

numéro  
1

# QUASAR

- SOMMAIRE - ———— PAGES -

<i>Editorial &amp; Sommaire</i> .....	11
<i>Actus</i> .....	12
<i>Petites Annonces</i> .....	13
<i>Courrier</i> .....	14 - 15
<i>Histoires Perpendiculaires</i> .....	16
<i>Tests</i> .....	17
<i>Basic au Berceau</i> .....	18
<i>Initiation à l'Assembleur</i> .....	19
<i>Initiation au Turbo Pascal</i> .....	20 - 21
<i>Concours</i> .....	22
<i>Perfectionnement au Basic</i> .....	23
<i>Perfectionnement à l'Assembleur</i> ..	24 - 25
<i>Perfectionnement au Turbo Pascal</i> ..	26 - 27

## EDITORIAL

Salut, tout nouveau, tout beau, voici le premier numéro de Quasar CPC. Ca y est, j'en étais sûr, le petit blond à lunettes vient d'écorcher le titre de ce nouveau fanzine ; il faut prononcer "couazar" et pas "kazar" ! Enfin, là n'est pas l'essentiel. Je pense qu'il serait tout de même utile que je vous explique ce qu'est un quasar (ça y est, le petit blond à recommencé...), en voici la définition simplifiée : il s'agit d'un astre stellaire massif très lointain et très brillant qui cache encore de nombreux secrets et nous promet de grandes surprises. Ca y est, le petit blond se marre ! Vous vous demandez certainement quel est le rapport avec la choucroute ? Eh bien, relisez bien cette définition, ne pourrait-elle point s'appliquer à nos merveilleux CPC ?

Bon, trêve de philosophie ! Je vais maintenant tout vous dévoiler (ou presque) sur le contenu de Quasar. Pour ceusses qui veulent des tests de jeu à gogo, c'est mal parti. Vous ne devriez en principe trouver que celui de Moktar. Nous, on est plutôt (au pied !) branché sur la programmation ; d'ailleurs (d'ici aussi !), vous trouverez dans notre fanzine de nombreux cours de programmation (perfectionnement et initiation à l'Assembleur, au Basic et Turbo Pascal 3. Si, si...). Bref j'arrête là mon barratin et vous laisse le soin d'explorer Quasar...

Bai !

— OffseT —

Ecrivez-nous, qu'on ne soit pas à sec pour le prochain numéro qui paraîtra en Novembre 92.

La Rédac : OffseT, Zik, et le vent...

Disponibilité : envoyez 4F en timbres et une enveloppe au format A4 à l'adresse suivante :



Philippe et Gilles RIMAURO  
8, chemin des Maillos  
Quartier Lédar  
09200 SAINT-GIRONS

# ACTUS

Bon, je n'ai pas grand chose à me mettre sous la dent ce mois-ci et cette rubrique sera donc écourtée. Vous ne m'en voulez pas ? Si ?

Bref, on va commencer par une bonne nouvelle, un méga fazine vient de paraître, il s'appelle Quasar CPC. Je l'ai lu et l'ai trouvé vraiment génial ! (Aie, mes chaussures commencent me faire mal !). Il faut à tout prix que vous vous le procuriez ! Comment ? Vous le possédez déjà ? Non ?!? Si ?

Enfin, on va continuer par une autre bonne nouvelle : Lankhor "brade" ses jeux CPC, profitez-en, ils sont en général sublimes ! Personnellement je possède "Le Manoir de Mortevielle", c'est un chef d'oeuvre ! Y'a des zicks digitalisées on se croirait au ciné !

Mais cette bonne nouvelle en cache une mauvaise... Lankhor dit adieu à nos CPC... c'est dommage mais il faut dire que les fervents de jeux d'aventure sont (très) peu nombreux, j'en sais quelque chose...

Au fait, une nouvelle ROM va sortir, son nom, la INCU. C'est vraiment une révolution dans le monde du CPC ! Comme la majorité des ROMs, elle se connecte sur le ROMBOARD et vous offre alors près de 70 RSX sous BASIC et 120 routines en Assembleur ! Votre CPC devient ainsi un vrai petit monstre : il copie tous les formats (DATA, SYSTEM, etc... et les secteurs de taille 6 sont reconnus), il vous permet de gérer n'importe quel format de secteur directement depuis le BASIC. De plus, celle-ci vous permet une gestion approfondie (pour ne pas dire souterraine) de vos banks et imprimantes. Bref, le LUXE. Un détail tout de même, son prix : 7000FF HT environ (c'est cher !). Elle devrait sortir en Février 1993 en France et dès Décembre 1992 en Allemagne.

  
Offset



Ce numéro étant le premier (si on vous le dit !), nous n'avons pas de petites annonces à vous dévoiler sinon quelques personelles... Mais nous aimerions (voire même nous exigeons) que vous nous en envoyez. Pour cela, il faut que vous nous écriviez à l'adresse du sommaire.

Cherche "THE MUSIC MACHINE" ou tout autre logiciel équivalent en vue d'exploiter la sortie MIDI du CPC. Me contacter à l'adresse du fanzine ; s'adresser à Gilles RIMAURO.

Cherche contact pour créer jeux et démos : graphiste ou animateur. Ecrire à l'adresse du fanzine.

