUT"NOM";N$(
S)
3510 LOCATE 1,24:PRINT CHR$(18);"NOM:";N$(S)
3520 RETURN
3580 REM -----
3590 REM LISTE DES NOMS
3600 CLS
3610 FOR I=1 TO N
3620 IF N$(I)="" THEN 3650
3630 PRINT I;"- ";N$(I)
3640 IF INKEY$<>" " THEN WHILE INKEY$="":WEND
3650 NEXT I
3660 IF INKEY$="" THEN 3660
3670 RETURN
3780 REM -----
3790 REM ENTRER LES VALEURS DECIMALES
3800 CLS
3810 PRINT 1;2,CHR$(128);CHR$(129):PRINT 3;4,C
HR$(130);CHR$(131):PRINT
3820 INPUT"NUMERO DU CARACTERE";C
3830 IF C<1 OR C>4 THEN 3900
3840 FOR I=1 TO 8
3850 PRINT"VALEUR";I;:INPUT S(S,C,I)
3860 IF S(S,C,I)<0 OR S(S,C,I)>255 THEN 3850
3870 NEXT I
3880 SYMBOL 127+C,S(S,C,1),S(S,C,2),S(S,C,3),S
(S,C,4),S(S,C,5),S(S,C,6),S(S,C,7),S(S,C,8)
3890 GOTO 3800
3900 RETURN
3980 REM -----
3990 REM RECOPIER UN AUTRE GRAPHIQUE
4000 LOCATE 1,22:PRINT CHR$(18);"No DE L'AUTRE
GRAPHIQUE";
4010 INPUT A:IF A<1 OR A>N THEN 4050

```

- 3110 Un graphique comportant quatre caractères, la variable $Z=1$ indique que l'on transfère les caractères 1 et 2, et $Z=3$ les caractères 3 et 4. La variable ZZ donne le numéro de ligne dans le tableau S.
- 3120 On place dans une ligne alphanumérique du tableau B\$ le contenu binaire de deux lignes du tableau S.
- 3130 NEXT des boucles.
- 3140 Retour du sous-programme.
- 3300 Il y a seize valeurs à transférer.
- 3310 Un graphique comportant quatre caractères, la variable $Z=1$ indique que l'on transfère les caractères 1 et 2, et $Z=3$ les caractères 3 et 4. La variable ZZ donne le numéro de ligne dans le tableau S.
- 3320 On place, dans deux lignes du tableau S, le contenu numérique du tableau B\$.
- 3330 NEXT des boucles.
- 3340 Retour du sous-programme.
- 3500 Le programme demande le nom du caractère.
- 3510 On affiche ce nom en bas de l'écran.
- 3520 Retour du sous-programme.
- 3600 Efface l'écran.
- 3610 Il y a N noms à afficher.
- 3620 Si le nom est une chaîne vide, on n'affiche rien.
- 3630 On affiche le numéro et le nom du caractère.
- 3640 Si une touche est enfoncée, on stoppe le programme et l'on attend qu'une autre touche soit appuyée.
- 3650 NEXT de la boucle.
- 3660 On attend qu'une touche soit appuyée.
- 3670 Retour du sous-programme.
- 3800 Efface l'écran.
- 3810 On affiche les quatre caractères qui composent le graphique.
- 3820 Le programme saisit le numéro du caractère à modifier.
- 3830 Si la réponse est incorrecte, on revient à la saisie.
- 3840 Il y a huit valeurs à saisir.
- 3850 Le programme saisit la valeur I.
- 3860 Si la réponse est impossible, on repose la question.
- 3870 NEXT de la boucle.
- 3880 On initialise, avec la fonction SYMBOL, le caractère ainsi modifié.
- 3890 On revient à la saisie d'un caractère (toujours pour le graphique en cours).
- 3900 Retour du sous-programme.
- 4000 Le programme demande le numéro du graphique à recopier.
- 4010 On saisit et on vérifie la réponse de l'utilisateur.

```

4020 FOR I=1 TO 4:FOR J=1 TO 8
4030 S(S,I,J)=S(A,I,J)
4040 NEXT J,I
4050 LOCATE 1,22:PRINT CHR$(18)
4060 RETURN
4180 REM -----
4190 REM SAUVEGARDE
4200 CLS:PRINT"APPUYEZ SUR ENTER QUAND LE DATA
CORDER EST PRET POUR L'ENREGISTREMENT"
4210 IF INKEY$<>CHR$(13) THEN 4210
4220 OPENOUT "!CARACTERES"
4230 PRINT#9,N
4240 FOR I=1 TO N
4250 PRINT#9,N$(I)
4260 FOR J=1 TO 4
4270 FOR K=1 TO 8
4280 PRINT#9,S(I,J,K)
4290 NEXT K,J,I
4300 CLOSEOUT
4310 CLS:PRINT"MERCI....!"
4320 END

```

4020 Il y a quatre caractères contenant chacun huit valeurs.
 4030 On recopie le graphique A dans le graphique S.
 4040 NEXT des deux boucles.
 4050 On efface de l'écran le message affiché au début du sous-programme.
 4060 Retour du sous-programme.
 4200 Le programme demande à l'utilisateur de mettre le magnétophone en position enregistrement puis d'enfoncer la touche ENTER.
 4210 On attend que la touche ENTER soit appuyée.
 4220 On ouvre le fichier CARACTERES.
 4230 On écrit sur cassette le nombre de graphiques.
 4240 Il y a N graphiques à sauver.
 4250 On écrit le nom du graphique.
 4260 Il y a quatre caractères par graphique.
 4270 Il y a huit valeurs par caractère.
 4280 On écrit les valeurs contenues dans le tableau S.
 4290 NEXT des trois boucles.
 4300 On ferme le fichier.
 4310 On efface l'écran.
 4320 Fin du programme.

Liste des variables

A	Numéro du graphique à recopier.
A\$	Réponse de l'utilisateur.
B\$()	Valeurs binaires du graphique en cours.
BB\$()	Sauvegarde du tableau B\$().
C	Numéro du caractère choisi pour l'entrée directe en décimale.
F\$()	Dessin des touches de déplacement.
I	Indice de boucle FOR/NEXT.
J	Indice de boucle FOR/NEXT.
K	Indice de boucle FOR/NEXT.
M	Mode en cours.
M\$()	Nom des modes.
N	Nombre de graphiques dans le fichier.
N\$()	Nom des graphiques.
P	Numéro du caractère en cours de modification.
S	Numéro du graphique en cours de modification.
S(,,)	Valeurs décimales des graphiques du fichier.
X & Y	Coordonnées du curseur dans le graphique en cours.

XX & YY	Coordonnées du curseur dans le caractère en cours.
X() & Y()	Coordonnées des valeurs décimales affichées à l'écran.
Z	Indique si l'on transfère les deux caractères du haut ou les deux du bas.
ZZ	Indique, dans le transfert d'un graphique, le numéro de ligne du caractère.

Comment modifier le générateur

Bien qu'il comporte déjà beaucoup de commandes, il peut être intéressant de lui en ajouter de nouvelles. Pour effectuer des modifications dans de bonnes conditions, voici quelques remarques complémentaires.

Tous les graphiques sont stockés en permanence dans le tableau à trois dimensions S. La première dimension concerne le numéro du graphique (de 1 à 100), la deuxième, le numéro du caractère (de 1 à 4), et la troisième, les codes qui composent un caractère (de 1 à 8). Prenons comme exemple le robot du mode d'emploi (il est ici dans le graphique n° 1).

S(1,1,1)	15	224	S(1,2,1)
S(1,1,2)	13	160	S(1,2,2)
S(1,1,3)	15	224	S(1,2,3)
S(1,1,4)	3	128	S(1,2,4)
S(1,1,5)	31	240	S(1,2,5)
S(1,1,6)	63	248	S(1,2,6)
S(1,1,7)	103	204	S(1,2,7)
S(1,1,8)	199	198	S(1,2,8)
S(1,3,1)	199	246	S(1,4,1)
S(1,3,2)	7	224	S(1,4,2)
S(1,3,3)	6	224	S(1,4,3)
S(1,3,4)	6	192	S(1,4,4)
S(1,3,5)	6	192	S(1,4,5)
S(1,3,6)	6	192	S(1,4,6)
S(1,3,7)	6	192	S(1,4,7)
S(1,3,8)	30	240	S(1,4,8)

Figure 1. – Tableau S

La routine en 3100 transfère un graphique du tableau S vers le tableau B\$ sous forme binaire. Le graphique choisi est pointé par la variable S. Après l'appel de cette routine, le tableau B\$ contient les valeurs suivantes (à partir de l'exemple de la figure 1).

B\$(1)	0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0
B\$(2)	0 0 0 0 1 1 0 1 1 0 1 0 0 0 0 0
B\$(3)	0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0
B\$(4)	0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
B\$(5)	0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0
B\$(6)	0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0
B\$(7)	0 1 1 0 0 1 1 1 1 1 0 0 1 1 0 0
B\$(8)	1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0
B\$(9)	1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1
B\$(10)	0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0
B\$(11)	0 0 0 0 0 1 1 0 1 1 1 1 0 0 0 0
B\$(12)	0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0
B\$(13)	0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0
B\$(14)	0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0
B\$(15)	0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0
B\$(16)	0 0 0 1 1 1 1 0 1 1 1 1 0 0 0 0

Figure 2. – Tableau B\$

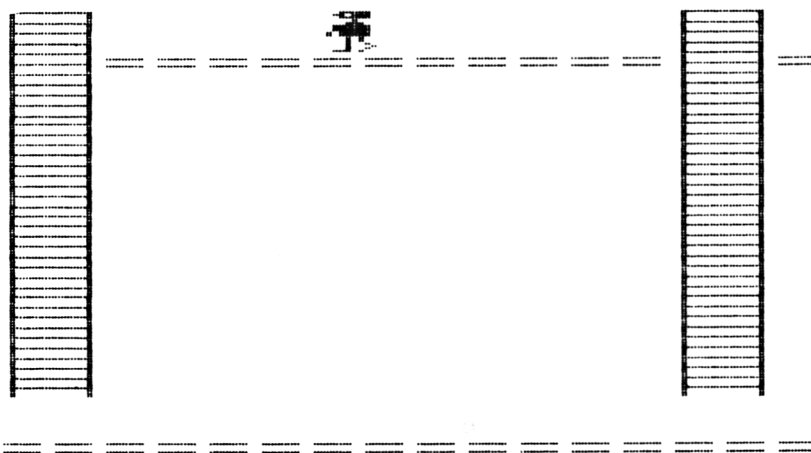
Le tableau de chaînes B\$ est beaucoup plus souple à manipuler pour vos nouvelles commandes que le tableau décimal S qui demande l'utilisation des fonctions logiques OR, AND, NOT ou XOR. La routine en 3300 replace dans le tableau S le contenu du tableau B\$.

Vous devez ajouter les nouvelles routines à partir des lignes 4400. Les lignes qui y font appel seront placées entre 820 et 950 sans effacer celles qui s'y trouvent déjà. Si vous avez effacé l'écran, faites un GOTO 400 au retour de la routine, sinon GOTO 600 suffit pour afficher uniquement le graphique. N'oubliez pas d'ajouter vos nouvelles commandes dans le menu d'aide. Vérifiez bien que l'initiale employée n'existe pas encore. Les lettres restantes sont B G J K O P Q S T U W X Y Z.

3

Exemple d'utilisation

Le programme qui suit donne un exemple d'utilisation des caractères graphiques sur Amstrad. Il est relativement simple dans son exécution, malgré la très longue liste de définition des graphiques. Son but est, avant tout, d'expliquer l'animation de personnages ou d'objets sur l'écran. Les caractères ont été définis avec le générateur en utilisant le plus souvent possible la commande A pour multiplier à volonté le dessin de base de l'androïde.



Programme

```

10 REM EXEMPLE SUR AMSTRAD
20 REM -----
30 DEFINT A-Z
40 MODE 0
60 B$(1)=" "+CHR$(8)+CHR$(10)+" "
70 B$(2)=CHR$(200)+CHR$(201)
180 REM -----
190 REM DEFINITION DES CARACTERES
200 SYMBOL AFTER 128
210 SYMBOL 128,3,2,3,0,1,7,7,11
220 SYMBOL 129,224,160,224,128,192,240,240,232
230 SYMBOL 130,11,11,1,1,1,1,1,3
240 SYMBOL 131,232,232,64,64,64,64,64,112
250 SYMBOL 132,3,2,3,0,3,3,7,7
260 SYMBOL 133,224,160,224,128,192,224,208,208
270 SYMBOL 134,11,11,3,3,2,2,4,7
280 SYMBOL 135,208,200,64,64,64,64,64,112
290 SYMBOL 136,3,3,3,0,1,3,3,3
300 SYMBOL 137,224,176,224,128,192,224,224,224
310 SYMBOL 138,3,3,1,1,1,1,1,1
320 SYMBOL 139,224,224,64,64,64,64,64,224
330 SYMBOL 140,3,3,3,0,1,3,3,3
340 SYMBOL 141,224,176,224,128,192,240,240,240
350 SYMBOL 142,3,3,1,1,2,4,2,1
360 SYMBOL 143,232,232,64,64,64,64,64,112
370 SYMBOL 144,3,3,3,0,1,3,7,7
380 SYMBOL 145,224,176,224,128,192,224,224,224
390 SYMBOL 146,11,11,1,2,2,4,4,7
400 SYMBOL 147,208,208,64,32,32,16,16,28
410 SYMBOL 148,63,35,35,40,63,47,35,35
420 SYMBOL 149,254,234,234,138,254,250,226,226
430 SYMBOL 150,63,35,34,34,63,34,34,34
440 SYMBOL 151,254,226,34,34,254,2,2,2
450 SYMBOL 152,63,43,43,40,63,47,35,35
460 SYMBOL 153,254,226,226,138,254,250,226,226
470 SYMBOL 154,63,35,34,34,63,32,32,32
480 SYMBOL 155,254,226,34,34,254,34,34,34
490 SYMBOL 156,7,13,7,1,3,7,7,7
500 SYMBOL 157,192,192,192,0,128,192,192,192
510 SYMBOL 158,7,7,2,2,2,2,2,7
520 SYMBOL 159,192,192,128,128,128,128,128,128
530 SYMBOL 160,7,13,7,1,3,7,7,7
540 SYMBOL 161,192,192,192,0,128,192,224,224
550 SYMBOL 162,11,11,2,4,4,8,8,56
560 SYMBOL 163,208,208,128,64,64,32,32,224
570 SYMBOL 164,7,13,7,1,3,15,15,15
580 SYMBOL 165,192,192,192,0,128,192,192,192
590 SYMBOL 166,23,23,2,2,2,2,2,14
600 SYMBOL 167,192,192,128,128,64,32,64,128
610 SYMBOL 200,63,32,32,32,63,32,32,32
620 SYMBOL 201,254,2,2,2,254,2,2,2
680 REM -----

```

Commentaires du programme

30 Toutes les variables numériques contiendront des nombres entiers pour une question de rapidité.

40 L'écran est en mode texte 20×25.

60 Chaîne qui effacera le personnage pendant ses déplacements horizontaux.

70 Chaîne qui effacera le personnage pendant ses déplacements verticaux.

200 Les caractères définis commencent au code ASCII 128.

210 }
620 } Définitions des caractères 128 et 201 pour le dessin de l'androïde dans toutes les positions.

700 Il y a 10 positions du personnage.

710 La définition commence au code ASCII 128 ($124 + 1 \times 4 = 128$).

720 Chaque graphique est stocké dans le tableau F\$. Le code 10 permet de descendre d'une ligne, les codes 8 de revenir deux positions en arrière. Les quatre caractères se retrouvent regroupés pour former un graphique.

730 NEXT de la boucle.

800 Les passerelles occupent 16 colonnes sur l'écran.

```

690 REM DEFINITION DES GRAPHIQUES
700 FOR I=1 TO 10
710 Z=124+I*4
720 F$(I)=CHR$(Z)+CHR$(Z+1)+CHR$(10)+CHR$(8)+C
HR$(8)+CHR$(Z+2)+CHR$(Z+3)
730 NEXT
780 REM -----
790 REM ON PLANTE LE DECOR
800 FOR I=1 TO 16
810 LOCATE I,22:PRINT "=":LOCATE I,3:PRINT "="
820 NEXT
830 FOR I=1 TO 19
840 LOCATE 14,I:PRINT CHR$(200);CHR$(201)
850 LOCATE 1,I:PRINT CHR$(200);CHR$(201)
860 NEXT
980 REM -----
990 REM DEPLACEMENT DE L'ANDROIDE
1000 X=1:Y=20
1010 LOCATE X,Y:PRINT F$(1)
1020 FOR I=1 TO 1000:NEXT I
1030 LOCATE X,Y:PRINT F$(2)
1040 FOR I=1 TO 800:NEXT I
1050 D=3:F=6:N=14:IX=1:IY=0:B=1:EX=0:EY=0:T=40
0
1060 GOSUB 1300
1070 LOCATE X,Y:PRINT B$(1)
1080 Y=Y-2:X=X-1
1090 D=6:F=8:N=18:IX=0:IY=-1:B=2:EX=0:EY=1:T=6
00
1100 GOSUB 1300
1110 Y=Y+1
1120 LOCATE X,Y:PRINT B$(2)
1130 X=X-2
1140 D=8:F=11:N=10:IX=-1:IY=0:B=1:EX=1:EY=0:T=
400
1150 GOSUB 1300
1160 LOCATE X+1,Y:PRINT B$(1)
1170 X=X-1
1180 D=6:F=8:N=18:IX=0:IY=1:B=2:EX=0:EY=0:T=60
0
1190 GOSUB 1300
1200 LOCATE X,Y:PRINT B$(2)
1210 GOTO 1000
1280 REM -----
1290 REM DEPLACEMENT
1300 P=D
1310 FOR I=1 TO N
1320 LOCATE X,Y:PRINT F$(P);
1330 FOR J=1 TO T:NEXT J
1340 P=P+1:IF P=F THEN P=D
1350 IF X+EX=0 OR Y+EY=0 THEN 1370
1360 LOCATE X+EX,Y+EY:PRINT B$(B);
1370 Y=Y+IY:X=X+IX
1380 NEXT
1390 RETURN

```

810 On affiche les passerelles.
 820 NEXT de la boucle.
 830 Les échelles occupent 19 lignes sur l'écran.
 840 On affiche la première échelle.
 850 Puis la deuxième.
 860 NEXT des boucles.
 1000 Première position de l'androïde.
 1010 On l'affiche de face.
 1020 Boucle d'attente.
 1030 Il pivote d'un huitième de tour.
 1040 Boucle d'attente.
 1050 Toutes ces variables contiennent les renseignements pour déplacer le personnage vers la droite.
 1060 Ce sous-programme le déplace en fonction des caractéristiques données par la ligne précédente.
 1070 On efface l'androïde.
 1080 On change sa dernière position.
 1090 Toutes ces variables contiennent les renseignements pour déplacer le personnage vers le haut.
 1100 Sous-programme de déplacement.
 1110 On change la position verticale de l'androïde.
 1120 On l'efface.
 1130 On change sa position horizontale.
 1140 Toutes ces variables contiennent les renseignements pour déplacer le personnage vers la gauche.
 1150 Sous-programme de déplacement.
 1160 On efface l'androïde.
 1170 On modifie sa position horizontale.
 1180 Le personnage descend.
 1190 Sous-programme de déplacement.
 1200 On efface l'androïde.
 1210 On revient au début.
 1300 La première position graphique est définie par D (début).
 1310 On déplace le personnage N fois.
 1320 On affiche le personnage en X,Y avec le graphique désigné par P.
 1330 On attend T fois.
 1340 On incrémente le compteur P. S'il est identique à la variable F (fin), il prend la valeur D (début).
 1350 On vérifie que l'on ne sort pas de l'écran.
 1360 On efface le personnage en fonction des incréments EX et EY.
 1370 On incrémente les coordonnées avec les variables IX et IY.
 1380 NEXT de la boucle.
 1390 Fin du sous-programme.

Liste des variables

B	Type d'effacement (tableau B\$).
B\$()	Chaînes pour effacer le personnage.
D	Début des graphiques dans le déplacement.
EX & EY	Incréments horizontal et vertical pour l'effacement du personnage.
F	Fin des graphiques dans le déplacement.
F\$()	Chaînes contenant toutes les positions graphiques du personnage.
I	Indice de boucle FOR/NEXT.
IX & IY	Incréments horizontal et vertical dans le déplacement.
J	Indice de boucle FOR/NEXT.
N	Nombre de déplacements.
P	Pointe, dans le tableau F\$, le prochain graphique à afficher.
T	Longueur de la boucle d'attente.
X & Y	Coordonnées sur l'écran du personnage.
Z	Pointe les caractères graphiques à mettre dans le tableau F\$.

4

Graphiques et langage machine

Pour l'instant, nous n'avons étudié le générateur de caractères que du côté BASIC. Si cela est amplement suffisant dans la plupart des cas, les jeux d'arcades demandent, eux, une rapidité impossible à obtenir avec le BASIC. Nous allons donc examiner les principales caractéristiques de l'écran et du générateur de caractères, ainsi que les routines et les variables système qui permettent de manipuler rapidement des caractères.

A la mise sous tension d'Amstrad, le générateur de caractères se trouve placé dans la ROM inférieure. La table occupe les zones mémoires de &3800 à &3FFF (14336d à 16383d), soit 256 caractères composés de 8 octets. Cette remarque est valable pour tous les modèles existant à l'heure actuelle.

Pour modifier cette table, il faut bien sûr qu'elle soit stockée dans la mémoire vive. C'est le rôle de l'instruction SYMBOL AFTER. Dès que le BASIC rencontre cette instruction, il recopie la partie concernée de la ROM vers la RAM. L'adresse de la nouvelle table est codée sur deux octets aux adresses &AE7D et &AE7E (44669d et 44670d) pour le 464 et &AE60 et &AE61 (44640d et 44641d), pour le 664 et le 6128.

Pour regarder le générateur tel qu'il est dans la RAM, essayez le programme qui suit. Si vous possédez un 664 ou un 6128, il faut modifier la ligne 30 avec les adresses données précédemment.


```

10 REM LE GENERATEUR DE L'AMSTRAD
20 REM -----
30 D=1+PEEK(&AE7D)+256*PEEK(&AE7E)
40 FOR I=D TO 44031
50 IF (I-D)MOD 8=0 THEN PRINT
60 A=PEEK(I)
70 PRINT I;
80 FOR J=7 TO 0 STEP -1
90 IF A AND 2^J THEN PRINT CHR$(143); ELSE PRI
NT" ";
100 NEXT
110 PRINT " ";A
120 NEXT

```

Listing 1

En fait, il existe bien 256 caractères dans le générateur d'Amstrad. Les trente-deux premiers ne sont pas accessibles par CHR\$ en BASIC car ils sont interceptés par l'interpréteur. Refaites tourner le programme en le faisant précéder par la commande SYMBOL AFTER 0 pour découvrir ces caractères. Il est cependant possible d'afficher ces caractères grâce à la routine système en &BB5D.

Les caractères, ainsi transposés dans la RAM, peuvent être modifiés à volonté soit par POKE en BASIC, soit par l'instruction LD en assembleur. Si vous avez déjà une table en mémoire, vous pouvez la recopier par l'instruction LDIR en plaçant dans DE l'adresse de la table d'Amstrad, dans HL l'adresse de votre table et dans BC le nombre (multiplié par 8) de caractères à transférer. Le programme en assembleur qui suit effectue cette opération. Le programme se limite ici à un seul caractère mais rien ne vous empêche d'en transférer 256. Le CALL BBAE place dans HL l'adresse de début du générateur en RAM.

```

10 REM COPIE GENERATEUR (L.M)
20 REM -----
30 MODE 1
40 SYMBOL AFTER 32
50 FOR I=43000 TO 43020
60 READ A:POKE I,A
70 NEXT I
80 CALL 43000
85 DATA &CD,&AE,&BB
90 DATA &EB
100 DATA &21,&05,&A8
110 DATA &01,&08,&00
120 DATA &ED,&B0
130 DATA &C9
140 DATA &6C,&FE,&FE,&FE,&7C,&38,&10,&0

```

Listing 2

43000	CD,AE,BB	CALL BBAE	; Place dans HL l'adresse de la
			; table.
43003	EB	EX DE,HL	; Echange DE et HL.
43004	21,00,A5	LD HL,A805	; HL=adresse de votre table.
43007	01,08,00	LD BC,8	; Nombre d'octets à transférer.
43010	ED,B0	LDIR	; Transfert.
43012	C9	RET	; Retour au BASIC.
43013	6C	DEF	; Matrice du cœur;
43014	FE	DEF	; , ,
43015	FE	DEF	; , ,
43016	FE	DEF	; , ,
43017	7C	DEF	; , ,
43018	38	DEF	; , ,
43019	10	DEF	; , ,
43020	00	DEF	; , ,

Après l'exécution du programme, tous les espaces seront remplacés par des cœurs. Tapez SYMBOL AFTER 32 pour revenir à la normale.

Contrairement à la plupart des micro-ordinateurs, l'Amstrad ne place pas dans la mémoire d'écran le code du caractère à afficher mais y recopie tous les octets qui composent sa matrice. Ceci permet de mélanger facilement texte et graphique haute résolution.

La mémoire d'écran commence généralement à l'adresse &C000, mais elle peut commencer aussi en &0000, &4000 ou &8000. C'est la variable système &B1CB qui donne le poids fort du début d'écran. De plus, à cause du scrolling, il faut ajouter l'offset de la paire d'octets &B1C9 et &B1CA pour obtenir le début réel de l'écran.

Les octets qui composent une ligne haute résolution de l'écran sont placés les uns à côté des autres. La première ligne occupe donc les octets &C000 à &C04F (80 octets de 8 bits = 640 points).

Par contre, la deuxième ligne a un décalage de 2K. Elle occupe donc les octets &C800 à &C84F.

Le tableau suivant donne l'organisation de l'écran en mémoire (on suppose le début en &C000 avec un offset de 0) :

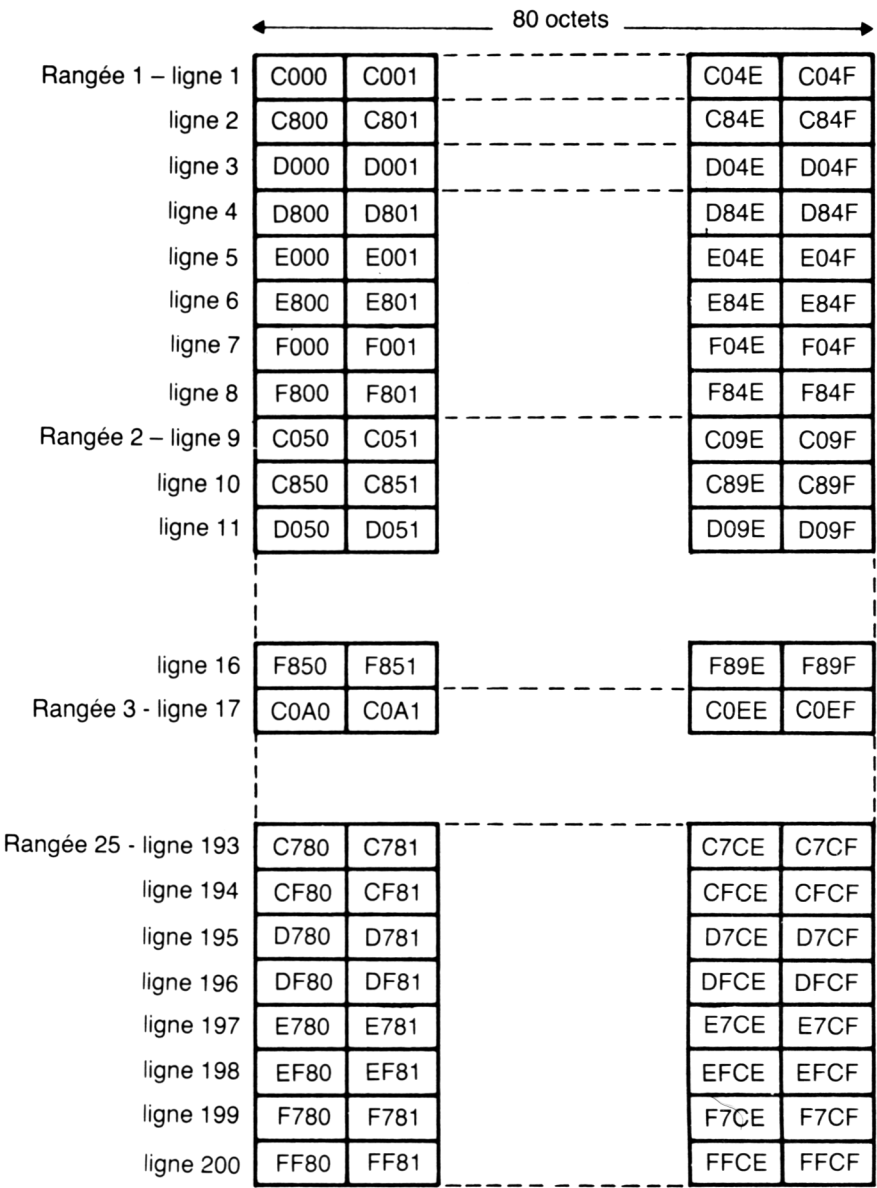


Tableau 1

Pour écrire un caractère sur l'écran, l'Amstrad prend les huit octets dans la matrice et les stocke dans la mémoire d'écran à la position du curseur. Si

nous affichons la lettre A à la position 1,1 en mode 2, l'analyse de la mémoire d'écran nous donne le résultat suivant :

<i>Adresse</i>	<i>Contenu</i>
C000	24
C800	60
D000	102
D800	102
E000	126
E800	102
F000	102
F800	0

Le contenu correspond exactement à la matrice de la lettre A. En mode 2, un point correspond à un pixel sur l'écran puisqu'il n'y a qu'une couleur de fond et une couleur d'encre.

Si nous faisons la même chose en mode 1, il faut analyser deux octets successifs, un point occupant deux pixels sur l'écran. Chaque groupe de quatre points peut prendre quatre couleurs différentes :

<i>Adresse</i>	<i>Contenu</i>	<i>Adresse</i>	<i>Contenu</i>
C000	16	C001	128
C800	48	C801	192
D000	96	D001	96
D800	96	D801	96
E000	112	E001	224
E800	96	E801	96
F000	96	F001	96
F800	0	F801	0

Notre nombre 24 (première ligne de la matrice du caractère A) s'est transformé en deux nombres 16 et 128. En représentation binaire, on obtient ceci :

24 = 00011000
 16 = 00010000
 128 = 10000000

Dans notre exemple, nous étions en PEN 1. Si nous avons changé les couleurs avec PEN 2 et PEN 3, nous aurions obtenu respectivement les chiffres 1-8 et 17-136 :

PEN 2 1 = 00000001 8 = 00001000
 PEN 3 17 = 00010001 136 = 10001000

Pour bien comprendre le problème des couleurs, le programme qui suit va vous permettre de mettre des couleurs dans vos graphiques. Vous devez donner successivement le code du graphique suivi de la couleur des quatre points à gauche et de la couleur des quatre points à droite (entre 0 et 3), chaque élément étant séparé par une virgule. A titre d'exemple, vous pouvez essayer la séquence suivante :

```
255,1,2
129,1,2
129,1,2
255,1,2
129,1,2
129,1,2
255,1,2
0,1,2
```

```
10 REM GRAPHIQUE MULTI-COULEURS
20 REM -----
30 MODE 1
40 E=&C000
50 FOR I=1 TO 8
60 LOCATE 1,10:PRINT CHR$(18); "L. "; I;
70 INPUT"(NB, COULEUR 1, COULEUR 2)"; N, C(1), C(2)
80 N(1)=(N AND 240)/16: N(2)=N AND 15
90 FOR J=1 TO 2
100 ON C(J) GOTO 110, 130, 120
110 N(J)=N(J)*16: GOTO 130
120 N(J)=N(J)*16+N(J)
130 POKE E-1+J, N(J)
140 LOCATE 20+J*5, I:PRINT N(J)
150 NEXT
160 E=E+&800
170 NEXT
180 LOCATE 1, 15
```

Listing 3

Le principe est le même pour le mode 0 mais un octet de la matrice prend maintenant quatre octets dans l'écran. Dans ce mode, deux points occupent huit pixels et offrent une palette de seize couleurs. En reprenant notre exemple du caractère A à la position 1,1, on obtient les octets suivants avec PEN 1 :

Adresse	Adresse+0	Adresse+1	Adresse+2	Adresse+3
C000	0	64	128	0
C800	0	192	192	0
D000	64	128	64	128

Adresse	Adresse+0	Adresse+1	Adresse+2	Adresse+3
D800	64	128	64	128
E000	64	192	192	128
E800	64	128	64	128
F000	64	128	64	128
F800	0	0	0	0

Si vous vouliez afficher directement dans la mémoire d'écran un caractère dans le mode 0, votre programme devrait transférer 32 octets pour un caractère simple et 128 pour un graphique composé de quatre caractères. Il faut donc utiliser au maximum les routines existantes de l'Amstrad. La liste qui suit vous donne les principaux points d'entrée des routines système qui utilisent l'écran, avec leurs buts, les registres à modifier avant l'appel (entrée) et les registres modifiés au retour de la routine (sortie), ainsi que les adresses des variables système utilisées par ces routines :

- **BB5A** : Envoie sur l'écran un caractère ou un code de contrôle (de 0 à 1F) à la position courante du curseur.
Entrée : A contient le caractère à afficher.
Sortie : Les registres sont préservés.

- **BB5D** : Identique à BB5A mais les codes de contrôle sont affichés avec leurs graphiques correspondants (voir listing 1).
Entrée : A contient le caractère à afficher.
Sortie : AF, BC, DE et HL sont modifiés.

- **BB60** : Lit le caractère sous le curseur.
Entrée : Aucune condition.
Sortie : Si le caractère est reconnu, le drapeau CARRY est vrai et A contient le code du caractère, sinon CARRY est faux et A=0.

- **BB66** : Fixe la taille de la fenêtre courante.
Entrée : H contient la colonne des deux coins à gauche.
D contient la colonne des deux coins à droite.
L contient la ligne des deux coins en haut.
E contient la ligne des deux coins en bas.
Sortie : AF, BC, DE et HL sont modifiés.

- **BB69** : Lit la taille de la fenêtre courante.
Entrée : Aucune condition.
Sortie : H contient la colonne des deux coins à gauche.
D contient la colonne des deux coins à droite.

L contient la ligne des deux coins en haut.
 E contient la ligne des deux coins en bas.
 Si la fenêtre couvre tout l'écran, le drapeau CARRY est faux,
 sinon il est vrai. A est modifié.

- **BB6C** : Efface la fenêtre courante.
 Entrée : Aucune condition.
 Sortie : AF, BC, DE et HL sont modifiés.

- **BB6F** : Modifie la position horizontale du curseur.
 Entrée : A contient le numéro de la colonne.
 Sortie : A et HL sont modifiés.

- **BB72** : Modifie la position verticale du curseur.
 Entrée : A contient le numéro de ligne.
 Sortie : A et HL sont modifiés.

- **BB75** : Modifie la position du curseur.
 Entrée : H contient le numéro de la colonne.
 L contient le numéro de la ligne.
 Sortie : A et HL sont modifiés.

- **BB78** : Demande la position du curseur.
 Entrée : Aucune condition.
 Sortie : H contient le numéro de la colonne.
 L contient le numéro de la ligne.
 A contient le nombre de SCROLL (défilement) déjà effectué
 dans la fenêtre courante.

- **BB7B** : Autorisation d'afficher le curseur.
 Entrée : Aucune condition.
 Sortie : A est modifié.

- **BB7E** : Interdiction d'afficher le curseur.
 Entrée : Aucune condition.
 Sortie : A est modifié.

- **BB87** : Teste si une position donnée du curseur est à l'intérieur d'une
 fenêtre.
 Entrée : H contient le numéro de colonne à tester.
 L contient le numéro de ligne à tester.
 Sortie : Si l'affichage à cette position ne doit entraîner aucun SCROLL
 (défilement), le drapeau CARRY est vrai et B est modifié.

S'il y a un défilement vers le haut, CARRY est faux et B contient FE. S'il y a un défilement vers le bas, CARRY est faux et B contient 0. Dans tous les cas, A est modifié.

- *BB90* : Définit la couleur des caractères.
 Entrée : A contient le numéro d'encre.
 Sortie : A et HL sont modifiés.

- *BB93* : Demande la couleur des caractères.
 Entrée : Aucune condition.
 Sortie : A contient le numéro d'encre.

- *BB96* : Définit la couleur de fond des caractères.
 Entrée : A contient le numéro d'encre.
 Sortie : A et HL sont modifiés.

- *BB99* : Demande la couleur de fond des caractères.
 Entrée : Aucune condition.
 Sortie : A contient le numéro d'encre du papier.

- *BB9C* : Echange la couleur des caractères avec la couleur de fond.
 Entrée : Aucune condition.
 Sortie : A et HL sont modifiés.

- *BB9F* : Détermine le mode opaque ou transparent.
 Entrée : A contient 0 pour le mode opaque. A est différent de 0 pour le mode transparent.
 Sortie : A et HL sont modifiés.

- *BBA2* : Demande le mode d'écriture du fond.
 Entrée : Aucune condition.
 Sortie : A contient 0 pour le mode opaque. A est différent de 0 pour le mode transparent.

- *BBA5* : Demande l'adresse de la matrice d'un caractère.
 Entrée : A contient le code du caractère à rechercher.
 Sortie : Si la matrice est définie par l'utilisateur, le drapeau CARRY est vrai. Si la matrice est dans la ROM, CARRY est faux. HL contient l'adresse de cette matrice. A est modifié.

- *BBA8* : Donne l'adresse de la matrice d'un caractère défini par l'utilisateur.

- Entrée : A contient le caractère à définir. HL contient l'adresse de la matrice.
- Sortie : Si le caractère est redéfinissable par l'utilisateur, le drapeau CARRY est vrai, sinon il est faux. Tous les registres sont modifiés.
- **BBAB** : Donne l'adresse de la table d'une matrice de caractères définie par l'utilisateur.
- Entrée : DE contient le premier caractère de la table.
HL contient l'adresse de la nouvelle table.
- Sortie : S'il n'y a pas de table définie, le drapeau CARRY est faux, A et HL sont modifiés. Sinon CARRY est vrai, A contient le premier caractère de l'ancienne table et HL son adresse. BC et DE sont modifiés.
- **BBAE** : Demande l'adresse et le premier caractère de la table définie par l'utilisateur.
- Entrée : Aucune condition.
- Sortie : Si aucune table n'est définie, le drapeau CARRY est faux, A et HL sont modifiés. Sinon CARRY est vrai, A contient le premier caractère de la table et HL son adresse.

Adresses 464	Adresses 664 et 6128	Nombre d'octets	Explications
AE7D	AE60	2	Utilisé par l'instruction BASIC SYMBOL.
B1C8	B7C3	1	Mode d'écran.
B1C9	B7C4	2	Offset de l'écran (de 0 à 7FF).
B1CB	B7C6	1	Octet de poids fort du début de la mémoire écran.
B1DA	B7D4	32	Couleurs des encres (deux octets par couleur).
B1FC	B7F7	1	Utilisé par BORDER.
B285	B726	1	Position ligne du curseur.
B286	B727	1	Position colonne du curseur.
B287	B728	1	Indicateur de fenêtre.
B288	B729	1	Ligne des deux coins en haut de la fenêtre courante.

<i>Adresses 464</i>	<i>Adresses 664 et 6128</i>	<i>Nombre d'octets</i>	<i>Explications</i>
B289	B72A	1	Colonne des deux coins à gauche de la fenêtre courante.
B28A	B72B	1	Ligne des deux coins en bas de la fenêtre courante.
B28B	B72C	1	Colonne des deux coins à droite de la fenêtre courante.
B28D	B72E	1	0=curseur autorisé, 255=curseur interdit.
B28F	B72F	1	Encre courante pour le crayon.
B290	B730	1	Encre courante pour le papier.
B291	B731	1	0=affichage du fond permis, 255=affichage du fond interdit.
B294	B734	2	Premier caractère et état de la table des matrices définie par l'utilisateur.
B296	B736	2	Adresse de la table des matrices définie par l'utilisateur.

Les programmes qui suivent donnent des exemples d'utilisation des routines système de l'Asmtrad. Elles sont à utiliser sous la même forme que le listing 2.

– *Définir la couleur de l'encre et celle du papier :*

43000	3E,00	LD A,0	; encre bleue
43002	CD,90,BB	CALL BB90	; routine encre
43005	3E,01	LD A,1	; papier jaune
43007	CD,96,BB	CALL BB96	; routine papier
43010	C9	RET	; retour BASIC

– *Définir la position du curseur :*

43000	21,0A,0A	LD HL,0A0A	; curseur en 10,10
43003	CD,75,BB	CALL BB75	; détermine la position
43006	C9	RET	; retour BASIC

– *Afficher un caractère :*

43000	3E,F9	LD A,F9	; caractère 249
43002	CD,5A,BB	CALL BB5A	; affiche le caractère
43005	C9	RET	; retour BASIC

– *Afficher un message (le message se termine par FF) :*

43000	21,05,A8	LD HL,A805	; adresse message
43003	7E	LD A,(HL)	; charge un caractère
43004	FE,FF	CP FF	; fin message
43006	C8	RET Z	; retour BASIC si fin
43007	CD,5A,BB	CALL BB5A	; affiche un caractère
43010	23	INC HL	; caractère suivant
43011	18,F6	JR -8	; on continue
43013	1F,0A,0A	DEF	; curseur en 10,10
43016	42,4F,4E	DEF	; message
43019	4A,4F,55	DEF	; "BONJOUR"
43022	52	DEF	;
43023	FF	DEF	; fin message

– *Définir une fenêtre de 10,10 à 20,20 :*

43000	CD,6C,BB	CALL BB6C	; CLS
43003	21,0A,0A	LD HL,0A0A	; ligne/colonne haut/gauche
43006	11,14,14	LD DE,1414	; ligne/colonne bas/droite
43009	CD,66,BB	CALL BB66	; définit la fenêtre
43012	C9	RET	; retour BASIC

Ces routines sont la base des programmes en assembleur, car elles permettent d'afficher un ou plusieurs caractères, de changer les couleurs et de définir des fenêtres. Utilisez-les le plus possible dans vos programmes "BASIC-assembleur" pour en accélérer l'exécution. N'oubliez pas qu'il existe des caractères de contrôle que l'Amstrad interprète à l'appel de la routine en &BB5A. Leur interprétation est bien sûr un peu plus lente qu'un appel direct de la routine correspondante, mais largement suffisante dans la plupart des cas. Le programme "afficher un message" vous donne un exemple d'utilisation des codes de contrôle (ici l'adressage curseur, code &1F). Les codes de contrôle sont répertoriés au chapitre 9 du guide utilisateur.

Ecrire des programmes en assembleur est beaucoup plus simple qu'il n'y paraît. Les routines "toutes prêtes" de l'Amstrad permettent de suivre un raisonnement identique à celui du BASIC, les variables étant simplement remplacées par des octets en mémoire ou les registres du microprocesseur. Armez-vous de patience, c'est en programmant qu'on devient programmeur.

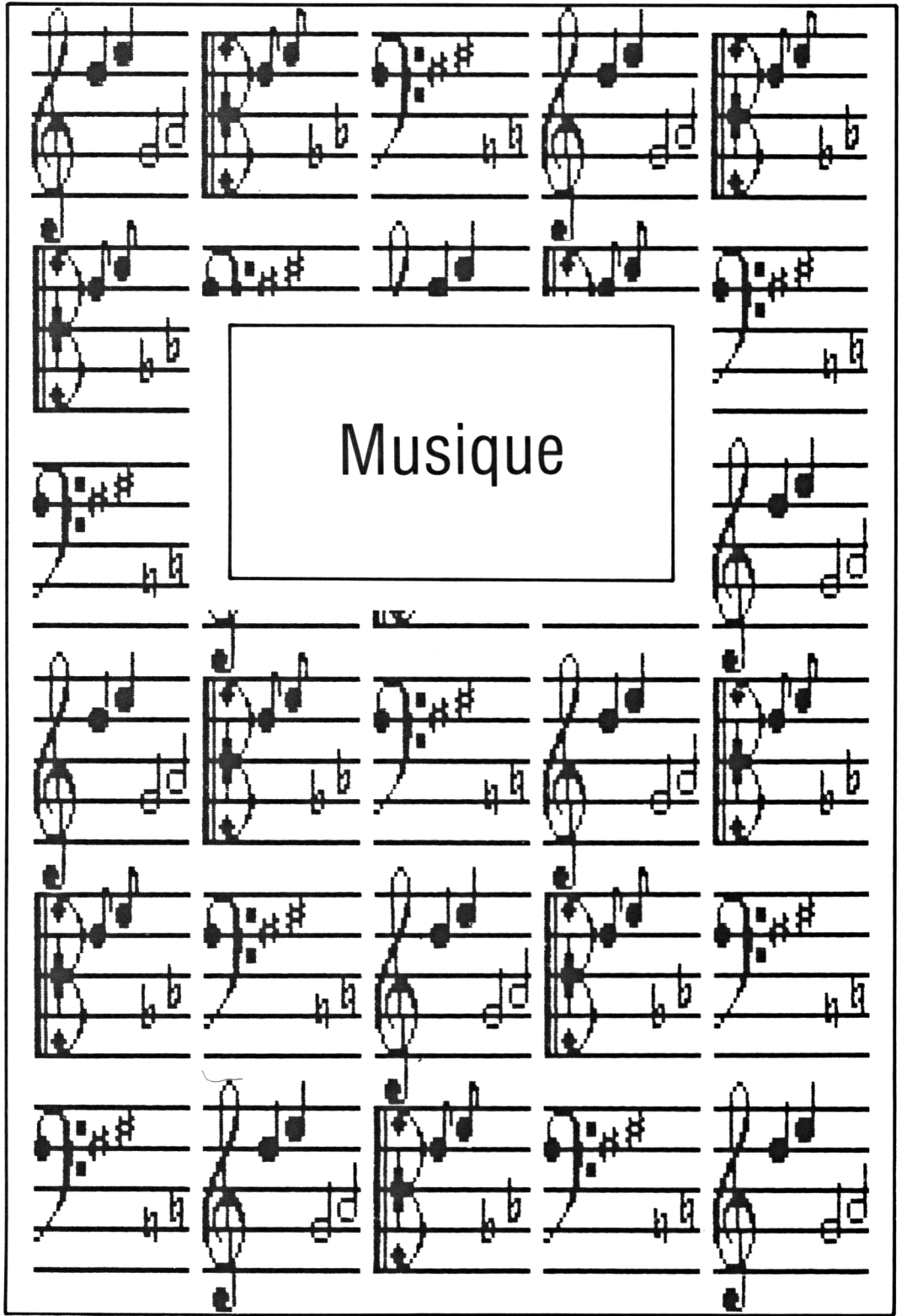
Seconde partie

--- Graphiques prêts à l'emploi

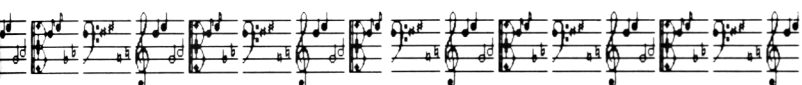
Dans les pages qui suivent, vous allez trouver toute une série de graphiques déjà réalisés et groupés par thèmes.

Chaque graphique est représenté sur une grille et accompagné de ses valeurs décimales et hexadécimales. Ces données vous permettront de définir directement des caractères dans vos programmes, soit par l'instruction SYMBOL en BASIC, soit directement dans la mémoire en langage machine.

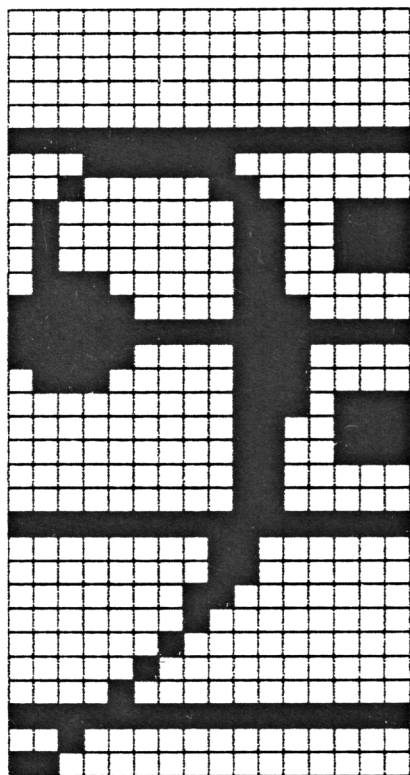
Si vous désirez conserver dans votre fichier certains des graphiques présentés ici, vous devrez utiliser la commande A du générateur et entrer au clavier la liste des valeurs décimales.

The background of the page is a grid of musical staves. Each staff contains a sequence of notes and rests, primarily in treble and bass clefs, with various accidentals (sharps, flats, and naturals). The notation is arranged in a repeating pattern across the page, creating a musical texture. In the center of the page, there is a white rectangular box with a black border.

Musique



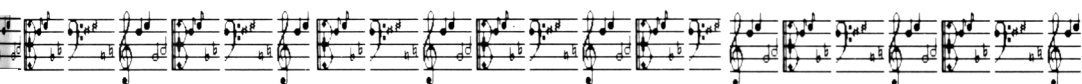
CLEF DE FA



Dec.

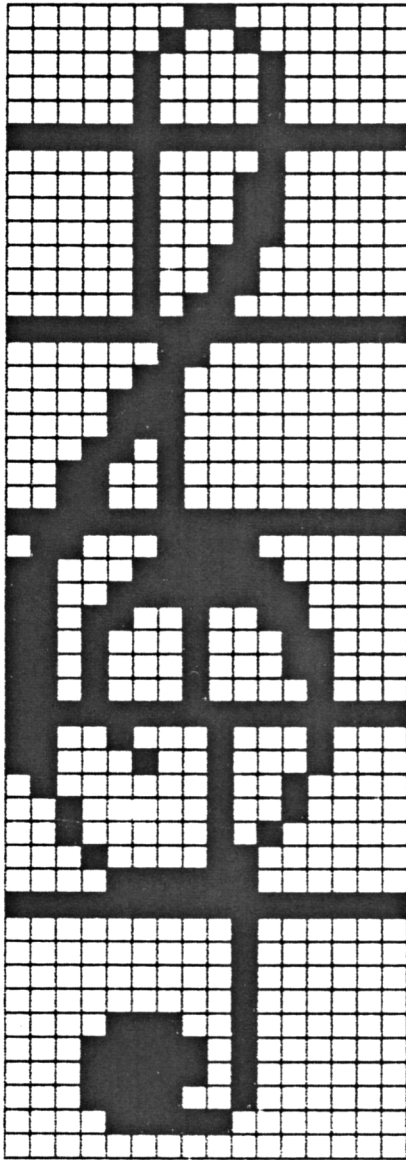
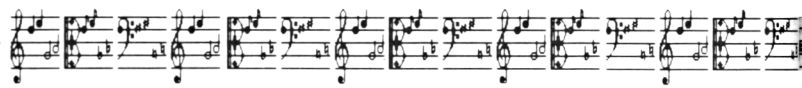
Hex.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
2	5	0	F
5	5	1	F
3	1	2	0
3	2	1	0
6	4	1	0
6	4	1	0
6	4	1	0
1	1	0	0
2	4	1	0
5	5	2	F
5	5	1	F
2	4	1	F
1	1	7	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
2	5	0	F
0	0	0	0
0	0	0	0
1	1	1	0
2	0	2	0
4	0	4	0
8	0	8	0
2	5	F	F
3	2	0	0
1	9	C	0
2	0	0	0





CLEF DE SOL

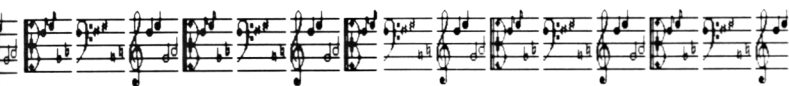


Dec.

Hex.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

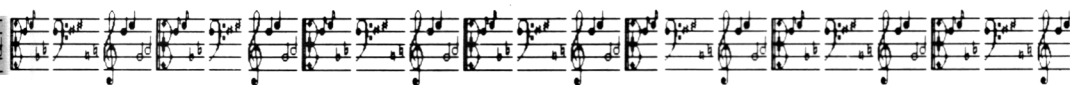
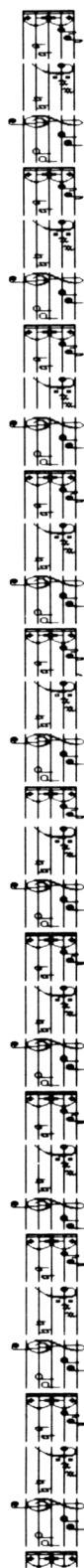
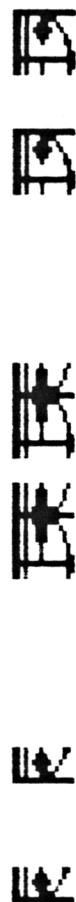
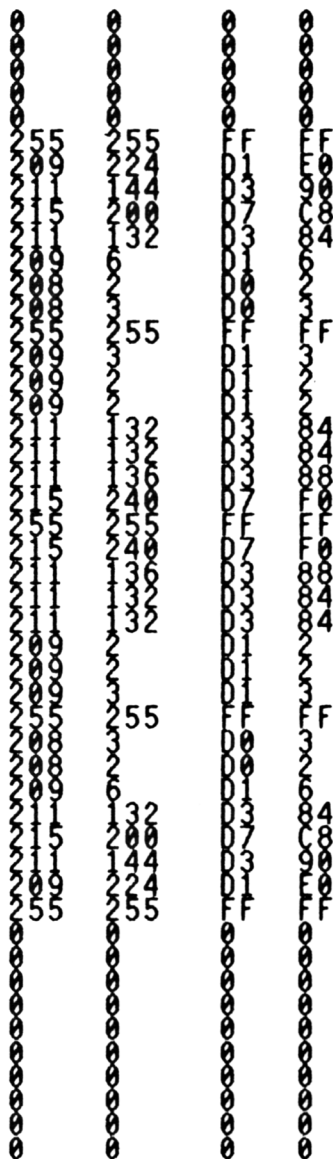
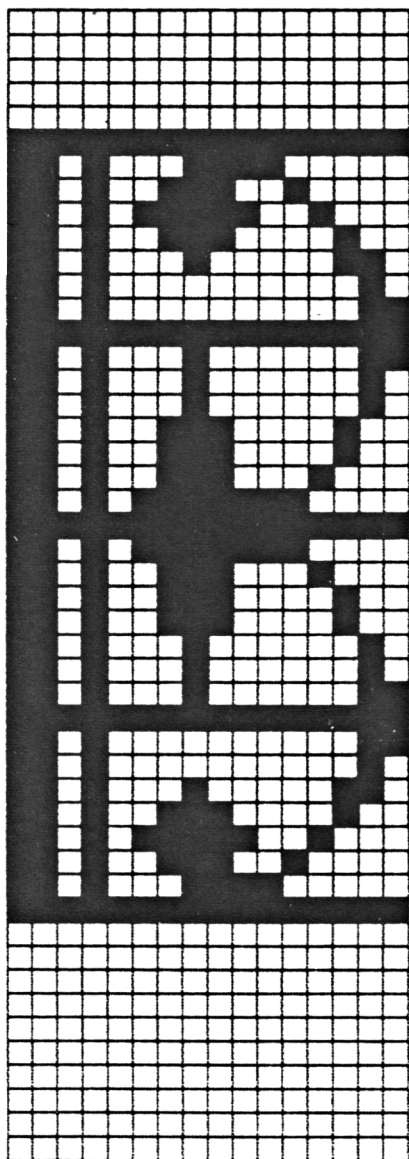




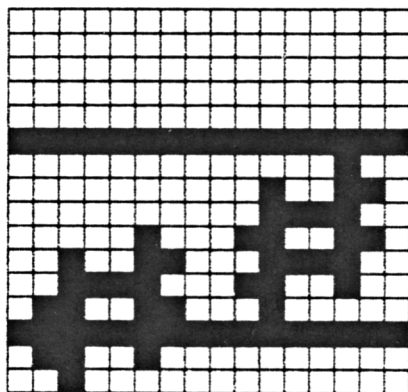
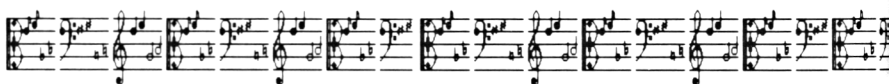
CLEF D'UT

Dec.

Hex.



DIÈSE



Dec.

Hex.

0
0
0
0
0
2
0
0
4
3
6
2
1
2
3
2

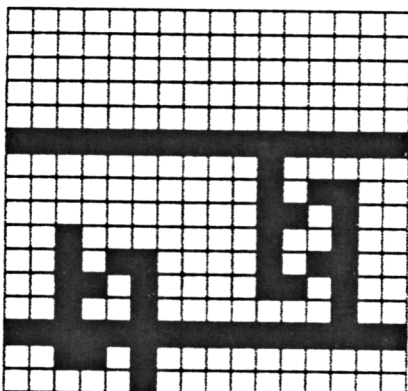
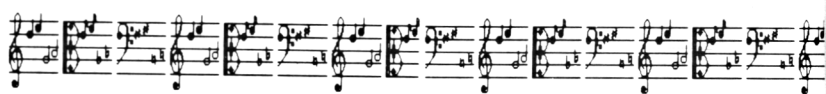
0
0
0
0
0
2
4
3
0
0
0
2
1
2
3
2

0
0
0
0
0
F
F
0
4
0
0
0
F
F
F
6
2
0

0
0
0
0
0
F
F
0
4
0
0
0
F
F
F
6
2
0



BÉCARRE



Dec.

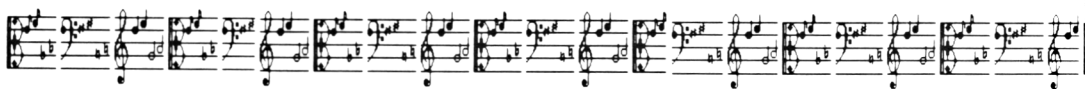
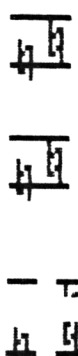
Hex.

0
0
0
0
0
2
0
0
4
3
6
2
1
2
3
2

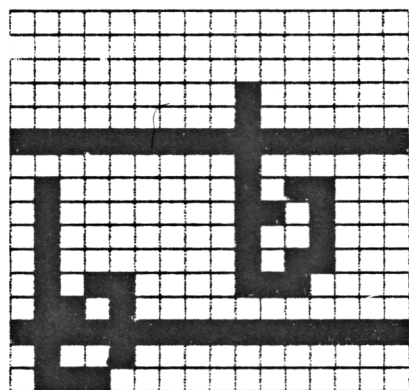
0
0
0
0
0
2
4
3
0
0
0
2
1
2
3
2

0
0
0
0
0
F
F
0
4
0
0
0
F
F
F
6
2
0

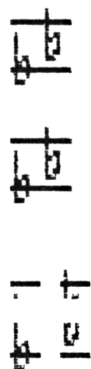
0
0
0
0
0
F
F
0
4
0
0
0
F
F
F
6
2
0



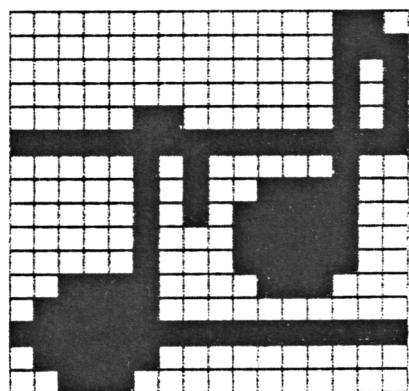
BÉMOL



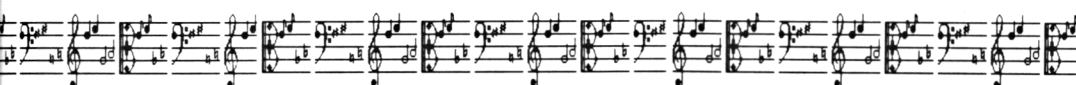
Dec.	Hex.
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
17	11
18	12
19	13
20	14
21	15
22	16
23	17
24	18
25	19
26	1A
27	1B
28	1C
29	1D
30	1E
31	1F
32	20
33	21
34	22
35	23
36	24
37	25
38	26
39	27
40	28
41	29
42	2A
43	2B
44	2C
45	2D
46	2E
47	2F
48	30
49	31
50	32
51	33
52	34
53	35
54	36
55	37
56	38
57	39
58	3A
59	3B
60	3C
61	3D
62	3E
63	3F
64	40
65	41
66	42
67	43
68	44
69	45
70	46
71	47
72	48
73	49
74	4A
75	4B
76	4C
77	4D
78	4E
79	4F
80	50
81	51
82	52
83	53
84	54
85	55
86	56
87	57
88	58
89	59
90	5A
91	5B
92	5C
93	5D
94	5E
95	5F
96	60
97	61
98	62
99	63
100	64
101	65
102	66
103	67
104	68
105	69
106	6A
107	6B
108	6C
109	6D
110	6E
111	6F
112	70
113	71
114	72
115	73
116	74
117	75
118	76
119	77
120	78
121	79
122	7A
123	7B
124	7C
125	7D
126	7E
127	7F
128	80
129	81
130	82
131	83
132	84
133	85
134	86
135	87
136	88
137	89
138	8A
139	8B
140	8C
141	8D
142	8E
143	8F
144	90
145	91
146	92
147	93
148	94
149	95
150	96
151	97
152	98
153	99
154	9A
155	9B
156	9C
157	9D
158	9E
159	9F
160	100
161	101
162	102
163	103
164	104
165	105
166	106
167	107
168	108
169	109
170	10A
171	10B
172	10C
173	10D
174	10E
175	10F
176	110
177	111
178	112
179	113
180	114
181	115
182	116
183	117
184	118
185	119
186	11A
187	11B
188	11C
189	11D
190	11E
191	11F
192	120
193	121
194	122
195	123
196	124
197	125
198	126
199	127
200	128
201	129
202	12A
203	12B
204	12C
205	12D
206	12E
207	12F
208	130
209	131
210	132
211	133
212	134
213	135
214	136
215	137
216	138
217	139
218	13A
219	13B
220	13C
221	13D
222	13E
223	13F
224	140
225	141
226	142
227	143
228	144
229	145
230	146
231	147
232	148
233	149
234	14A
235	14B
236	14C
237	14D
238	14E
239	14F
240	150
241	151
242	152
243	153
244	154
245	155
246	156
247	157
248	158
249	159
250	15A
251	15B
252	15C
253	15D
254	15E
255	15F



CROCHES

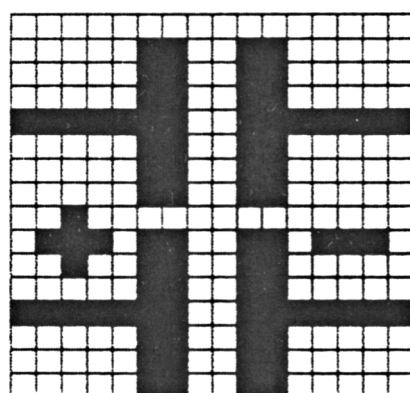


Dec.	Hex.
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
17	11
18	12
19	13
20	14
21	15
22	16
23	17
24	18
25	19
26	1A
27	1B
28	1C
29	1D
30	1E
31	1F
32	20
33	21
34	22
35	23
36	24
37	25
38	26
39	27
40	28
41	29
42	2A
43	2B
44	2C
45	2D
46	2E
47	2F
48	30
49	31
50	32
51	33
52	34
53	35
54	36
55	37
56	38
57	39
58	3A
59	3B
60	3C
61	3D
62	3E
63	3F
64	40
65	41
66	42
67	43
68	44
69	45
70	46
71	47
72	48
73	49
74	4A
75	4B
76	4C
77	4D
78	4E
79	4F
80	50
81	51
82	52
83	53
84	54
85	55
86	56
87	57
88	58
89	59
90	5A
91	5B
92	5C
93	5D
94	5E
95	5F
96	60
97	61
98	62
99	63
100	64



Électronique

CAPACITÉS



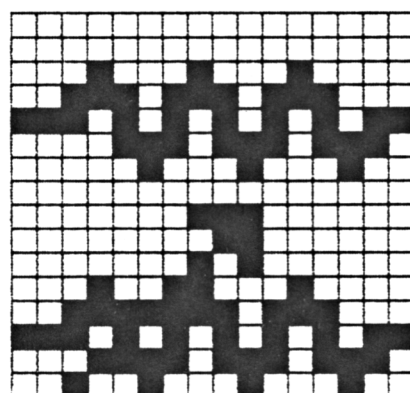
Dec.

Hex.

00000000	00000000	00000000	00000000
54	7	E	00000000
2	0	00000000	00000000
8	1	00000000	00000000
54	7	E	00000000
00000000	00000000	00000000	00000000



RÉSISTANCES



Dec.

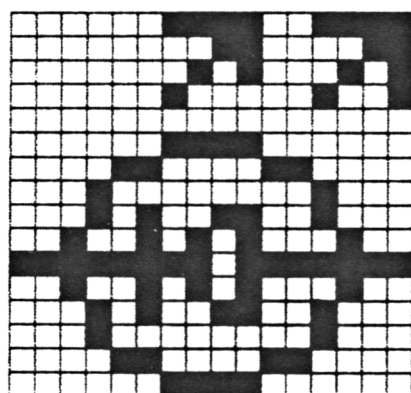
Hex.

00000000	00000000	00000000	00000000
7	6	1	00000000
5	4	BA	00000000
9	8	00000000	00000000
3	7	00000000	00000000
4	0	00000000	00000000
4	8	00000000	00000000
0	0	00000000	00000000
0	0	00000000	00000000
0	0	00000000	00000000
0	0	00000000	00000000
0	0	00000000	00000000
0	0	00000000	00000000
0	0	00000000	00000000
0	0	00000000	00000000
0	0	00000000	00000000



LED

LED



Dec.

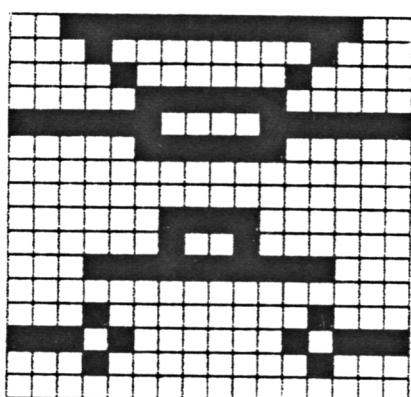
Hex.

3	207	3	CF
1	105	0	C3
0	69	0	45
0	9	0	9
1	192	0	C0
2	48	0	30
0	8	0	8C
2	200	1	44
2	68	2	55
2	127	2	7F
2	68	2	44
2	200	2	C8
0	8	1	80
1	48	1	30
2	192	0	C0



HAUT-PARLEUR

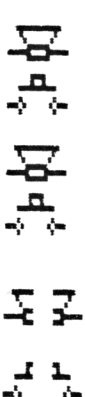
HAUT-PARLEUR



Dec.

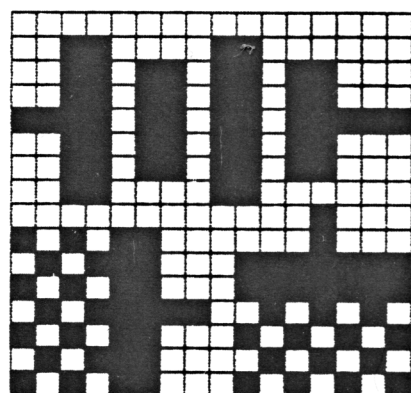
Hex.

63	252	3F	FC
16	8	10	81
8	16	8	10
7	224	7	FE
252	63	FC	3F
7	224	7	FE
0	0	0	0
0	0	0	0
3	192	3	C0
3	48	1F	40
0	8	0	80
0	8	0	80
2	3	1E	07
16	0	10	0
2	0	10	0
0	0	0	0



HAUT-PARLEUR

BATTERIE/MASSE



Dec.

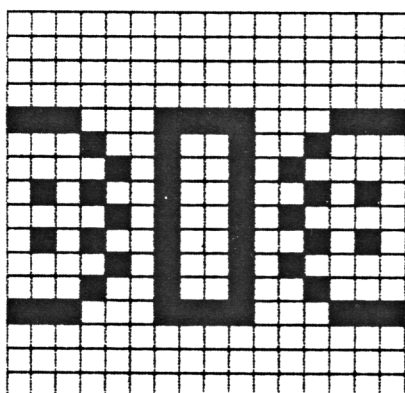
0
48
54
54
246
54
48
0
172
0
172
0
172
172

Hex.

0
33333333
33333333
33333333
33333333
33333333
33333333
33333333
33333333
33333333
33333333
33333333
33333333
33333333
33333333
33333333



TRANSFORMATEUR

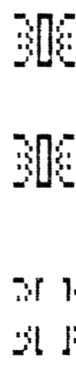


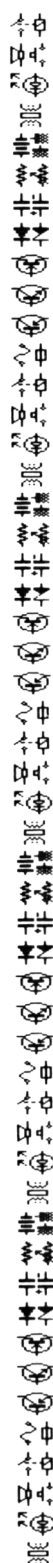
Dec.

0
0
0
0
227
18
10
82
10
82
10
18
227
0
0
0

Hex.

0
0
0
0
E3
12
A2
52
A2
52
A2
12
E3
0
0
0





Hex.

0	2	8
1	4	6
6	7	5
1	7	8
1	6	8
1	2	8
1	2	8
0	2	8
1	4	8
2	6	0
1	5	9
8	0	9
2	4	0
0		

148

9 7 5 9 1 7 0 0 9 0 0 0 1 1 A B 4 0 0 9
0 0 9 7 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0





Hex.

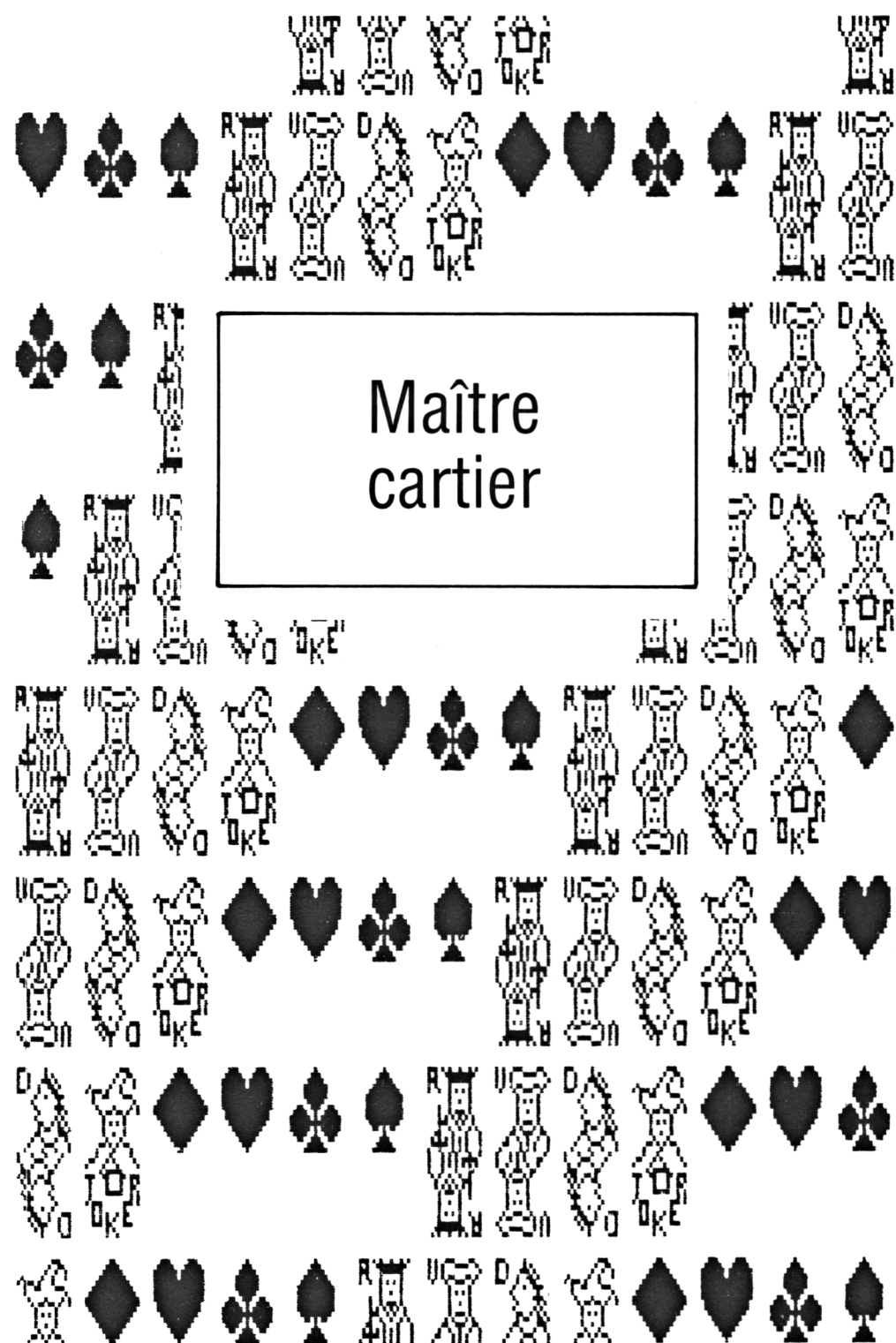
00011117400212120
00033332800465640
00033334000550

SECRET

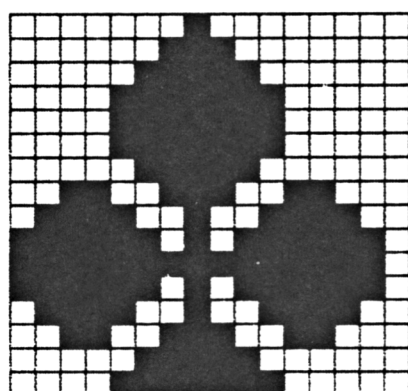
SECRET
SECRET



Maître
cartier



TRÈFLE



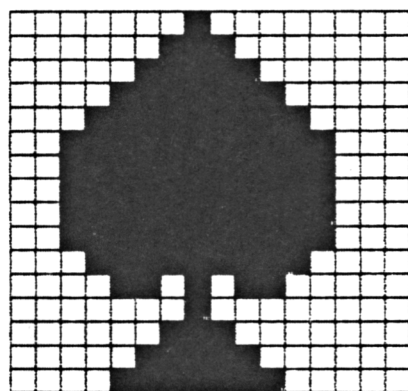
Dec.

Hex.

1	0	1	0
3	128	3	0000
7	192	7	0000
15	224	15	0000
15	224	15	0000
15	224	15	0000
15	224	15	0000
7	192	7	0000
51	152	33	0000
121	60	79	0000
121	126	79	0000
121	254	79	0000
121	126	79	0000
121	60	79	0000
51	152	33	0000
7	192	7	0000
15	224	15	0000



PIQUE



Dec.

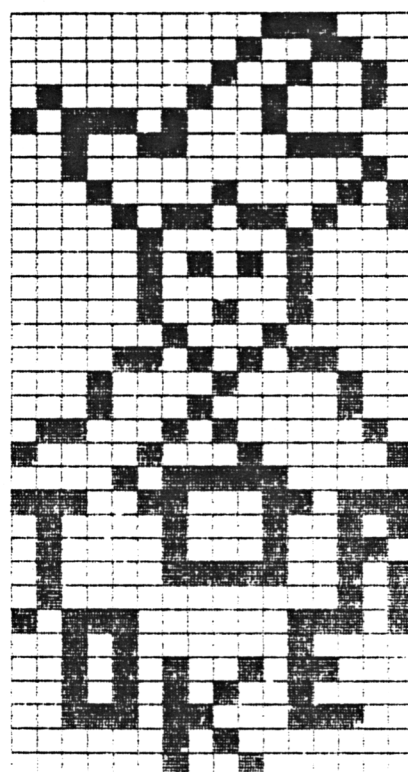
Hex.

1	0	1	0
3	128	3	0000
7	192	7	0000
15	224	15	0000
15	224	15	0000
15	224	15	0000
15	224	15	0000
7	192	7	0000
51	152	33	0000
121	60	79	0000
121	126	79	0000
121	254	79	0000
121	126	79	0000
121	60	79	0000
51	152	33	0000
7	192	7	0000
15	224	15	0000





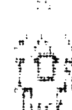
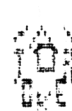
JOKER



Dec.

Hex.

0	56	0	38
00	76	00	4C
00	146	00	92
65	34	41	22
18	32	BA	20
38	28	26	1C
32	2	20	2
16	133	10	85
11	105	8	69
4	16	4	10
5	80	5	50
4	16	4	10
4	144	4	90
2	32	2	20
13	88	D	58
16	132	10	84
17	4	11	4
98	130	62	82
13	65	84	41
11	224	B	F0
230	55	F6	37
66	37	42	25
66	38	42	26
67	229	43	E5
64	5	40	5
184	29	B8	10
40	16	28	10
42	88	2A	58
42	144	2A	90
59	28	3B	1C
2	128	2	80
2	64	2	40

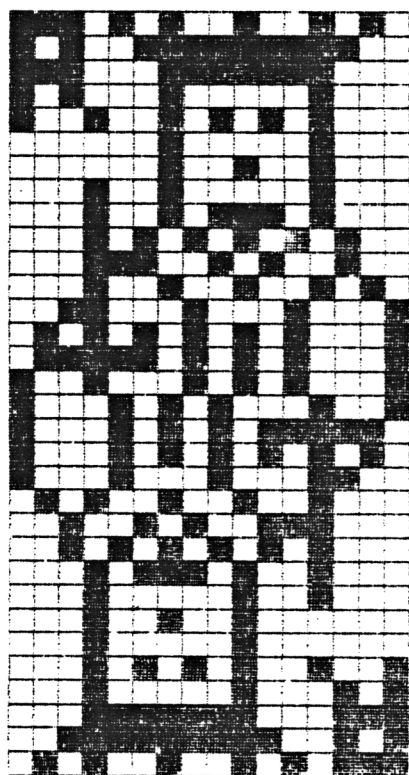




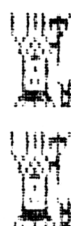
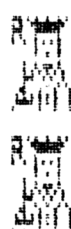
ROI

Dec.

Hex.



234	74	EA	4A
167	252	A7	FC
227	248	E3	F8
162	8	A2	8
146	168	92	A8
2	8	2	8
2	72	2	48
18	8	12	8
18	232	12	F8
21	84	15	54
28	164	1C	A4
18	74	12	4A
49	17	31	11
85	81	55	51
12	81	7D	51
14	81	91	51
13	137	8A	89
13	190	8A	BE
13	170	8A	AA
13	140	88	8C
82	72	52	48
37	56	25	38
42	168	2A	A8
23	72	17	48
16	72	10	48
18	64	12	40
16	64	10	40
21	73	15	49
16	69	10	45
31	199	1F	C7
63	229	3F	E5
82	87	52	57



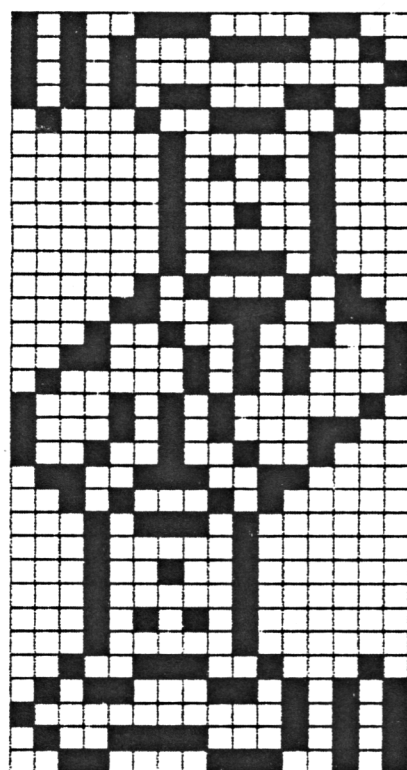


VALET



Dec.

Hex.

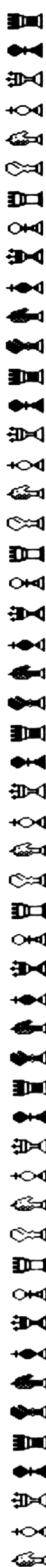


167	12	A7	C
168	242	A88	F2
168	1	A88	F1A
168	26	AB8	4
168	228	44	8A8
68	8	2	848
2	168	2	88
2	8	2	848
2	72	2	88
2	8	2	848
2	232	2	88
2	0	2	848
2	230	2	88
2	730	2	848
12	81	12	88
118	81	11	848
49	81	11	88
65	81	11	848
33	81	11	88
11	81	11	848
14	81	11	88
10	81	11	848
3	81	11	88
40	81	11	848
23	81	11	88
16	81	11	848
18	81	11	88
10	81	11	848
21	81	11	88
16	81	11	848
30	81	11	88
9	81	11	848
88	81	11	88
22	81	11	848
17	81	11	88
9	81	11	848
48	81	11	88

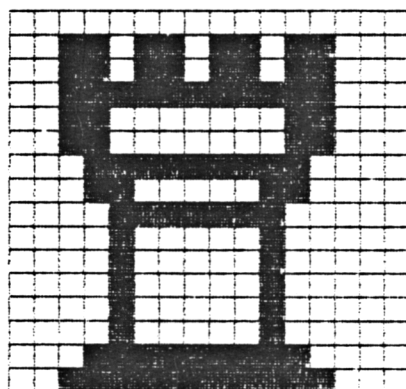


Échec
et mat





TOUR BLANCHE



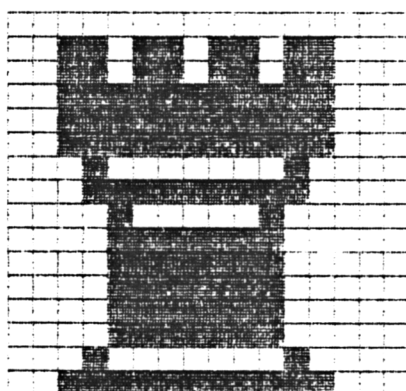
Dec.

Hex.

0	0	0	0
54	216	36	DD8
54	216	36	DD8
63	248	3F	F88
48	24	3F	F88
48	24	3F	F88
31	240	1F	F88
24	48	7	3F8
5	22	8	2F8
8	32	8	2F8
8	32	8	2F8
8	32	8	2F8
8	32	8	2F8
31	240	1F	F88
3	248	3F	F88



TOUR NOIRE



Dec.

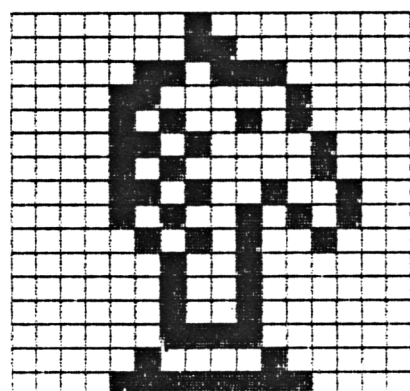
Hex.

0	0	0	0
54	216	36	DD8
54	216	36	DD8
63	248	3F	F88
63	248	3F	F88
63	248	3F	F88
16	16	10	1F8
31	240	1F	1F8
8	32	8	2F8
15	224	F	F88
15	224	F	F88
15	224	F	F88
15	224	F	F88
15	224	F	F88
16	16	10	1F8
63	248	3F	F88





CAVALIER BLANC

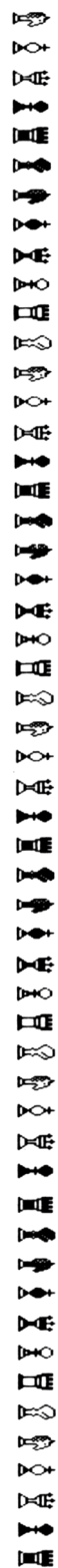


Dec.

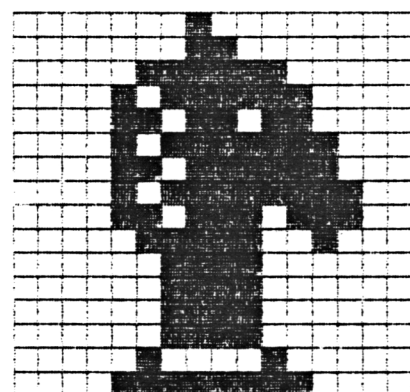
Hex.

1	0
1	12
6	22
1	16
1	80
1	8
1	36
1	84
5	72
5	64
2	64
2	64
2	192
3	132
4	240
15	

1	0
1	08
6	FE
1	15
1	80
1	8
1	24
5	54
5	48
2	40
2	40
2	40
3	4C
4	00
F	00



CAVALIER NOIR



Dec.

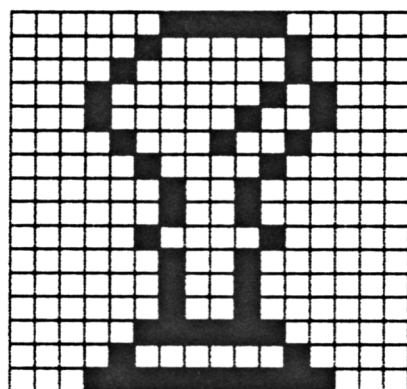
Hex.

1	0
1	12
7	22
1	24
1	40
1	176
1	248
1	248
1	252
1	200
1	200
7	200
3	192
3	180
3	192
3	180
4	32
15	240

1	0
1	08
7	FE
1	17
1	BD
1	BD
1	BD
1	BD
7	DC
3	08
3	00
3	00
3	00
3	00
4	00
F	00



FOU BLANC



Dec.

Hex.

3
4
8
1
1
8
4
2
2
2
2
2
7
8
3

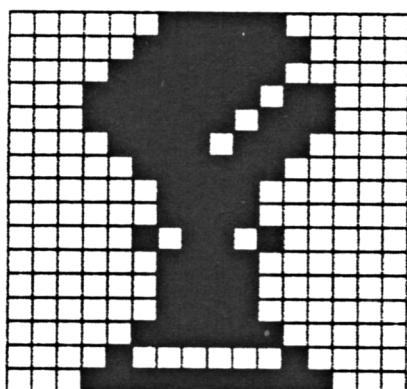
6
6

2
2
1
4
7
3
6
6
3
6
6
6
2
2
2
2
4

4
6
6
2
4
2
4
2
4
4
2
4
2
6
8

[illegible]

FOU NOIR

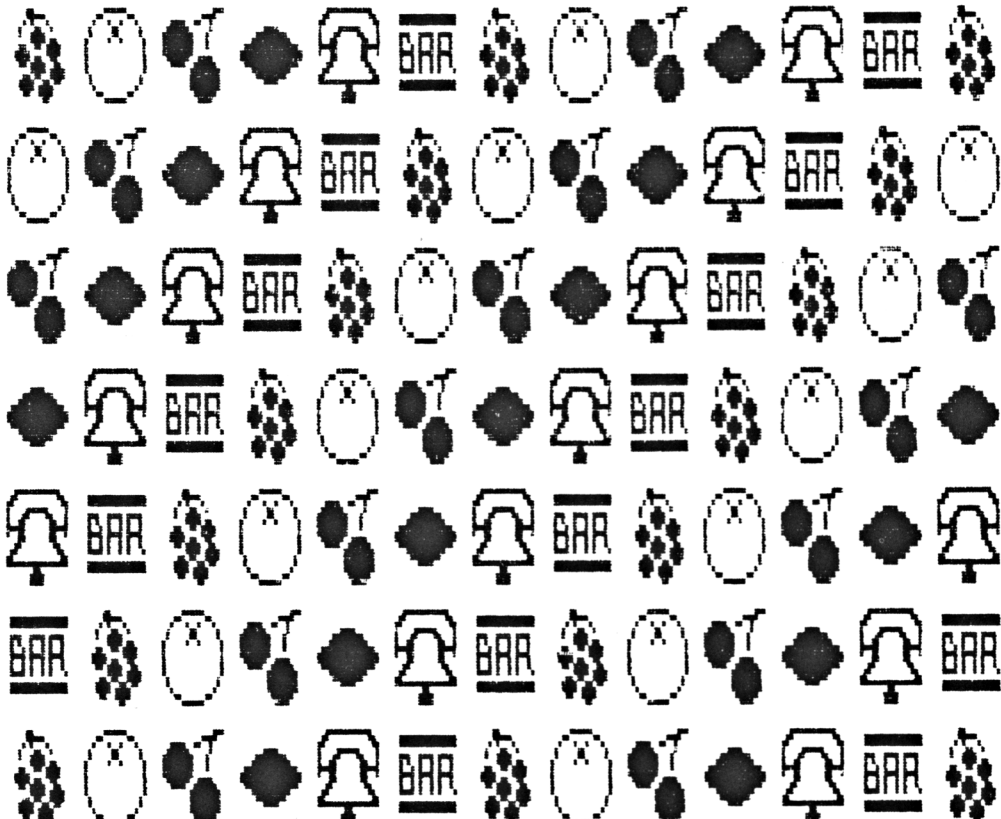
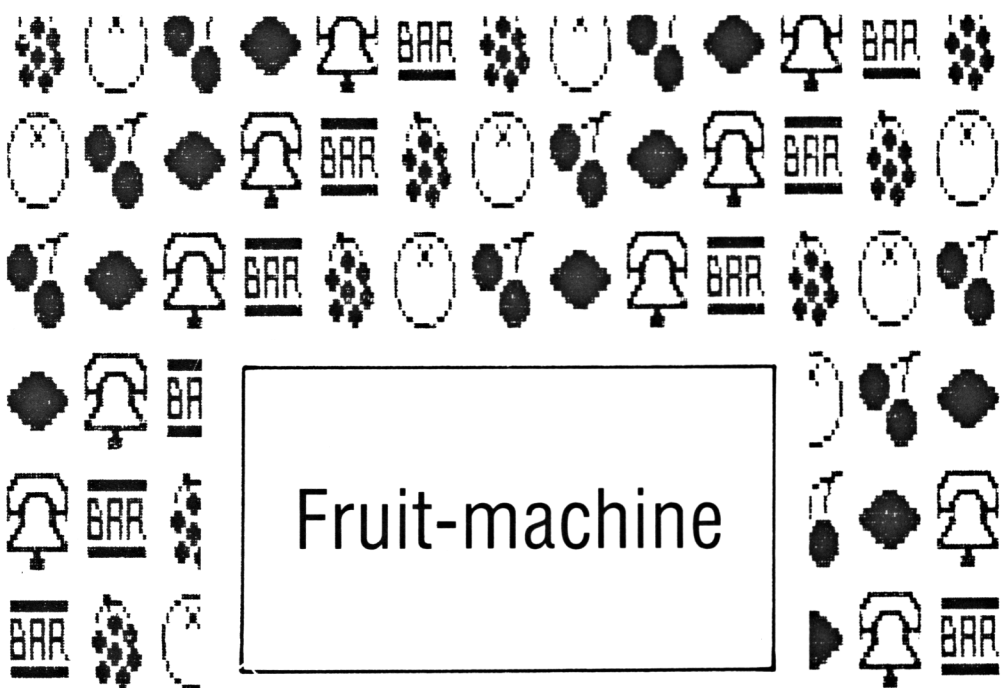


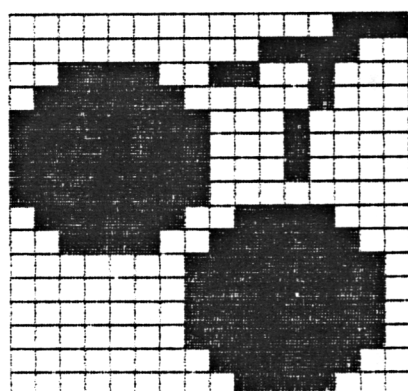
Dec.

Hex.

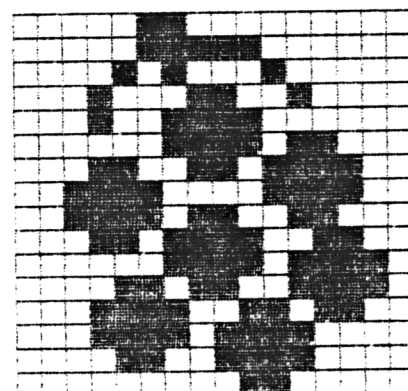
3 3
5 5
1 1

[illegible]



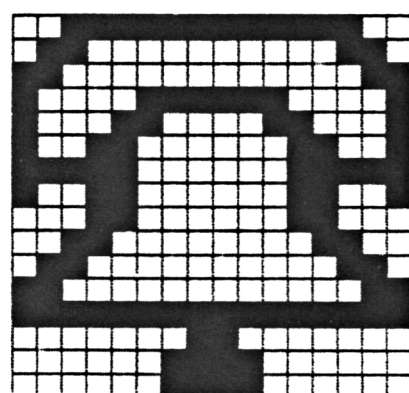


Dec.	Hex.
0	7
0	60
60	2000
12	8
556	16
22	16
55	16
55	16
55	16
55	16
55	16
22	0
55	1200
12	252
60	254
1	254
1	254
1	254
0	254
0	252
0	120



Dec.	Hex.
6	0
7	192
10	32
17	144
19	1192
3	2168
25	188
60	600
61	152
7	204
23	222
13	158
33	108
33	240
12	96
0	0





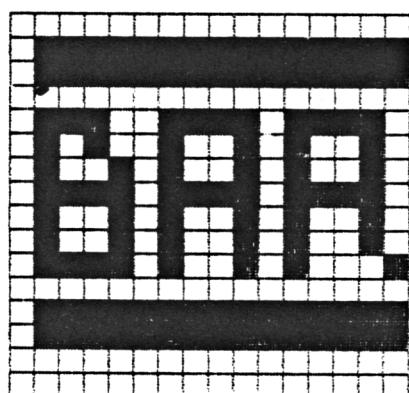
Dec.

Hex.

63
06
06
11
11
11
14
15
24
25
25
24
19
22
55
33

252
06
03
22
25
49
25
33
22
24
12
06
25
11
19
19

3F
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00



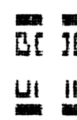
Dec.

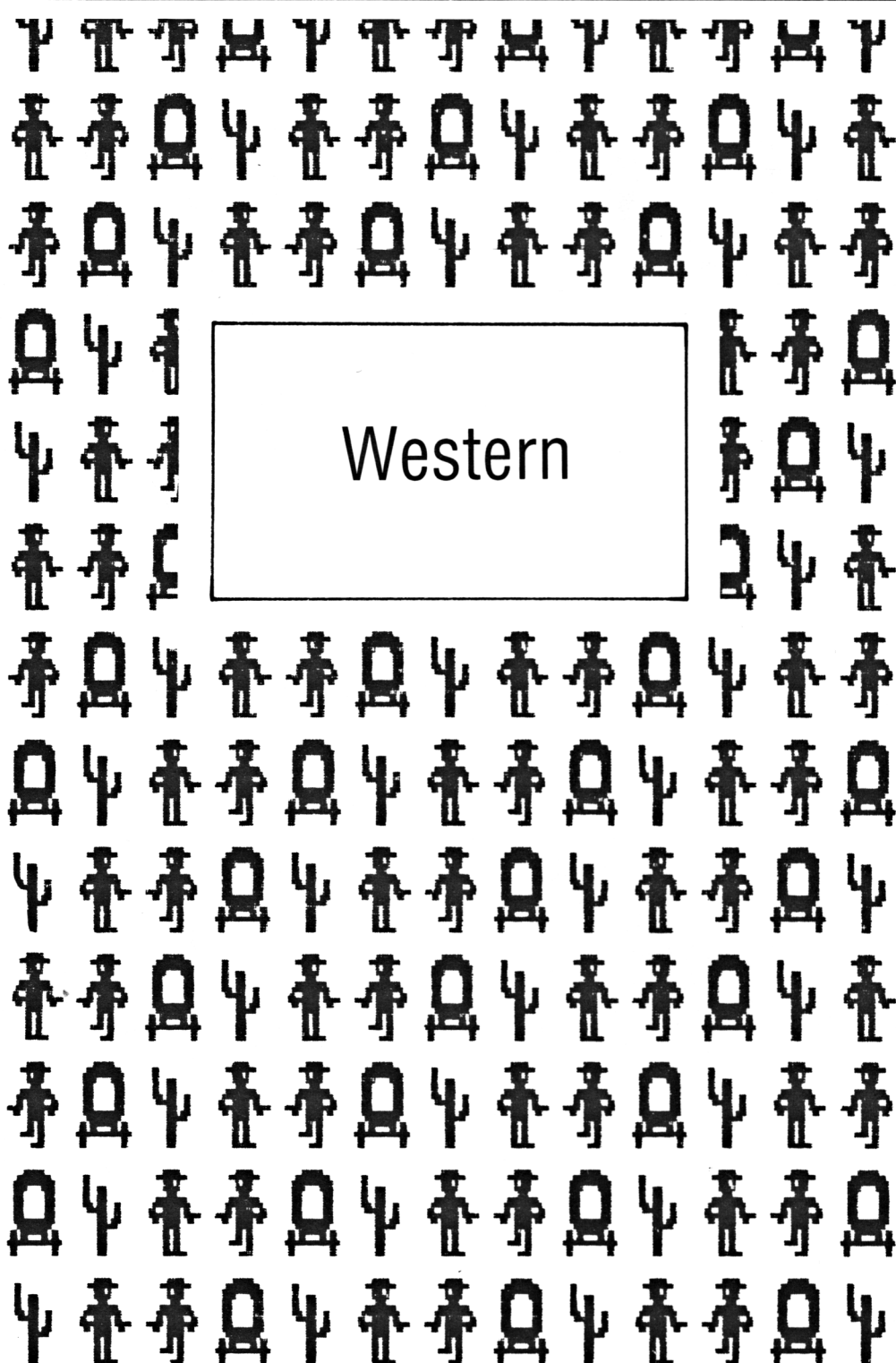
Hex.

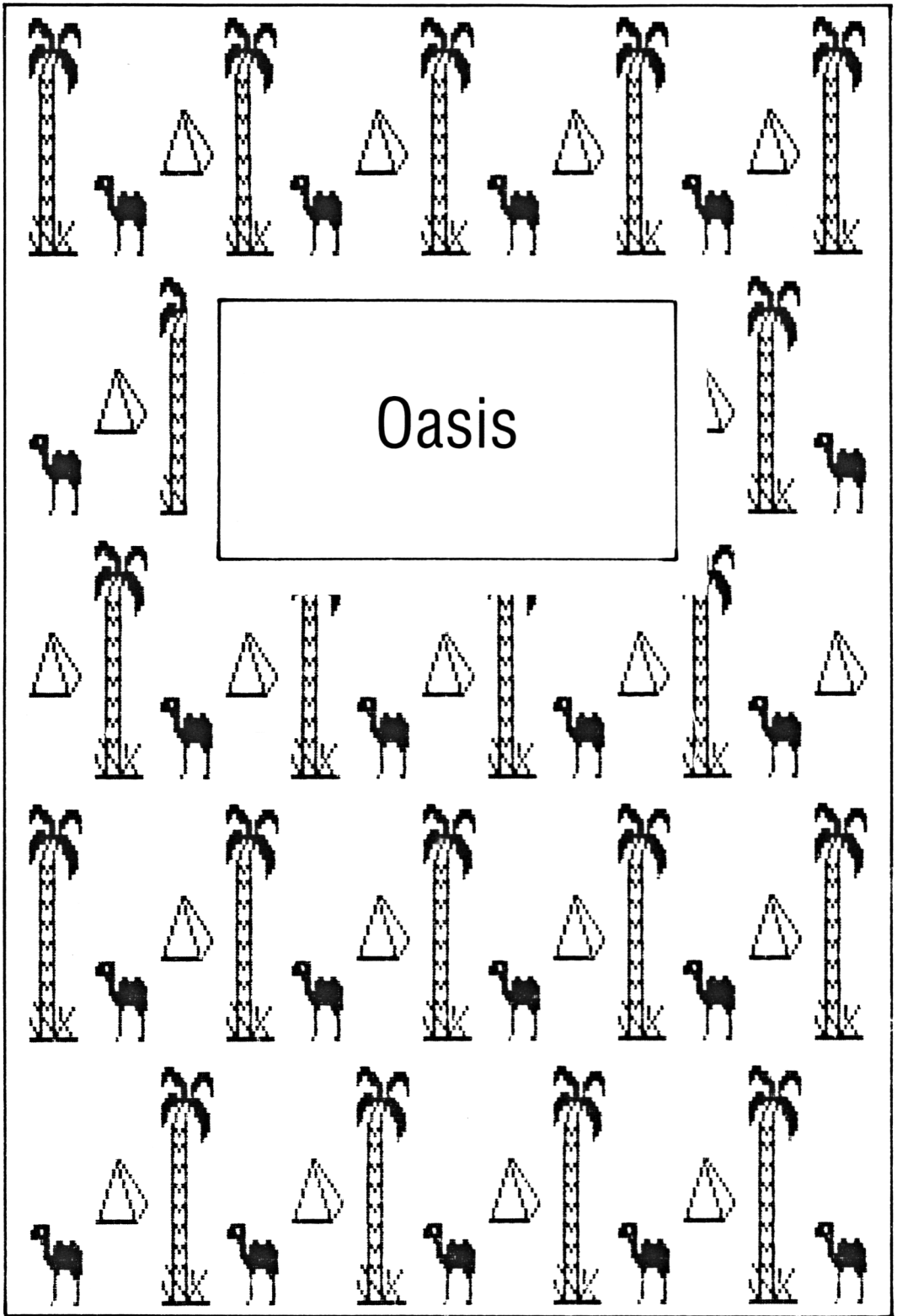
0
127
127
0
115
82
74
123
74
74
122
0
127
127
0
0

0
255
255
0
222
82
82
222
82
82
81
0
255
255
0
0

0
7F
7F
0
73
52
4A
7B
4A
4A
7A
0
7F
7F
0
0



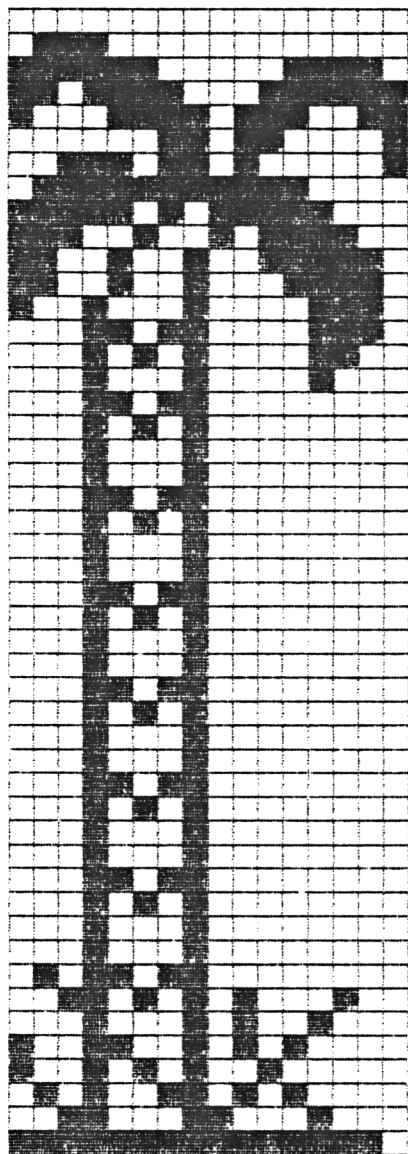
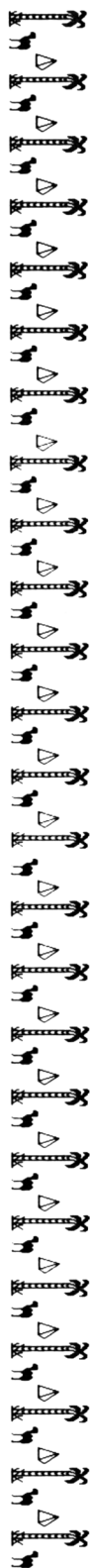




Oasis



COCOTIER

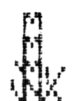
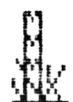


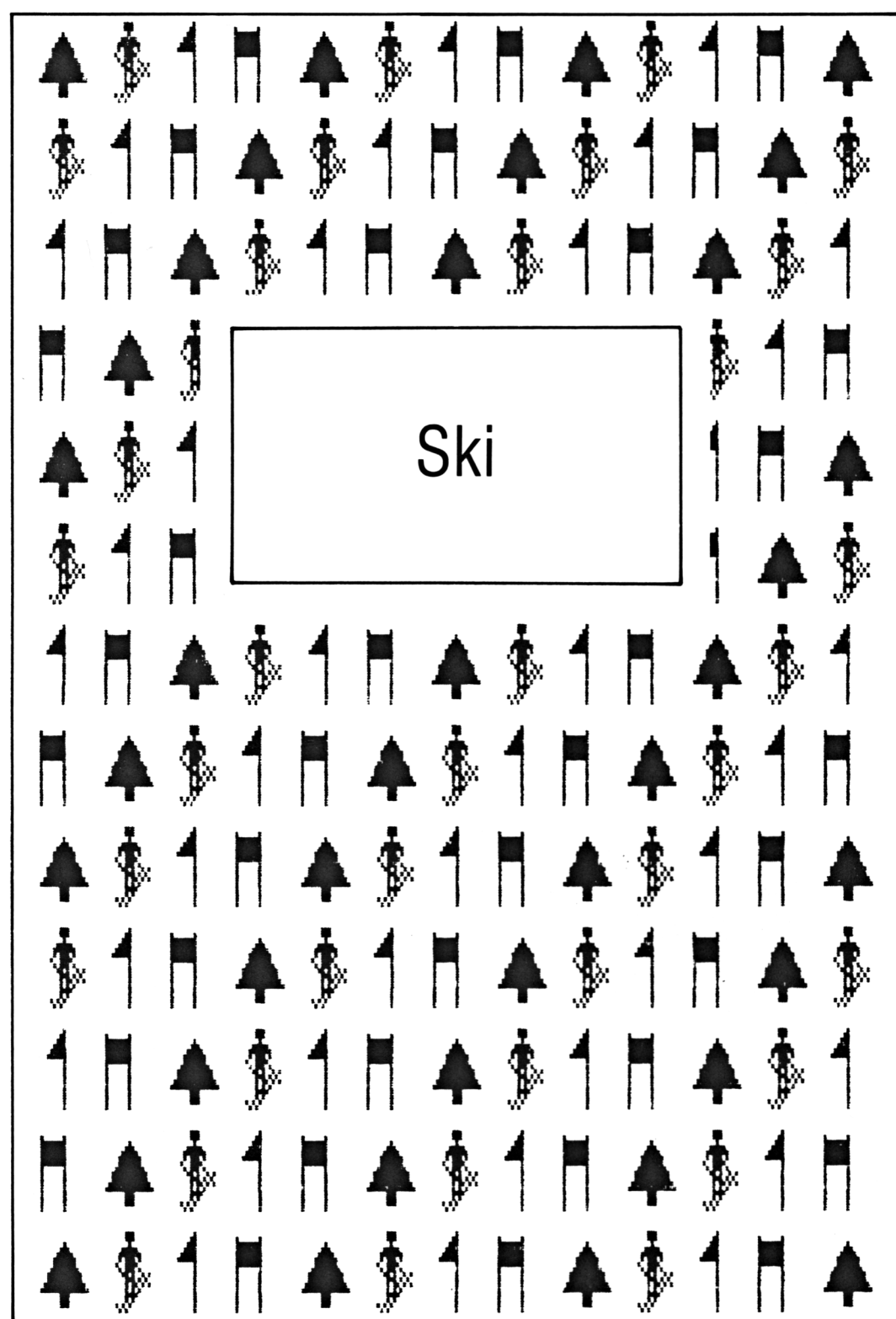
Dec.

Hex.

0 0 0 0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
6 6 6 6
7 7 7 7
8 8 8 8
9 9 9 9
A A A A
B B B B
C C C C
D D D D
E E E E
F F F F

0 0 0 0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
6 6 6 6
7 7 7 7
8 8 8 8
9 9 9 9
A A A A
B B B B
C C C C
D D D D
E E E E
F F F F







Guerre des étoiles



STAR WAR 3



Hex.

[illegible]

STAR WAR 4



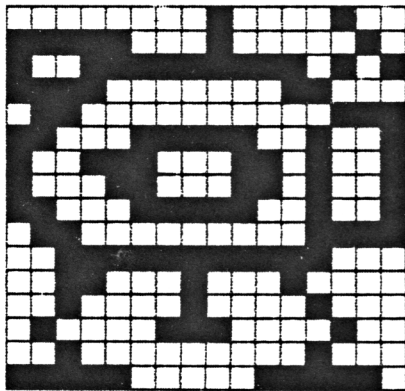
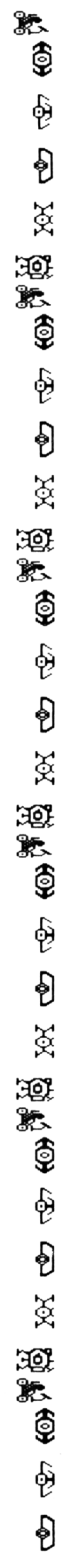
Hex.

42C
322
481
899
881
432
14C
81F
170
870
484
1F

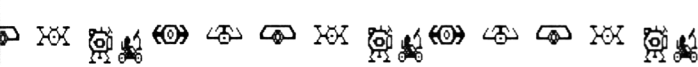
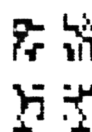




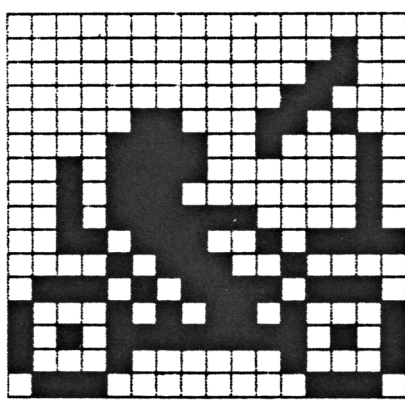
LEM



Dec.	Hex.
0	04
248	08
159	0C
240	0E
96	18
100	1C
156	20
140	24
199	28
96	2C
63	30
40	34
33	38
67	3C
32	40
248	44

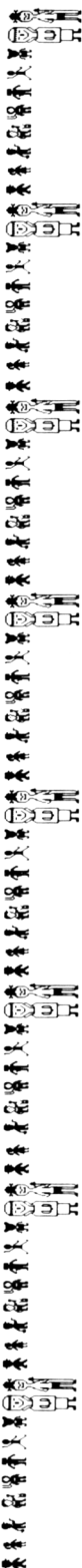


JEEP LUNAIRE

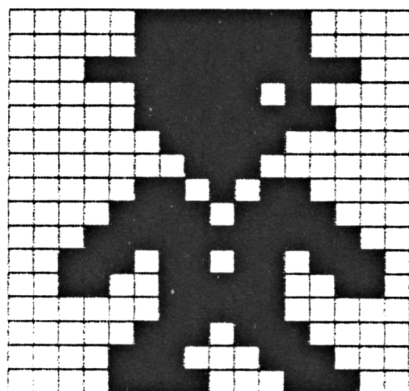
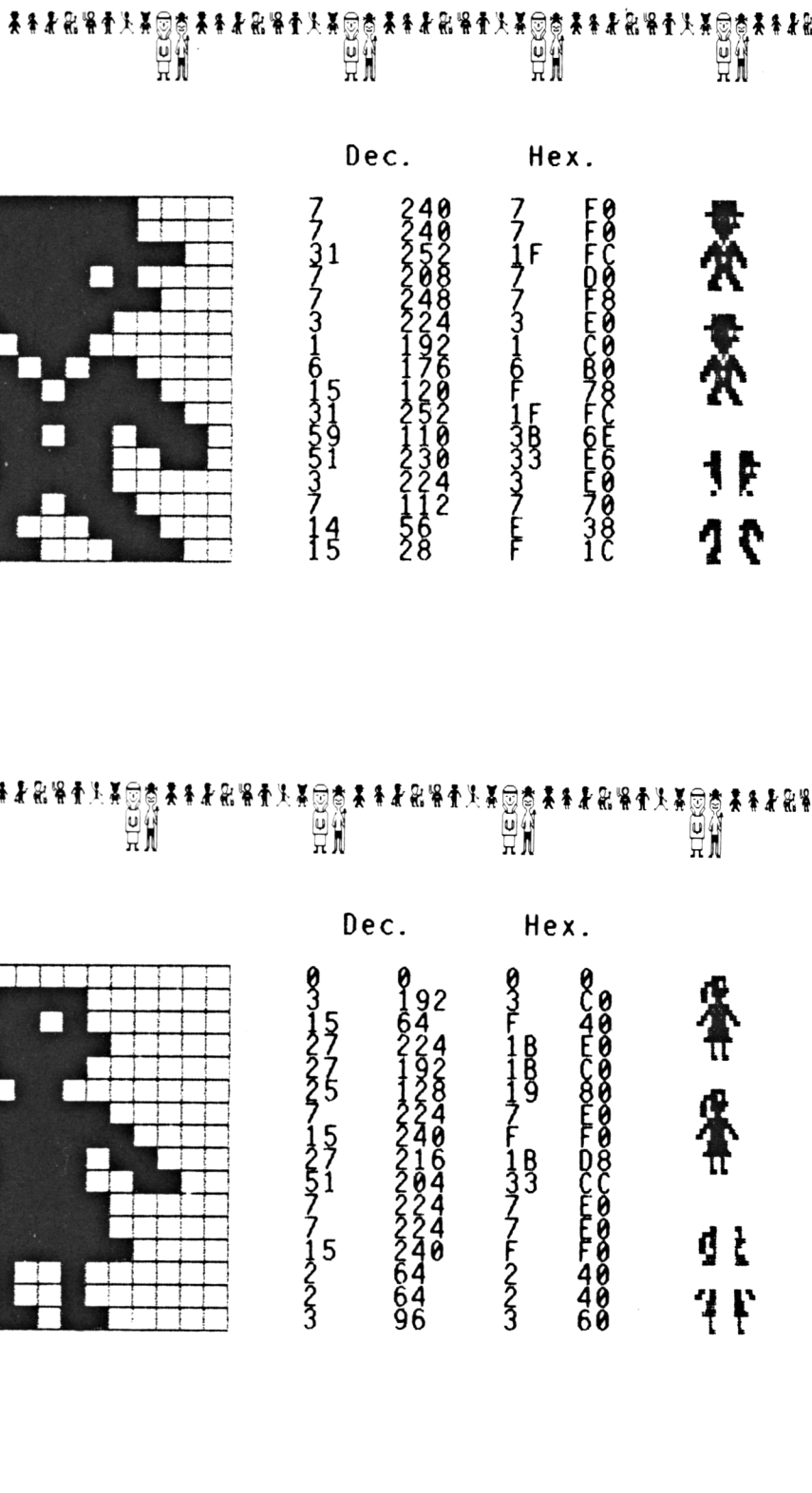


Dec.	Hex.
0	04
00	08
00	0C
00	0E
6	18
15	1C
47	20
46	24
47	28
55	2C
11	30
117	34
138	38
175	3C
136	40
112	44





HOMME



Dec.

Hex.

7
7
31
7
7
3
1
6
15
31
59
51
37
14
15

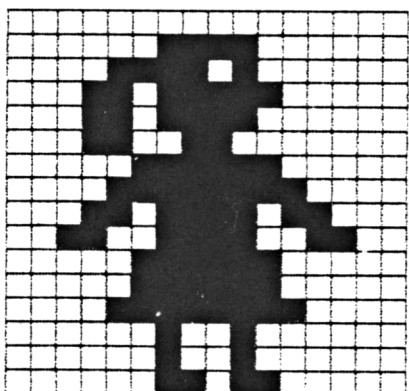
240
240
252
208
248
224
219
176
120
252
110
230
224
112
56
28

7
7
1F
7
7
3
1
6
F
1FB3
333
37
F

F0
F0
FFC
DF0
E00
CC0
B78
FC
6E
EE
70
38
1C



FEMME



Dec.

Hex.

0
3
15
27
27
25
7
15
27
51
7
7
15
22
3

0
192
64
224
228
224
240
216
220
224
224
240
64
64
96

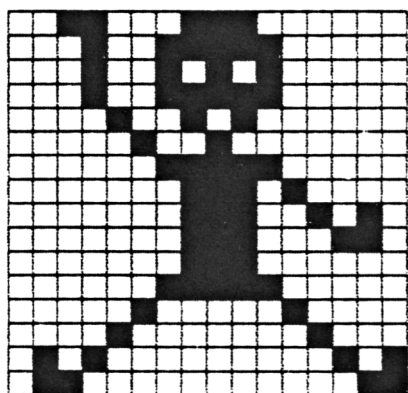
0
3
FB
1B
1B
7
F
1B3
37
7
F
22
3

0
C40
E00
CC0
000
D8
CC0
000
40
40
60





ENFANT



Dec.

Hex.

49
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108
1108

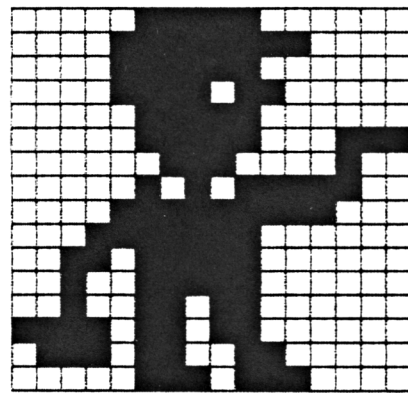
192
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224

31
113
113
113
113
113
113
113
113
113
113
113
113
113
113
113
113
113
113
113

C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0



OUVRIER



Dec.

Hex.

7
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115

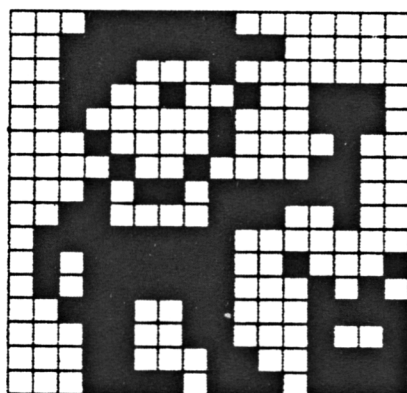
192
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224
224

7
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115
115

C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0
C0



PEINTRE



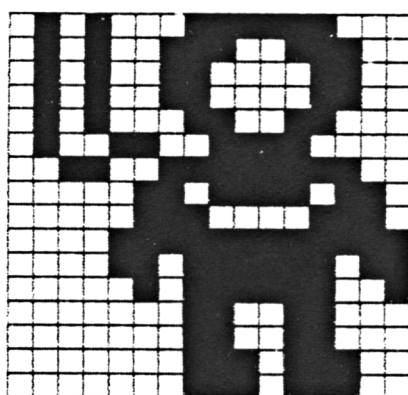
Dec.

Hex.

31	128	1F	80
63	224	3F	00
50	128	33	08
32	78	22	00
16	142	20	08
9	132	10	44
22	12	9	0C
48	252	16	FC
12	228	30	44
95	128	77	00
55	145	5F	80
37	138	5F	91
25	143	39	AD
34	137	10	09
30	207	18	8C
	239	1E	FF



SCAPHANDRIER



Dec.

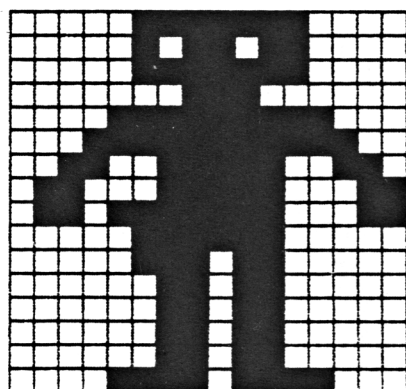
Hex.

81	248	51	F8
83	156	53	9C
83	12	53	0C
83	12	53	0C
16	152	53	00
76	240	4C	00
51	252	33	FC
67	246	36	66
14	155	7F	FF
15	251	D5	00
15	240	51	00
11	152	11	00
11	232	11	00
11	230	11	DC
11	230	11	DC

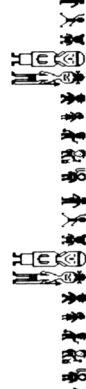




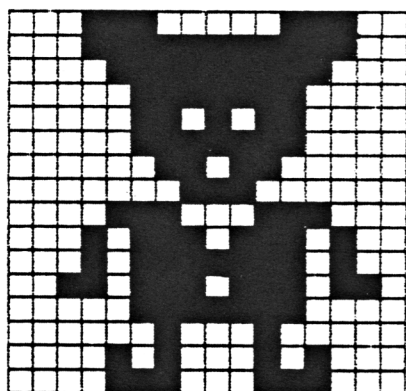
ANDROÏDE



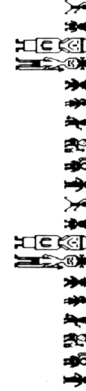
Dec.		Hex.	
7	240	7	F00
5	176	5	B00
7	240	7	B00
1	192	1	000
15	248	15	000
3	252	3	000
1	260	1	000
9	330	9	000
11	227	11	000
7	224	7	000
3	96	3	000
3	96	3	000
3	96	3	000
15	96	15	000
	120		708

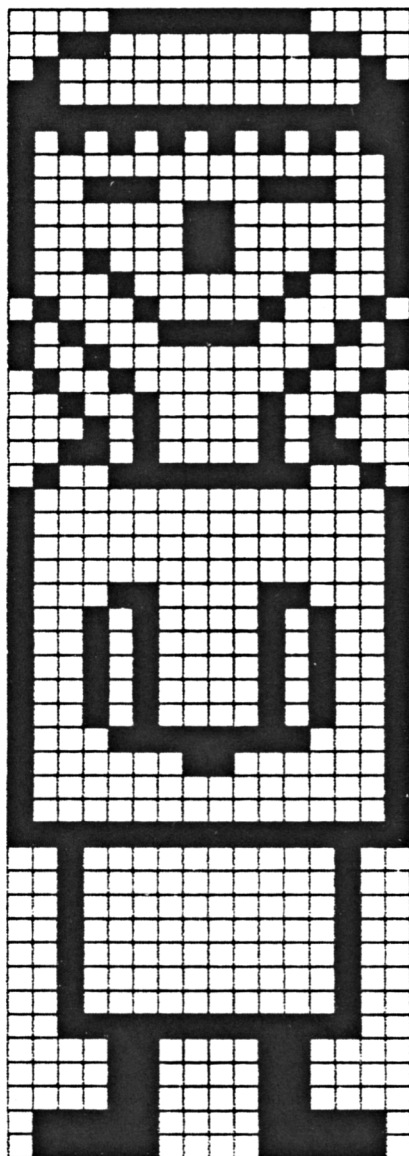
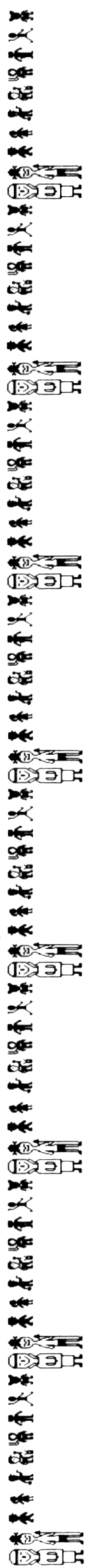


OURSON



Dec.		Hex.	
28	28	1C	1C
31	252	1F	00
15	248	F	00
7	240	7	00
6	176	6	00
7	240	7	00
3	96	3	00
1	192	1	00
14	56	E	00
23	116	17	48
33	244	27	74
55	118	37	76
7	240	7	00
2	32	2	00
10	40	A	08
14	56	E	38

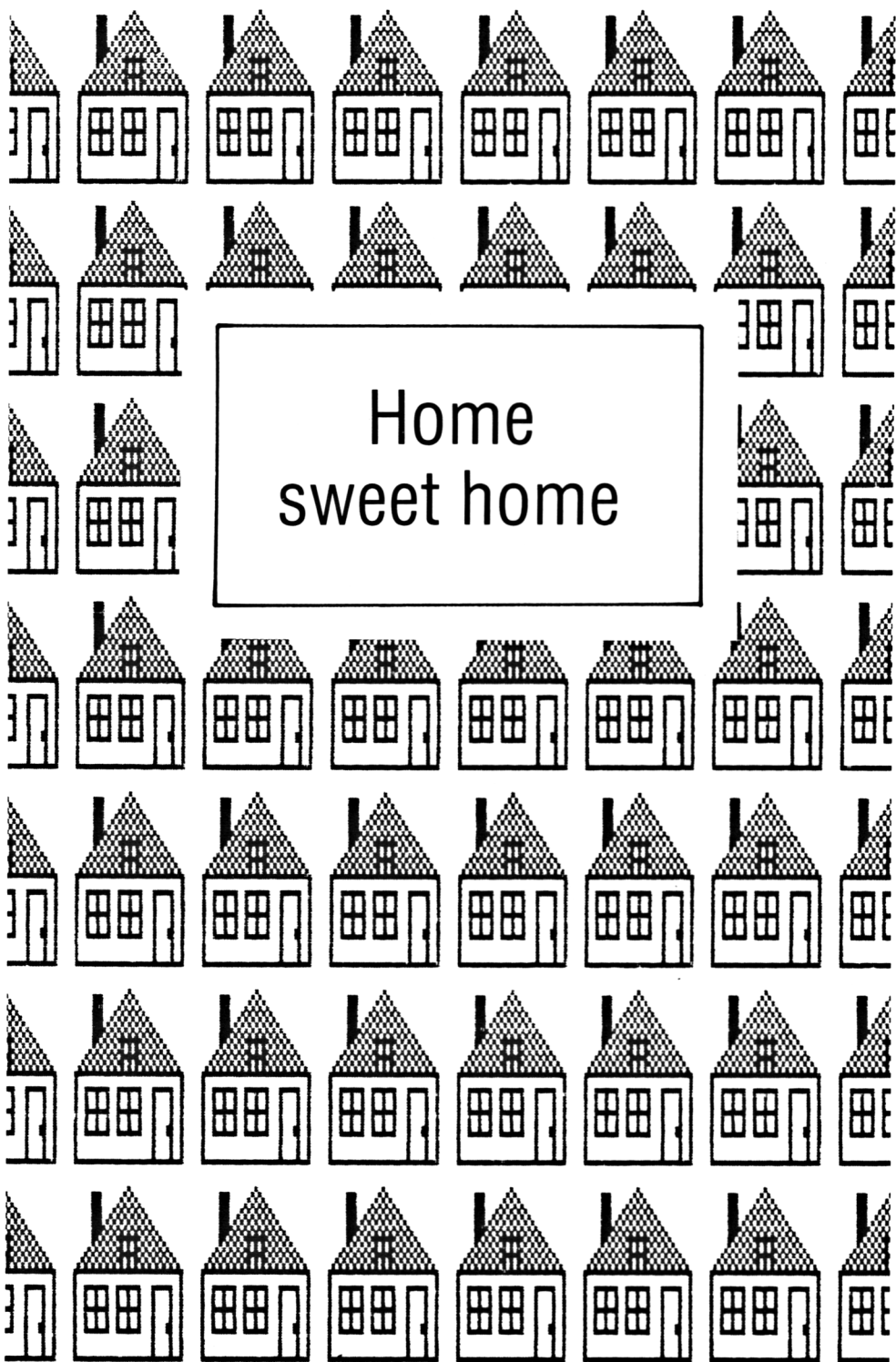




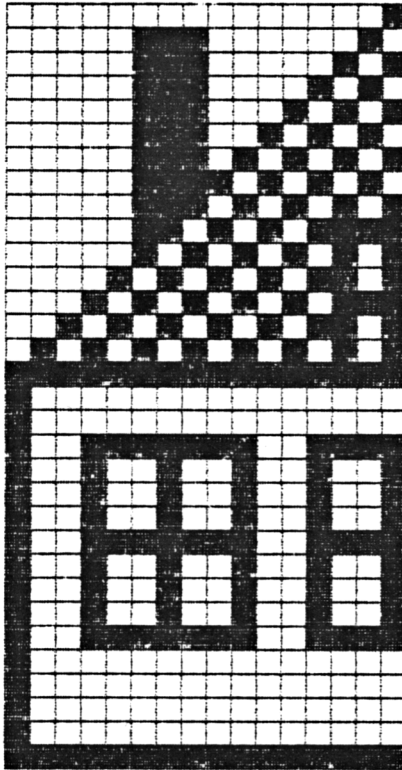
Hex.

15	240	F	F
148	12	30	0
146	23	40	0
145	2	FF	AA
144	55	80	80
143	17	81	11
142	57	88	88
141	11	88	88
140	22	88	88
139	57	88	88
138	11	88	88
137	22	88	88
136	57	88	88
135	11	88	88
134	22	88	88
133	57	88	88
132	11	88	88
131	22	88	88
130	57	88	88
129	11	88	88
128	22	88	88
127	57	88	88
126	11	88	88
125	22	88	88
124	57	88	88
123	11	88	88
122	22	88	88
121	57	88	88
120	11	88	88
119	22	88	88
118	57	88	88
117	11	88	88
116	22	88	88
115	57	88	88
114	11	88	88
113	22	88	88
112	57	88	88
111	11	88	88
110	22	88	88
109	57	88	88
108	11	88	88
107	22	88	88
106	57	88	88
105	11	88	88
104	22	88	88
103	57	88	88
102	11	88	88
101	22	88	88
100	57	88	88
99	11	88	88
98	22	88	88
97	57	88	88
96	11	88	88
95	22	88	88
94	57	88	88
93	11	88	88
92	22	88	88
91	57	88	88
90	11	88	88
89	22	88	88
88	57	88	88
87	11	88	88
86	22	88	88
85	57	88	88
84	11	88	88
83	22	88	88
82	57	88	88
81	11	88	88
80	22	88	88
79	57	88	88
78	11	88	88
77	22	88	88
76	57	88	88
75	11	88	88
74	22	88	88
73	57	88	88
72	11	88	88
71	22	88	88
70	57	88	88
69	11	88	88
68	22	88	88
67	57	88	88
66	11	88	88
65	22	88	88
64	57	88	88
63	11	88	88
62	22	88	88
61	57	88	88
60	11	88	88
59	22	88	88
58	57	88	88
57	11	88	88
56	22	88	88
55	57	88	88
54	11	88	88
53	22	88	88
52	57	88	88
51	11	88	88
50	22	88	88
49	57	88	88
48	11	88	88
47	22	88	88
46	57	88	88
45	11	88	88
44	22	88	88
43	57	88	88
42	11	88	88
41	22	88	88
40	57	88	88
39	11	88	88
38	22	88	88
37	57	88	88
36	11	88	88
35	22	88	88
34	57	88	88
33	11	88	88
32	22	88	88
31	57	88	88
30	11	88	88
29	22	88	88
28	57	88	88
27	11	88	88
26	22	88	





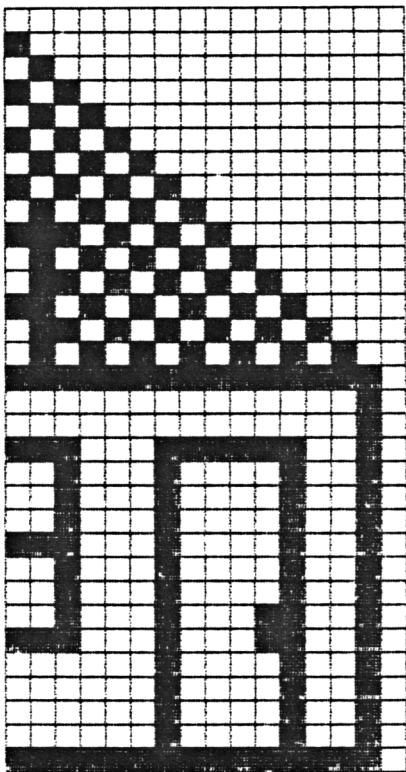
Home
sweet home



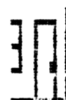
Hex.

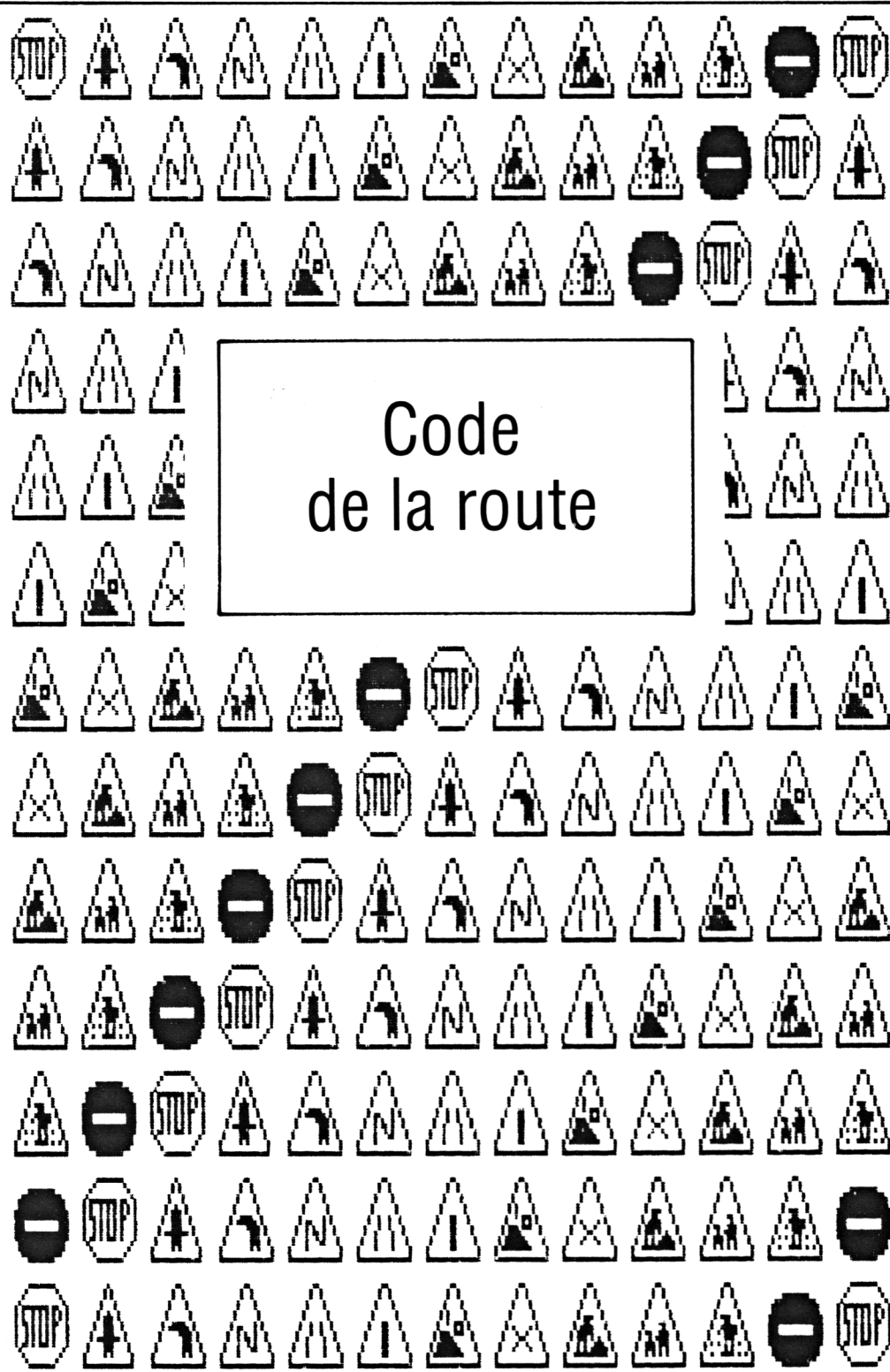
0	1	0	1
7	2	7	2
7	5	7	5
7	10	7	A
7	21	7	15
7	42	7	2A
7	85	7	55
7	170	7	AA
7	345	7	55
7	690	7	AF
6	1381	6	55
5	2762	5	AD
10	5524	A	57
21	11048	15	AD
42	22096	2A	55
85	44192	55	FF
170	88384	FF	00
345	176768	00	00
690	353536	00	00
1381	707072	00	00
2762	1414144	00	00
5524	2828288	00	00
11048	5656576	00	00
22096	11313152	00	00
44192	22626304	00	00
88384	45252608	00	00
176768	90505216	00	00
353536	181010432	00	00
707072	362020864	00	00
1414144	724041728	00	00
2828288	1448083456	00	00
5656576	2896166912	00	00
11313152	5792333824	00	00
22626304	11584667648	00	00
45252608	23169335296	00	00
90505216	46338670592	00	00
181010432	92677341184	00	00
362020864	185354682368	00	00
724041728	370709364736	00	00
1448083456	741418729472	00	00
2896166912	1482837458944	00	00
5792333824	2965674917888	00	00
11584667648	5931349835776	00	00
23169335296	11862699671552	00	00
46338670592	23725399343104	00	00
92677341184	47450798686208	00	00
185354682368	94901597372416	00	00
370709364736	189803194744832	00	00
741418729472	379606389489664	00	00
1482837458944	759212778979328	00	00
2965674917888	1518425577958656	00	00
5931349835776	3036851155917312	00	00
11862699671552	6073702311834624	00	00
23725399343104	12147404623669248	00	00
47450798686208	24294809247338496	00	00
94901597372416	48589618494676992	00	00
189803194744832	97179236989353984	00	00
379606389489664	194358473978707968	00	00
759212778979328	388716947957415936	00	00
1518425577958656	777433895914831872	00	00
3036851155917312	1554867791829663744	00	00
6073702311834624	3109735583659327488	00	00
12147404623669248	6219471167318654976	00	00
24294809247338496	12438942344637309952	00	00
48589618494676992	24877884689274619904	00	00
97179236989353984	49755769378549239808	00	00
194358473978707968	99511538757098479616	00	00
388716947957415936	199023077540196959232	00	00
777433895914831872	398046155080393918464	00	00
1554867791829663744	796092310160787836928	00	00
3109735583659327488	1592184620321575673856	00	00
6219471167318654976	3184369240643151347712	00	00
12438942344637309952	6368738481286302695424	00	00
24877884689274619904	12737476962572605390848	00	00
49755769378549239808	25474953925145210781696	00	00
99511538757098479616	50949907850290421563392	00	00
199023077540196959232	101899815700580843126784	00	





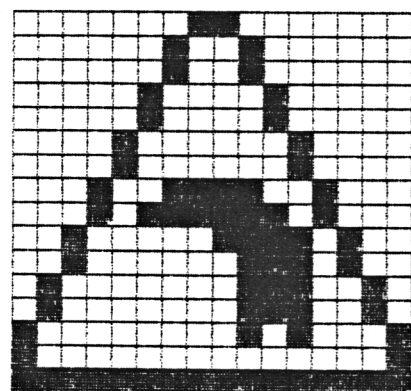
Dec.	Hex.
0	0
128	80
164	40
160	A0
80	50
168	A8
84	54
170	AA
85	55
234	EA
85	55
1106	6A
213	05
106	6A
85	55
255	FF
0	0
27	E3
34	22
34	22
34	22
34	22
26	EE
34	22
34	22
26	EE
2	2
2	2
2	2
2	2
255	FF







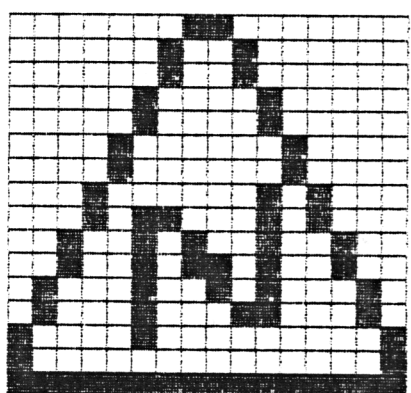
VIRAGE GAUCHE



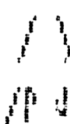
Dec.	Hex.
1	128
2	64
2	64
4	32
4	32
8	16
8	16
16	8
2	2
2	2
3	3
3	3
6	6
6	6
12	12
255	255



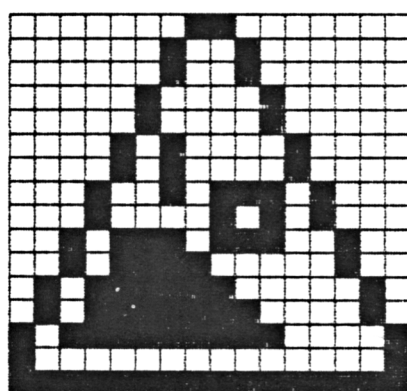
VIRAGE DROITE-GAUCHE



Dec.	Hex.
1	128
2	64
2	64
4	32
4	32
8	16
8	16
16	8
2	2
2	2
3	3
3	3
6	6
6	6
12	12
255	255



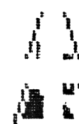
DANGER D'ÉBOULEMENT



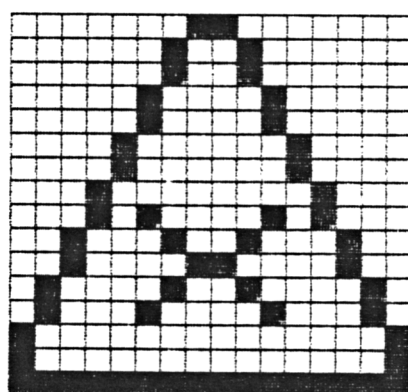
Dec.

Hex.

1	128	1	80
2	64	2	40
3	64	3	40
4	32	4	20
5	32	5	20
6	16	6	10
7	16	7	10
8	8	8	8
9	8	9	8
10	4	10	4
11	4	11	4
12	2	12	2
13	2	13	2
14	1	14	1
15	1	15	1
16	1	16	1



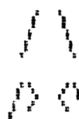
INTERSECTION



Dec.

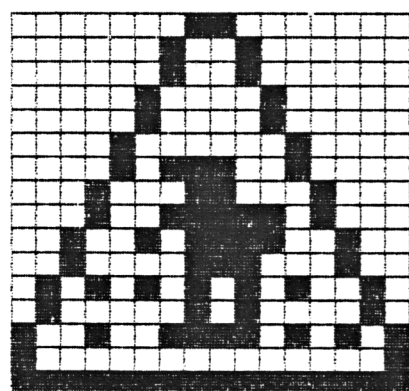
Hex.

1	128	1	80
2	64	2	40
3	64	3	40
4	32	4	20
5	32	5	20
6	16	6	10
7	16	7	10
8	8	8	8
9	8	9	8
10	4	10	4
11	4	11	4
12	2	12	2
13	2	13	2
14	1	14	1
15	1	15	1
16	1	16	1

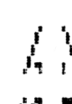




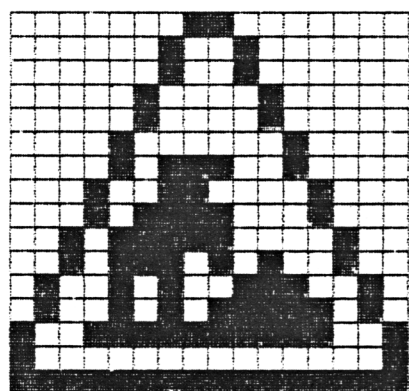
PASSAGE CLOUTÉ



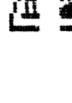
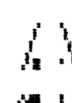
Dec.	Hex.
1	128
2	64
3	64
4	32
4	32
8	16
11	144
17	136
19	232
37	228
33	196
85	82
65	66
147	213
128	1
255	255



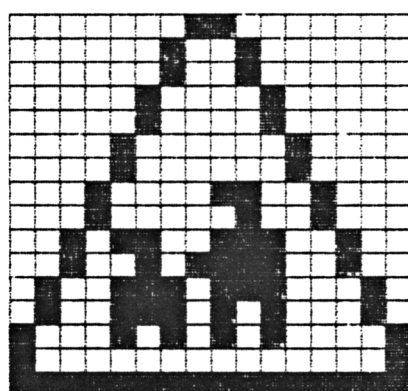
TRAVAUX



Dec.	Hex.
1	128
2	64
3	64
4	32
4	32
8	16
11	144
19	8
23	136
47	132
46	164
74	114
74	250
159	249
128	1
255	255



ÉCOLE



Dec.

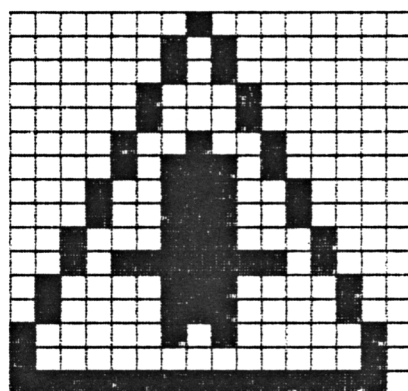
Hex.

1	128
2	64
2	64
4	32
4	32
8	16
8	16
16	8
16	8
44	22
37	22
78	22
78	16
138	16
125	1
255	255

1	80
2	40
2	40
4	20
4	20
8	10
8	10
16	5
16	5
44	4
37	4
78	4
78	2
138	2
125	1
255	255



PRIORITÉ A L'INTERSECTION



Dec.

Hex.

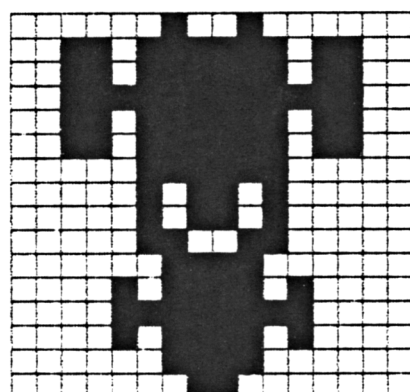
1	0
2	2
2	2
4	6
4	6
8	3
8	3
16	1
16	1
44	4
37	4
78	4
78	2
138	2
125	1
255	254

1	0
2	0
2	0
4	0
4	0
8	0
8	0
16	0
16	0
44	0
37	0
78	0
78	0
138	0
125	0
255	255



Traffic

BOLIDE A



Dec.

Hex.

1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

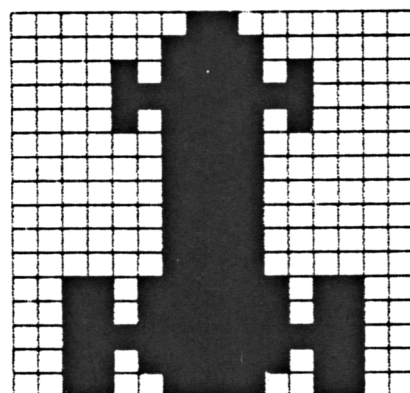
1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1



BOLIDE B



Dec.

Hex.

1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

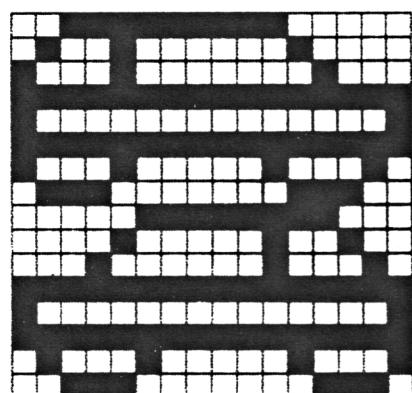
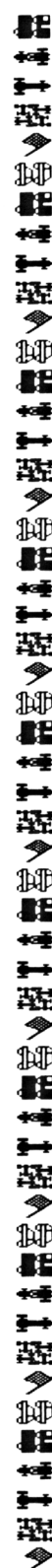
1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1





VOITURE



Dec.

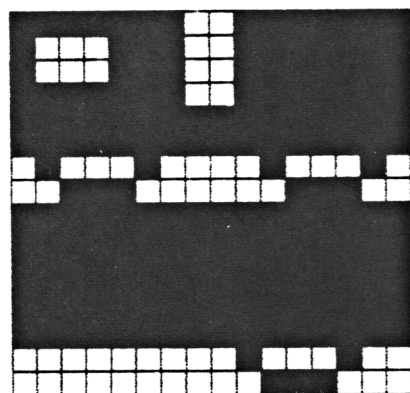
Hex.

63	224
72	16
136	8
255	255
1255	1
255	255
1330	34
112	28
7	48
8	36
16	34
255	255
1255	1
255	255
68	17
56	14

3F	E0
40	10
80	8F
80	1F
80	2C
70	F8
7	24
8	22
10	2F
80	1F
44	11
38	E



CAMION



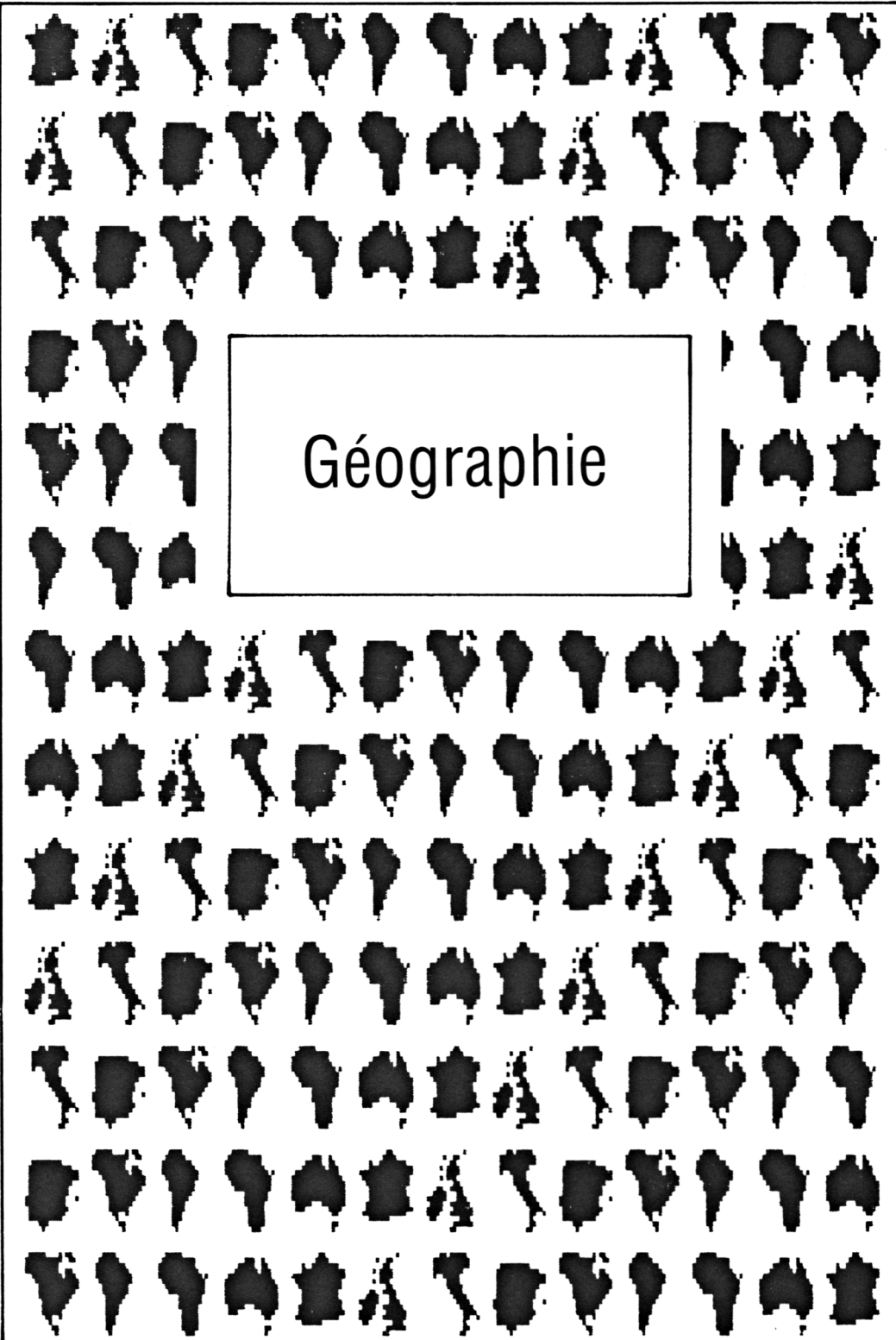
Dec.

Hex.

254	127
142	127
142	127
254	127
255	255
255	255
68	34
56	28
255	255
255	255
255	255
255	255
255	255
255	255
255	255
0	68
0	56

FE	7F
8FE	7F
8FE	7F
FE	7F
FE	7F
FE	7F
44	22
38	1C
FE	FE
FE	FE
FE	FE
FE	FE
FE	FE
FE	FE
0	44
0	38





Géographie

FRANCE



Hex.

[illegible]

ROYAUME-UNI A ZEPY 9024 ZEPY 9024 ZEPY 9024

Hex.

1-800-777-7430
9-000000



THE



Hex.

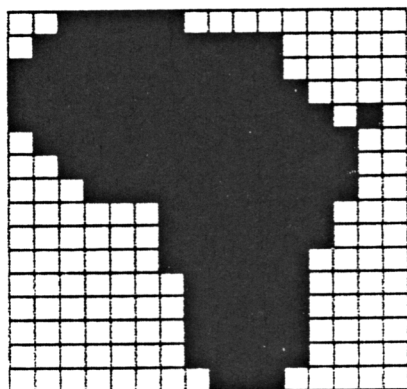
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840.



Hex.



AFRIQUE 9024 7879 9024 7879 9024 7879 9024 7879 9024



Dec.

Hex.

627
125
225
225
225
123
633
333
333
111
111
0

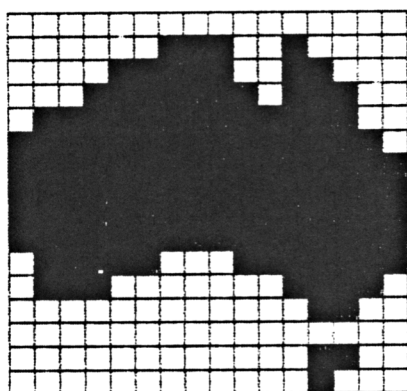
[illegible]

**E
F
F
F
F
F
F
F
F**

SECRET



AUSTRALIE



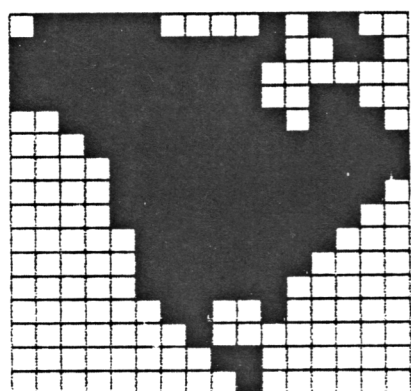
Dec.

Hex.

0313122222110000

[illegible][illegible][illegible]

THE UNIVERSITY OF CHICAGO PRESS



Dec.

Hex.

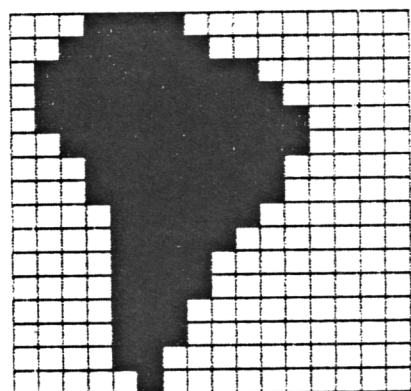
123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100

[illegible]

7C
7E
7F
7G
7H
7I
7J
7K
7L
7M
7N
7O
7P
7Q
7R
7S
7T
7U
7V
7W
7X
7Y
7Z

[illegible]

SECRET



Dec.

Hex.

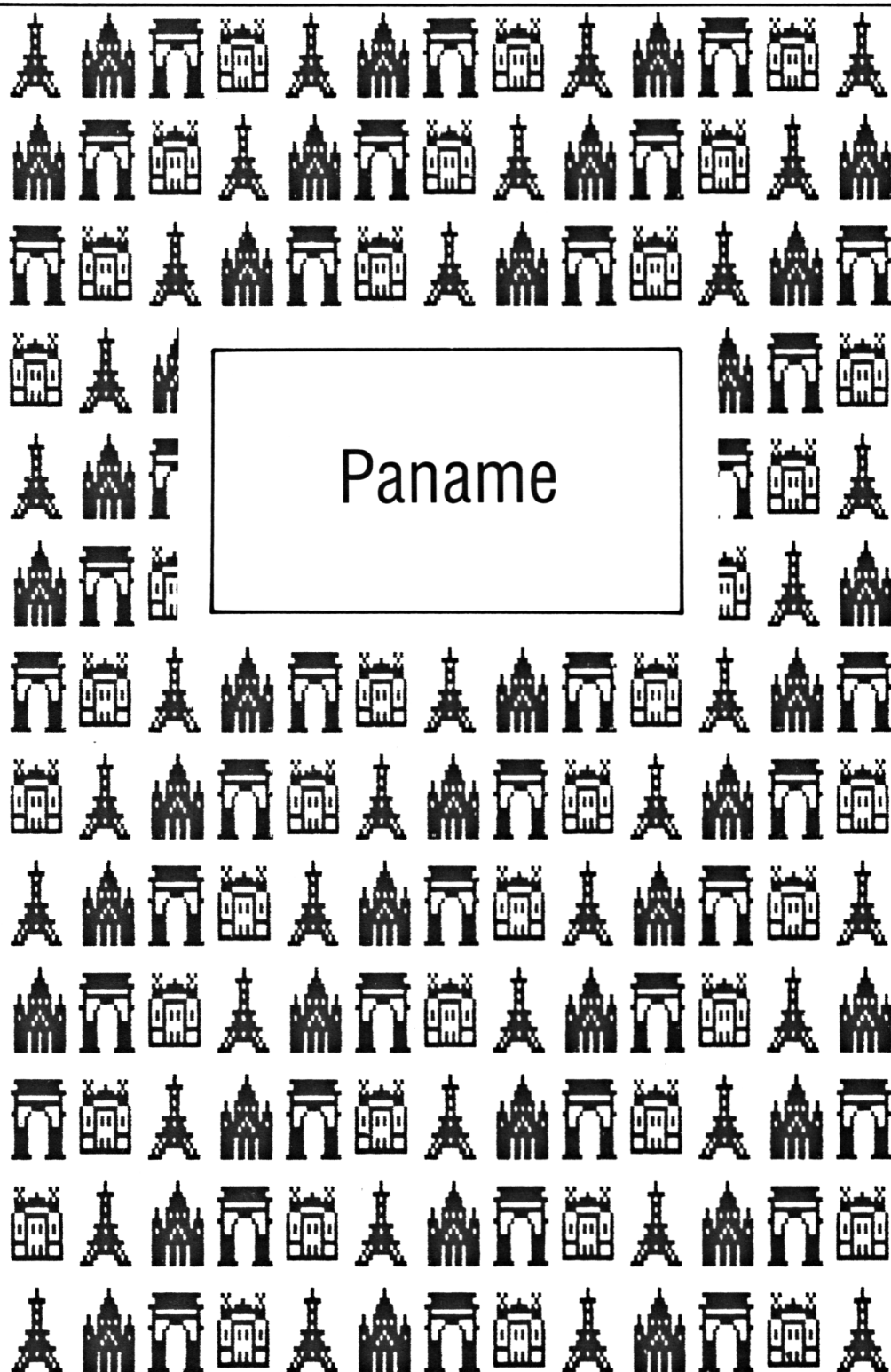
36111163311111114

0012222211000000

13777311FFFFC4

00000000
00000000
00000000

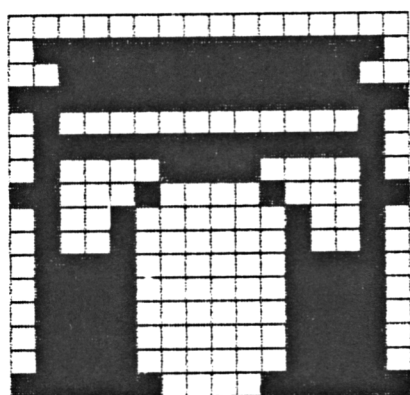




Paname



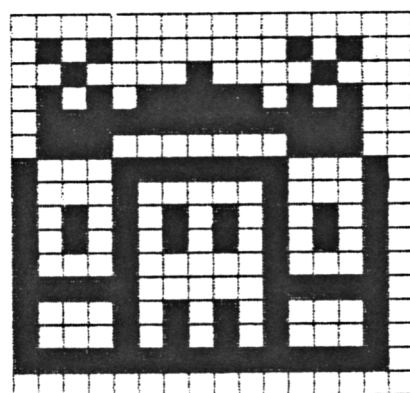
ARC DE TRIOMPHE



| Dec. | | Hex. | |
|------|------|------|----|
| 0 | 127 | 0 | 7F |
| 1 | 63 | 2 | 3F |
| 2 | 55 | 3 | FF |
| 3 | 64 | 4 | 0 |
| 4 | 127 | 5 | 7F |
| 5 | 67 | 6 | 43 |
| 6 | 196 | 7 | C4 |
| 7 | 72 | 8 | 48 |
| 8 | 72 | 9 | 48 |
| 9 | 1200 | A | 78 |
| A | 1200 | B | 78 |
| B | 1200 | C | 78 |
| C | 1200 | D | 78 |
| D | 1200 | E | 78 |
| E | 1200 | F | 78 |
| F | 252 | | FC |



OPÉRA



| Dec. | | Hex. | |
|------|------|------|----|
| 0 | 800 | 0 | 50 |
| 1 | 33 | 1 | 21 |
| 2 | 87 | 2 | 57 |
| 3 | 127 | 3 | 7F |
| 4 | 112 | 4 | 70 |
| 5 | 143 | 5 | 8F |
| 6 | 136 | 6 | 88 |
| 7 | 1700 | 7 | AA |
| 8 | 1700 | 8 | AA |
| 9 | 136 | 9 | 88 |
| A | 248 | A | F8 |
| B | 138 | B | 8A |
| C | 138 | C | 8A |
| D | 255 | D | FF |
| E | | E | 0 |



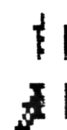


Hex.

[illegible]

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

00000000000000000000000000000000



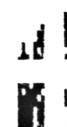
Hex.

[illegible]

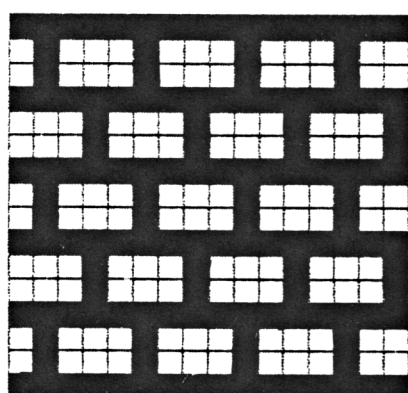
112222821122111
229224247993511
88242799351111

[illegible]

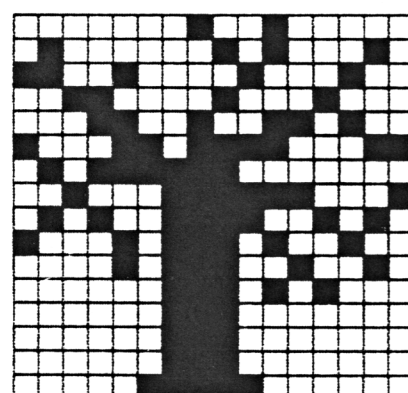
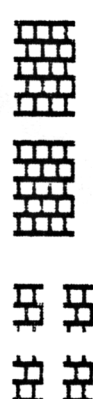
00000000000000000000000000000000







| Dec. | | Hex. | |
|------|-----|------|----|
| 255 | 255 | FF | FF |
| 68 | 68 | 44 | 44 |
| 68 | 68 | 44 | 44 |
| 255 | 255 | FF | FF |
| 17 | 17 | 11 | 11 |
| 17 | 17 | 11 | 11 |
| 255 | 255 | FF | FF |
| 68 | 68 | 44 | 44 |
| 68 | 68 | 44 | 44 |
| 255 | 255 | FF | FF |
| 17 | 17 | 11 | 11 |
| 17 | 17 | 11 | 11 |
| 255 | 255 | FF | FF |
| 68 | 68 | 44 | 44 |
| 68 | 68 | 44 | 44 |
| 255 | 255 | FF | FF |

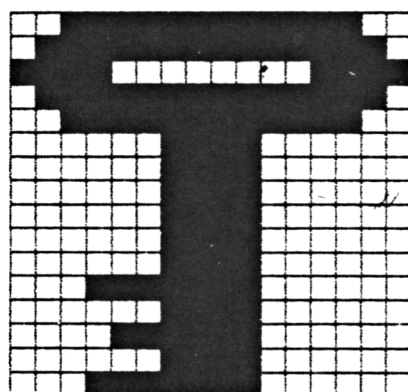


| Dec. | | Hex. | |
|------|-----|------|----|
| 1 | 32 | 1 | 20 |
| 64 | 162 | 40 | A2 |
| 200 | 68 | C0 | 44 |
| 48 | 136 | 30 | 88 |
| 80 | 52 | 50 | 34 |
| 234 | 227 | 0D | 33 |
| 10 | 128 | 0F | 80 |
| 55 | 241 | 33 | F1 |
| 33 | 116 | 33 | CA |
| 11 | 204 | 33 | A4 |
| 9 | 146 | 33 | 92 |
| 3 | 168 | 33 | A8 |
| 3 | 228 | 33 | 80 |
| 3 | 228 | 33 | 80 |
| 3 | 228 | 33 | 80 |
| 7 | 192 | 33 | C0 |





CLEF



Dec.

Hex.

63
12
24
11
12
33
33
33
33
33
33
11
15
31

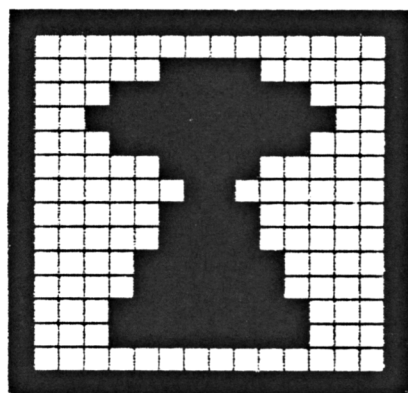
252
254
154
254
252
102
102
102
102
102
102
102
102
102

3F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F

FF
FF
FF
FF
FF
FF
FF
FF
FF
FF
FF
FF
FF
FF
FF



SERRURE



Dec.

Hex.

255
128
131
143
159
143
129
131
131
131
131
131
131
131
131

255
193
241
249
241
193
193
193
193
193
193
193
193
193

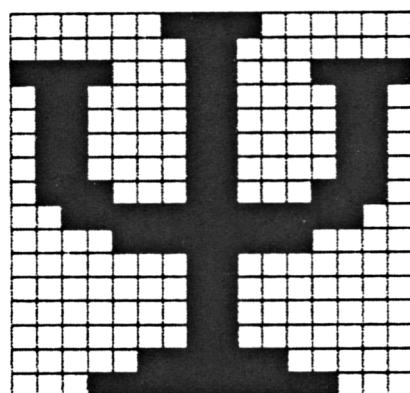
FF
88
88
88
88
88
88
88
88
88
88
88
88
88
88

FF
11
11
11
11
11
11
11
11
11
11
11
11
11
11





CHANDELIER



Dec.

Hex.

3
1
241
97
97
97
97
113
63
15
1
1
1
1
7
31

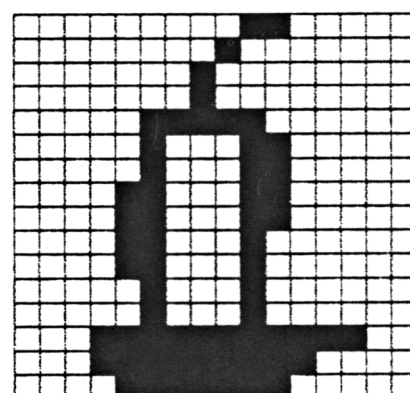
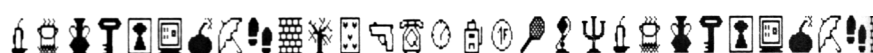
192
128
143
134
134
134
142
252
240
128
128
128
224
248

3
1
F1
61
61
61
61
7F
F
1
1
1
1
7
1F

C0
00
00
00
00
00
00
00
00
00
00
00
00
00
00



BOUGIE



Dec.

Hex.

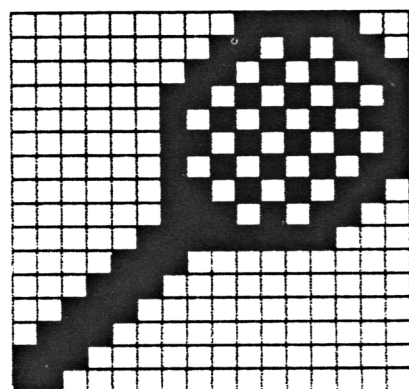
0
0
1
1
7
4
4
12
12
12
12
4
4
31
31
15

96
128
0
192
96
96
96
96
64
64
64
64
4
4
252
240
224

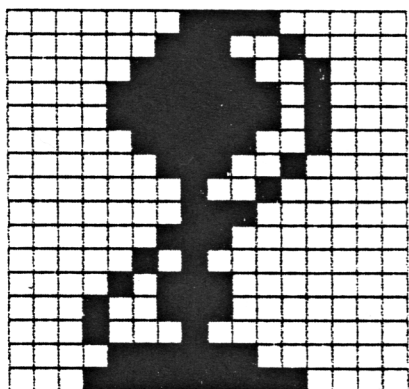
0
0
1
1
7
4
4
C
C
C
C
4
4
1F
1F
F

60
80
00
C0
60
60
60
60
40
40
40
40
40
40
FC
FE
E0



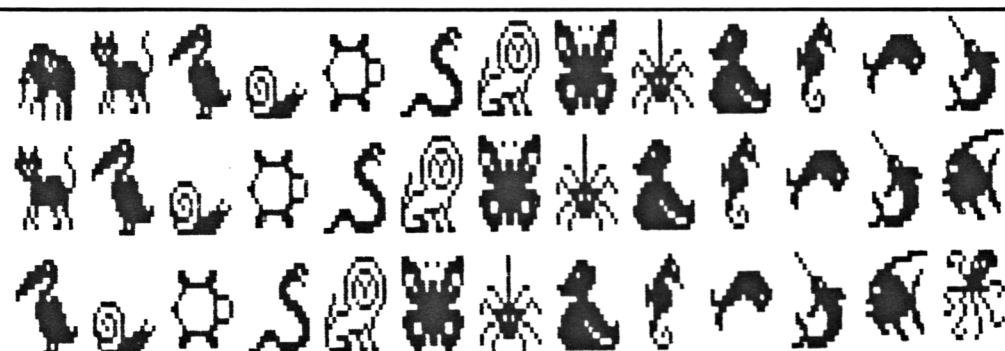


| Dec. | Hex. |
|------|------|
| 0 | 7C |
| 1 | 00 |
| 2 | 00 |
| 3 | 00 |
| 4 | 00 |
| 5 | 00 |
| 6 | 00 |
| 7 | 00 |
| 8 | 00 |
| 9 | 00 |
| 10 | 00 |
| 11 | 00 |
| 12 | 00 |
| 13 | 00 |
| 14 | 00 |
| 15 | 00 |
| 16 | 00 |
| 17 | 00 |
| 18 | 00 |
| 19 | 00 |
| 20 | 00 |



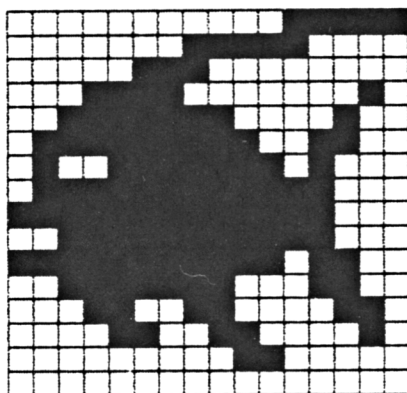
| Dec. | Hex. |
|------|------|
| 1 | F0 |
| 2 | 00 |
| 3 | 00 |
| 4 | 00 |
| 5 | 00 |
| 6 | 00 |
| 7 | 00 |
| 8 | 00 |
| 9 | 00 |
| 10 | 00 |
| 11 | 00 |
| 12 | 00 |
| 13 | 00 |
| 14 | 00 |
| 15 | 00 |
| 16 | 00 |
| 17 | 00 |
| 18 | 00 |
| 19 | 00 |
| 20 | 00 |





Ménagerie

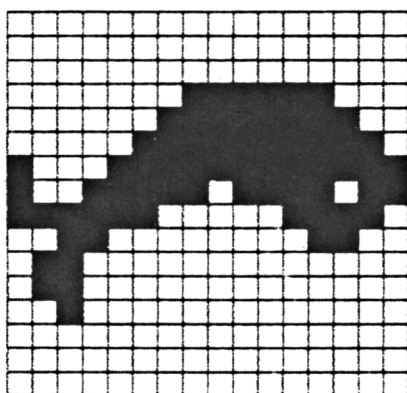




Hex.

[illegible]

```
01713747FFFF9  
F04C8888CC28  
F08888888888
```



Hex.

[illegible]

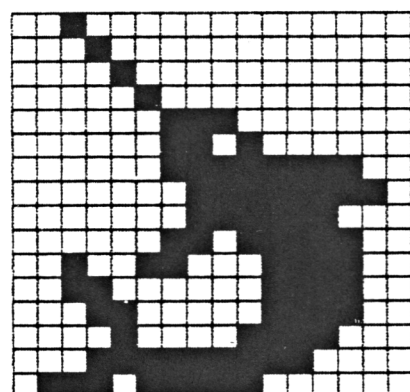
000000
000000

000000
000000





POISSON-SCIE



Dec.

Hex.

32
18
6
4
3
3
3
1
1
3
3
3
4
2
8
3
1
19

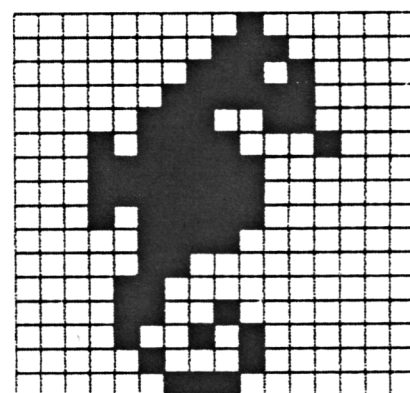
0
0
0
0
12
64
25
25
24
12
16
60
60
60
12
24
192

20
18
0
4
3
3
3
1
3
3
3
6
3
8
8
F
7

0
0
0
0
0
4
F
F
F
F
7
3
3
3
3
F
C



HIPPOCAMPE



Dec.

Hex.

0
0
1
3
7
2
3
3
2
7
6
1
1
2
9
4
3

64
22
20
24
48
11
10
10
10
10
2
0
12
64
128

0
0
1
3
7
1
1
1
1
7
6
C
C
C
4
3

40
F
D
F
3
8
8
C
C
C
8
8
8
4
8
8

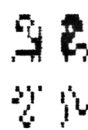


PIEUVRE



Hex.

38
34
69
99
73
77
84
42
28
81
24
18
0



SERPENT

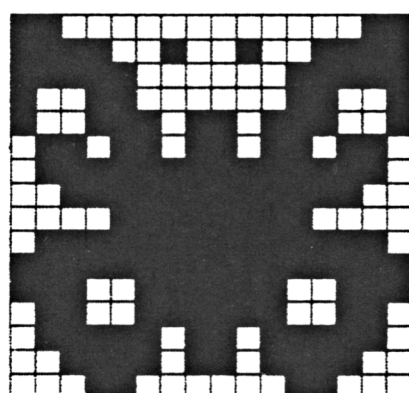


Hex.

[illegible]



PAPILLON



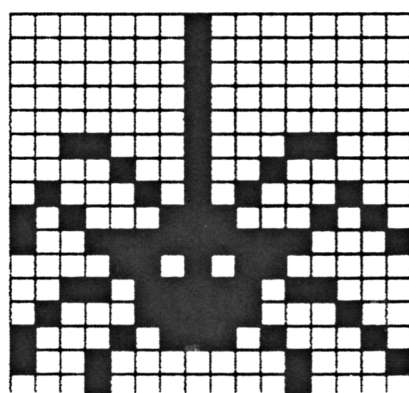
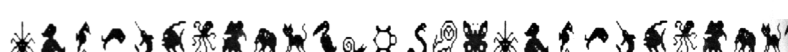
Dec.

Hex.

| | | | |
|-----|----|----|----|
| 192 | 3 | C0 | 3 |
| 242 | 79 | F8 | 4F |
| 248 | 31 | F8 | 4F |
| 152 | 25 | 90 | 80 |
| 157 | 18 | 90 | 80 |
| 109 | 55 | 60 | 00 |
| 127 | 25 | 7F | 00 |
| 63 | 25 | 7F | 00 |
| 15 | 24 | 7F | 00 |
| 127 | 25 | 7F | 00 |
| 255 | 55 | 7F | 00 |
| 233 | 33 | E7 | 7 |
| 121 | 19 | 67 | 6 |
| 61 | 18 | 7D | BC |
| 24 | 24 | 18 | 18 |



ARAIGNÉE



Dec.

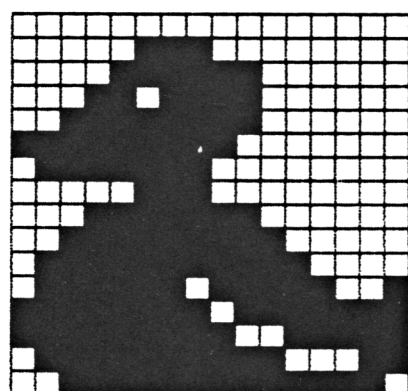
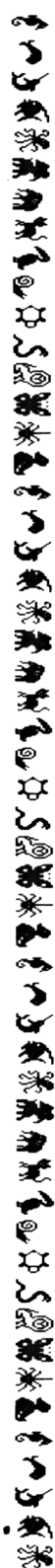
Hex.

| | | | |
|-----|----|----|----|
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 49 | 24 | 31 | 18 |
| 69 | 68 | 45 | 24 |
| 103 | 33 | A3 | 4A |
| 159 | 24 | 9F | 8A |
| 55 | 96 | D9 | F6 |
| 71 | 21 | 37 | 08 |
| 139 | 19 | 47 | C4 |
| 144 | 16 | 8B | A2 |
| 16 | 16 | 90 | 11 |





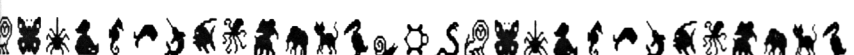
CANARD



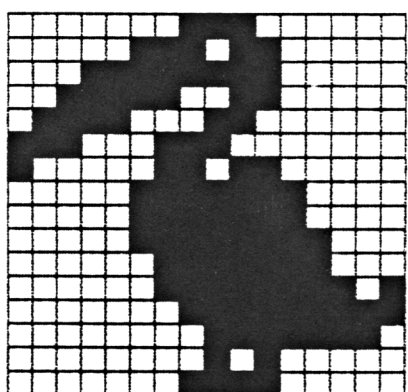
Dec.

Hex.

| | | | |
|----|----|----|---|
| 0 | 0 | 0 | 0 |
| 7 | 1 | 7 | 0 |
| 15 | 9 | F | 0 |
| 27 | 02 | 1B | 0 |
| 63 | 06 | 3B | 0 |
| 23 | 19 | 7F | 0 |
| 55 | 28 | 7F | 0 |
| 27 | 00 | 7F | 0 |
| 17 | 00 | 7F | 0 |
| 31 | 02 | 1F | 0 |
| 63 | 24 | 3F | 0 |
| 23 | 24 | 7F | 0 |
| 12 | 40 | 7F | 0 |
| 27 | 40 | 7F | 0 |
| 55 | 77 | 7F | 0 |
| 23 | 59 | 7F | 0 |
| 33 | 57 | 7F | 0 |
| 63 | 54 | 3F | 0 |



PÉLICAN



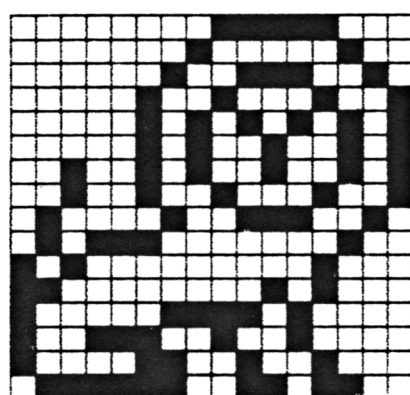
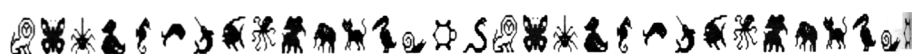
Dec.

Hex.

| | | | |
|----|----|----|----|
| 1 | 19 | 1 | C0 |
| 7 | 02 | 7 | 00 |
| 31 | 04 | 1F | 00 |
| 63 | 06 | 3F | 00 |
| 23 | 06 | 7F | 00 |
| 55 | 08 | 7F | 00 |
| 27 | 08 | 7F | 00 |
| 13 | 08 | 7F | 00 |
| 77 | 08 | 7F | 00 |
| 33 | 08 | 7F | 00 |
| 33 | 08 | 7F | 00 |
| 11 | 08 | 7F | 00 |
| 00 | 08 | 7F | 00 |
| 11 | 08 | 7F | 00 |
| 22 | 08 | 7F | 00 |
| 22 | 08 | 7F | 00 |



LION

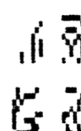


Dec.

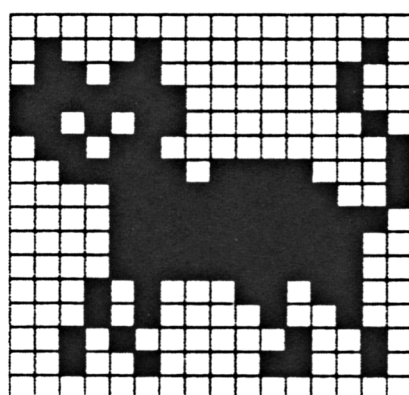
Hex.

0 248
1 411
2 114
3 417
4 857
5 377
6 1137
7 414
8 80
9 40
A 208
B 144
C 72
D 108
E 168

0 F8
1 472
2 800
3 550
4 550
5 550
6 550
7 550
8 550
9 550
A 550
B 550
C 550
D 550
E 550



CHAT



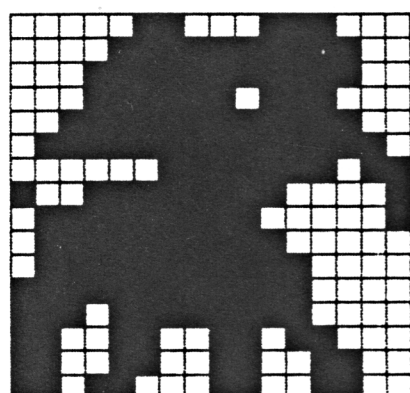
Dec.

Hex.

0 248
1 411
2 114
3 417
4 857
5 377
6 1137
7 414
8 80
9 40
A 208
B 144
C 72
D 108
E 168

0 F8
1 472
2 800
3 550
4 550
5 550
6 550
7 550
8 550
9 550
A 550
B 550
C 550
D 550
E 550





Dec.

6
3
3
6
1
3
1
1
1
2
2
2
2
2

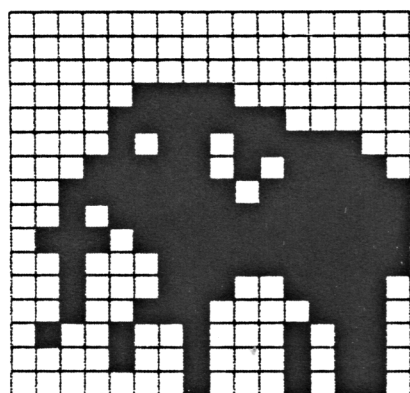
5
1
3
3
2
7
5
2
2
2
5
3
0
0
1

9
2
7
7
7
5
9
4
4
6

[illegible]

Hex.

6F
14373977FFCCD

[illegible]

Dec.

00071134133379
00071134133353
00071134133353⁹

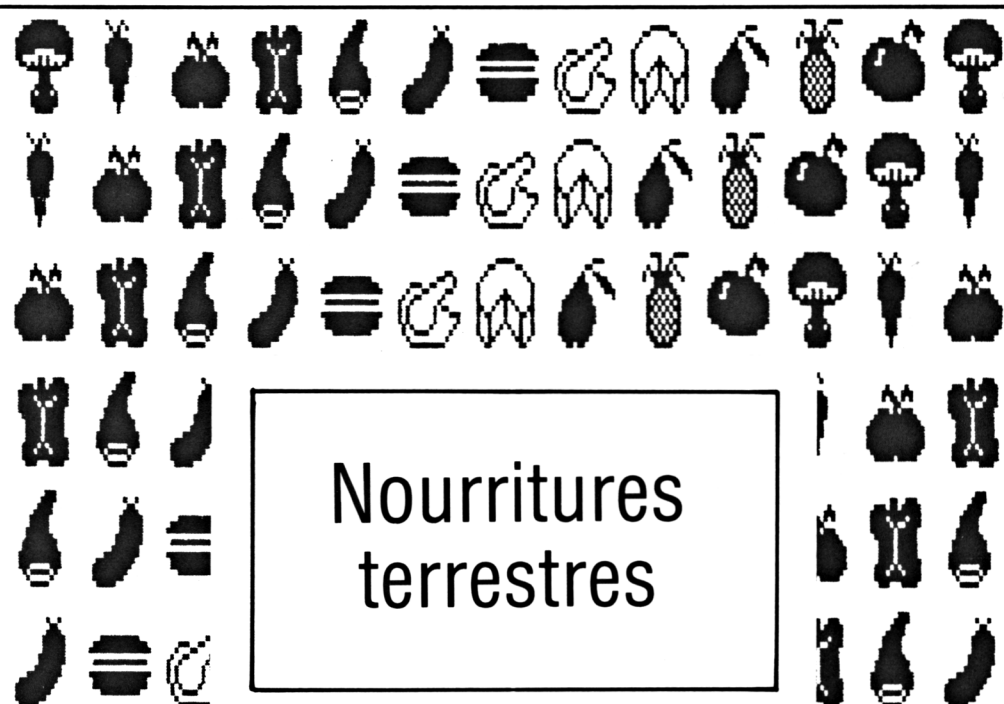
000122112221222
2224955554222
84415558

Hex.

0007FB1327222491
FF73379

SOCIAL SECURITY



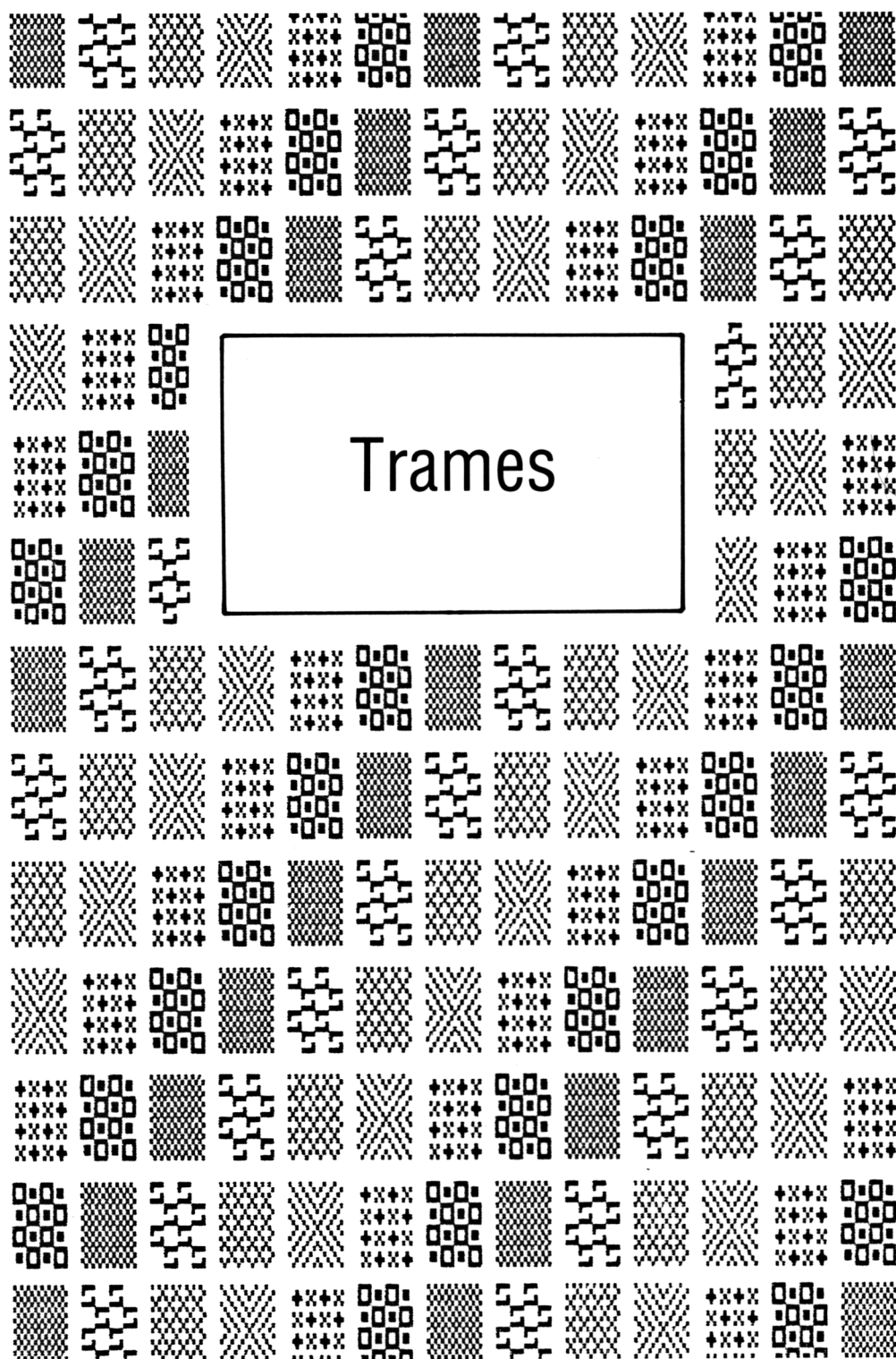


Nourritures terrestres

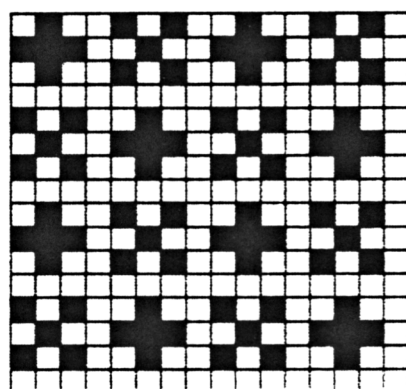




Frames



TRAME 3



Dec.

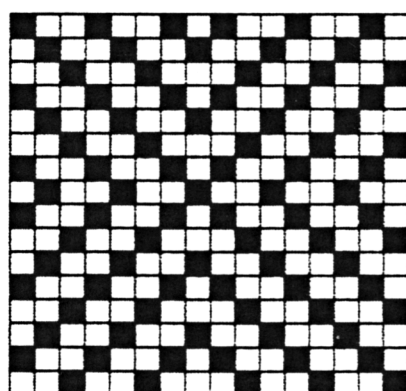
Hex.

74 74
228 228
74 74
0 0
164 164
78 78
164 164
0 0
74 74
228 228
74 74
0 0
164 164
78 78
164 164
0 0

4A 4A
E4 E4
4A 4A
0 0
A4 A4
4E 4E
A4 A4
0 0
4A 4A
E4 E4
4A 4A
0 0
A4 A4
4E 4E
A4 A4
0 0

+++
+++
+++
+++
+++
+++
+++
+++
++ ++
++ ++
++ ++
++ ++
++ ++
++ ++
++ ++
++ ++

TRAME 4



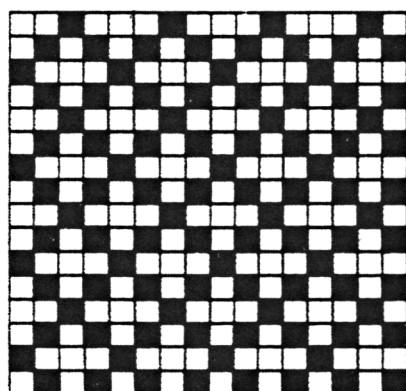
Dec.

Hex.

146 146
73 73
36 36
146 146
73 73
36 36
146 146
73 73
36 36
146 146
73 73
36 36
146 146
73 73
36 36
146 146
73 73

92 92
24 24
49 49
92 92
24 24
49 49
92 92
24 24
49 49
92 92
24 24
49 49
92 92
24 24
49 49
92 92
24 24

+++
+++
+++
+++
+++
+++
+++
+++
++ ++
++ ++
++ ++
++ ++
++ ++
++ ++
++ ++
++ ++



Dec.

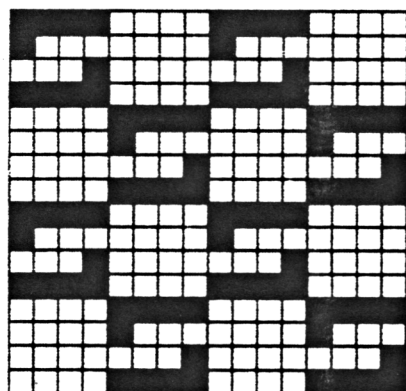
Hex.

001-0063001-0063001-0063001-0063
UNWUNWUNWUNWUNWUNWUNWUNWUNWUNW

6 6 6 6

[illegible]

U1000UTN X U1000UTN X U1000UTN



Dec.

Hex.

2400
11260
1240
1245
181
1125
1218
1245
181
1125

2400
112200
124500
121815
12124500
12126800
12124500
121815

[illegible][illegible]

Index des graphiques

| | | | |
|------------------|-----|-------------------|-----|
| Afrique | 150 | Cactus | 105 |
| Amérique du Nord | 151 | Camembert | 184 |
| Amérique du Sud | 151 | Camion | 145 |
| Ananas | 185 | Canard | 175 |
| Androïde | 125 | Cancer | 189 |
| Araignée | 174 | Cannibale | 127 |
| Arbre | 159 | Capacités | 74 |
| Arc de Triomphe | 154 | Capricorne | 192 |
| Australie | 150 | Carotte | 181 |
| | | Carreau | 83 |
| Balance | 191 | Carte | 158 |
| Bar | 101 | Cavalier blanc | 93 |
| Batterie/masse | 77 | Cavalier noir | 93 |
| Bécarre | 70 | Cerises | 99 |
| Bélier | 188 | Chameau | 110 |
| Bémol | 71 | Champignon | 181 |
| Bilboquet | 165 | Chandelier | 164 |
| Bolide A | 144 | Chariot | 105 |
| Bolide B | 144 | Chat | 176 |
| Bombe | 161 | Chaussée rétrécie | 134 |
| Bougie | 164 | Citron | 100 |
| Bric à brac | 157 | Clef | 162 |

| | | | |
|---------------------|-----|------------------------|----------|
| Clef de Fa | 67 | Hippocampe | 171 |
| Clef de Sol | 68 | Home sweet home | 129 |
| Clef d'Ut | 69 | Homme | 122 |
| Cloche | 101 | Humanoïdes | 121 |
| Cocotier | 109 | | |
| Code de la route | 133 | Intersection | 136 |
| Cœur | 83 | Italie | 149 |
| Coffre-fort | 161 | | |
| Course auto | 143 | Jack-pot | 166 |
| Cow-boy droite | 104 | Jambon | 182 |
| Cow-boy gauche | 104 | Jeep lunaire | 119 |
| Croches | 71 | Joker | 84 |
| | | | |
| Dame | 86 | LED | 76 |
| Danger | 134 | LEM | 119 |
| Danger d'éboulement | 136 | Lion | 176, 190 |
| Dauphin | 170 | | |
| Dièse | 70 | Maison part. 1 & 3 | 130 |
| Diodes | 75 | Maison part. 2 & 4 | 131 |
| Drapeau | 143 | Maître cartier | 81 |
| | | Marmite | 163 |
| Échec et mat | 89 | Ménagerie | 169 |
| École | 138 | Moine | 126 |
| Électronique | 73 | Montre | 167 |
| Éléphant | 177 | Mur | 159 |
| Éléphanteau | 177 | Musique | 65 |
| Enfant | 123 | | |
| Escargot | 173 | Notes blanches | 72 |
| Espagne - Portugal | 149 | Notes noires | 72 |
| | | Nourritures terrestres | 179 |
| Femme | 122 | | |
| Fou blanc | 94 | Oasis | 107 |
| Fou noir | 94 | Opéra | 154 |
| France | 148 | Orange | 100 |
| Fruit - machine | 97 | Ourson | 125 |
| Fusibles | 78 | Ouvrier | 123 |
| | | | |
| Gémeaux | 184 | | |
| Géographie | 147 | Paname | 153 |
| Guerre des étoiles | 115 | Papillon | 174 |
| | | Parapluie | 160 |
| Hamburger | 183 | Pas | 160 |
| Haut-parleur | 76 | Passage clouté | 137 |

| | | | |
|---------------------------|-----|----------------------|-----|
| Peintre | 124 | Ski | 111 |
| Pélican | 175 | Skieur | 113 |
| Pièce de 1 franc | 166 | Star war 1 | 117 |
| Pieuvre | 172 | Star war 2 | 117 |
| Pion blanc | 95 | Star war 3 | 118 |
| Pion noir | 95 | Star war 4 | 118 |
| Pique | 82 | Stop | 139 |
| Piquet | 112 | | |
| Piquets | 112 | Taureau | 188 |
| Pistolet | 158 | Téléphone | 167 |
| Poire | 184 | Tomate | 180 |
| Poisson | 193 | Tortue | 173 |
| Poisson-scie | 171 | Tour blanche | 92 |
| Poivron | 180 | Tour Eiffel | 155 |
| Pomme | 185 | Tour noire | 92 |
| Poulet | 183 | Trafic | 141 |
| Priorité à l'intersection | 138 | Trames | 195 |
| Pyramide | 110 | Trame 1 | 197 |
| | | Trame 2 | 197 |
| Raisin | 99 | Trame 3 | 198 |
| Raquette | 165 | Trame 4 | 198 |
| Reine blanche | 91 | Trame 5 | 199 |
| Reine noire | 91 | Trame 6 | 199 |
| Relais | 78 | Transformateur | 77 |
| Résistances | 74 | Transistor NPN | 79 |
| Roi | 85 | Transistor PNP | 79 |
| Roi blanc | 90 | Travaux | 137 |
| Roi noir | 90 | Trèfle | 82 |
| Royaume-Uni | 148 | T - unijonction | 75 |
| | | | |
| Sacré-Cœur | 155 | Valet | 87 |
| Sagittaire | 192 | Vase | 163 |
| Sapin | 113 | Verseau | 193 |
| Saucisse | 182 | Vierge | 190 |
| Scaphandrier | 124 | Virage droite-gauche | 135 |
| Scorpion | 191 | Virage gauche | 135 |
| Sens interdit | 139 | Voiture | 145 |
| Serpent | 172 | | |
| Serrure | 162 | Western | 103 |
| Signes zodiacaux | 187 | Zanclus cornutus | 170 |

Conseils de lecture

Pour approfondir vos connaissances en BASIC Amstrad et mieux connaître le système des CPC 464, 664 et 6128, P.S.I. vous propose une palette d'ouvrages utiles.

Pour maîtriser le BASIC Amstrad

- **BASIC plus 80 routines sur Amstrad** – Michel Martin (Editions du P.S.I.)

Pour pousser votre Amstrad au maximum de ses capacités ; 80 routines de simulation d'instructions qui n'existent pas en BASIC Amstrad.

- **BASIC Amstrad, CPC 464, 664 et 6128 : 1 – Méthodes pratiques ; 2 – Programmes** – Jacques Boisgontier (Editions du P.S.I.)

Pour ceux qui ont déjà pratiqué un BASIC, voici deux ouvrages de perfectionnement au BASIC Amstrad, illustrés par de nombreux programmes-exemples.

Pour programmer en assembleur et mieux connaître le système des CPC 464, 664 et 6128

- **Assembleur de l'Amstrad CPC 464 et 664** – Marcel Henrot (Editions du P.S.I.)

Une initiation complète à l'assembleur du Z80 ; un bon complément du *Clefs pour Amstrad*.

- **Clefs pour Amstrad** – Daniel Martin (Editions du P.S.I.)

Mémento présentant synthétiquement le jeu d'instructions du Z80, les points d'entrée des routines système, les connecteurs et brochages, etc. Le livre de chevet du programmeur sur Amstrad.

A paraître :

- **Périphériques et gestion de fichiers sur Amstrad, CPC 464, 664 et 6128** – Daniel-Jean David (Editions du P.S.I.)

Cet ouvrage s'adresse aux lecteurs, déjà initiés à l'Amstrad, souhaitant programmer tous les périphériques des CPC 464, 664 et 6128 et programmer des applications utilisant des fichiers sur cassette ou sur disquette.

- **Clefs pour Amstrad : 2 – Système disque** – Daniel Martin (Editions du P.S.I.)

Les compléments du tome 1 – *Système de base*, pour tout connaître du système disque des 664, 6128 et 464 version disque.

- **Turbo Pascal sur Amstrad** – Pierre Brandeis et Frédéric Blanc (Editions du P.S.I.)

Une initiation complète au Turbo Pascal des Amstrad 464, 664 et 6128, illustrée par de nombreux programmes d'un niveau de difficulté croissant : pour programmeurs avertis.

Pour être informé régulièrement de l'actualité des micros Amstrad

□ MICROSTRAD, revue bimestrielle du Groupe Tests

Pour exploiter au mieux les capacités de votre micro, vous trouverez au sommaire de chaque numéro, un rendez-vous avec les rubriques clés :

- Découvrez la face cachée de votre CPC : astuces, idées, conseils, tout pour comprendre votre micro, son anatomie, son fonctionnement, sa programmation et exploiter ses capacités graphiques et sonores.
- Domptez votre CPC 464, 664 et 6128 : passionnés, petits ou grands, spécialistes ou débutants, une information pratique et la compétence d'experts au service de votre micro.
- Programmez votre micro Amstrad : dans chaque numéro de MICROSTRAD, un cocktail de programmes (dessins, jeux, utilitaires, gestion, etc.) et des trucs de programmation.

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
BP 86
77402 Lagny-sur-Marne Cedex

Votre avis nous intéresse

- Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent livre.
- Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous adresserons votre demande à l'auteur qui ne manquera pas de vous répondre directement.

- **Ce livre vous donne-t-il toute satisfaction?**

- **Y a-t-il un aspect du problème que vous auriez aimé voir abordé?**

Comment avez-vous eu connaissance de ce livre?

- | | |
|---|-------------------------------------|
| ☐ publicité | ☐ cadeau |
| ☐ catalogue | ☐ librairie |
| ☐ boutique micro | ☐ exposition |
| ☐ autres | |

Avez-vous déjà acquis des livres PSI?

lesquels? _____

qu'en pensez-vous? _____

Nom _____ Prénom _____ Age _____

Adresse _____

Profession _____

Centre d'intérêt _____

Achevé d'imprimer
sur les presses de l'imprimerie IBP
à Rungis (Val-de-Marne 94) (1) 46.86.73.54
Dépôt légal - Janvier 1986

N° d'impression: 4864
N° d'édition: 86595-300-1
N° d'ISBN: 2-86595-300-9

SUPER GÉNÉRATEUR DE CARACTÈRES SUR AMSTRAD

Super générateur de caractères sur Amstrad vous propose un recueil de graphiques très variés (personnages, animaux, objets divers, éléments de jeux d'aventure...). Tout utilisateur de l'Amstrad 464, 664 ou 6128 pourra ainsi illustrer ou animer ses propres programmes éducatifs, ses jeux d'arcades, de rôle ou d'aventure.

La série de ces graphiques est précédée du programme BASIC permettant la création illimitée d'autres dessins au gré de votre imagination.

Si ce générateur de caractères a été écrit ici pour l'Amstrad, il est largement commenté ligne par ligne afin d'en permettre une adaptation aisée sur tout autre ordinateur individuel.



ÉDITIONS DU P.S.I.

BP 86 - 77402 LAGNY S/MARNE CEDEX - FRANCE

ISBN 2-86595-300-9

140 FF

SUPER GÉNÉRATEUR DE CARACTÈRES SUR AMSTRAD



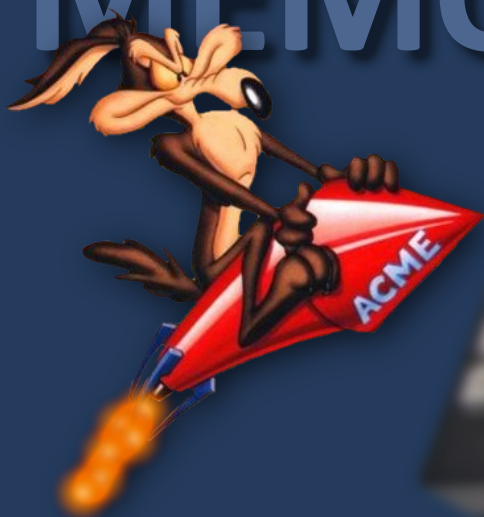


Document **numérisé**
avec amour par :

AMSTRAD

CPC 

MÉMOIRE ÉCRITE



<https://acpc.me/>