Achèterais La Bible du CPC et/ou les Clefs pour Amstrad à un prix raisonnable. Philippe et Gilles RIMAURO à l'adresse du fanzine.

Vends CPC6128 couleur avec extensions Vortex et DK'Tronics 256ko, Multiface II, Hacker, souris AMX et Kempson, crayon optique, lecteurs externes 3"1/2, 5"1/4 et 3", lecteur de cassettes, ampli stéréo, ROMBOARD avec RODOS, ROMDOS, UTOPIA, MAXAM et PROTEXT, imprimante couleur 24 aiguilles, environs 300 logiciels, livres, revues, etc..

PRIX : 5000FF TTC

Contacteur : TYHAKHRUS à l'église !

Recherche nouveaux membres pour participer à l'élaboration de Quasar. Quel que soit votre niveau, si vous voulez nous aider, contactez-nous ! On trouve toujours quelque chose à faire !

Envoyez-nous POKES, P.A., BIDOUILLES et lettres d'insultes. On compte ouvrir de nouvelles rubriques...

Zik



Petites Annonces

# Courrier

Quoique nous n'ayons évidemment pas de courrier, nous avons quand même décidé d'insérer cette rubrique dans le fanzine. Tout d'abord pour vous signaler qu'elle existe, puis pour prendre de la place et donc grossir Quasar !

En conséquence il faut bien remplir ce gros vide au-dessous de moi ! Et voilà, j'ai trouvé ! Je vais coller un programme (NDOffset : Mais ! Eh ben ! C'est mon programme ! Voleur !!!) de démonstration de rasters (si vous ne savez pas ce que c'est allez jeter un oeil

(n'oubliez pas de le récupérer ! (NDOffset(ironique) : Ha ! Ha ! Ha ! J'la connaissais pas !) dans la rubrique "Perfectionnement à l'Assembleur")...

Zikourrier : écrivez-moi !

Et moi alors, je signe aussi !  
Après tout, c'est mon prog !

L'Offset de service

```
10 '  
20 ' Lanceur de l'exemple de RASTERS  
30 '  
40 MODE 1  
50 CALL &BC00  
60 PRINT "Votre CRTC est-il un UM6845R ?"  
70 PRINT  
80 PRINT "  -1- Oui !"  
90 PRINT "  -2- Non !"  
100 PRINT "  -3- J'sais pas !"  
110 PRINT "  -4- Tu disais ?"  
120 PRINT  
130 PRINT "Réponse : ";  
140 CLEAR INPUT  
150 a$=INKEY$  
160 IF a$<"1" OR a$>"4" THEN GOTO 150  
170 IF a$<>"1" THEN OUT &BC00,3:OUT &BD00,8  
180 PRINT a$  
190 PRINT  
200 IF a$="1" OR a$="2" THEN PRINT "Ok !"  
210 IF a$="3" THEN PRINT "C'est pas grave !"  
220 IF a$="4" THEN PRINT "Non rien !"  
230 MEMORY &4FFF  
240 LOAD "RASTER.BIN",&5000  
250 INK 1,1:MODE 2  
260 FOR x=144 TO 1 STEP-1  
270 PRINT " CPC      ";  
280 SOUND 2,x,3,6  
290 NEXT  
300 PRINT " EXEMPLE DE RASTERS PAR XOFFSETX. LA PROCHAIN  
E FOIS JE GENERERAI UNE PAGE ECRAN DE 32Ko. Promis !"  
310 CALL &5000  
320 CALL &BC02  
330 MODE 1  
340 OUT &BC00,3:OUT &BD00,142
```

Oups !!!

Les X qui entourent mon pseudo s'obtiennent en faisant CTRL+X.

```
10 '  
20 ' Datas pour le codage du source assembleur  
30 '      par Offset 1992  
40 '  
50 MODE 2:MEMORY &4FFF  
60 y=990:RESTORE 1000  
70 FOR x=&5000 TO &52D7 STEP 8  
80 y=y+10  
90 LOCATE 1,1:PRINT "Lecture ligne ";USING"###";y  
100 q=0:FOR z=x TO x+7  
110 READ a$:a=VAL("&"+a$):q=q+a:POKE z,a  
120 NEXT  
130 READ a$:a=VAL("&"+a$):IF q<140 NEXT  
150 PRINT "Vérification terminée, sauvegarde du  
      fichier RASTER.BIN en cours..."  
160 SAVE "raster.bin",b,&5000,&2D8  
170 PRINT "Toutétoké !"  
180 END  
1000 DATA F3,2A,38,00,22,3C,52,21,226  
1010 DATA FB,C9,22,38,00,FB,01,06,320  
1020 DATA BC,ED,49,01,23,BD,ED,49,409  
1030 DATA 01,07,BC,ED,49,01,23,BD,2DB  
1040 DATA ED,49,01,02,BC,ED,49,01,32C  
1050 DATA 32,BD,ED,49,01,01,BC,ED,3D0  
1060 DATA 49,01,30,BD,ED,49,01,10,27E  
1070 DATA 7F,ED,49,0E,04,ED,49,01,2FE  
1080 DATA 00,F5,ED,78,1F,D2,42,50,3DD  
1090 DATA 01,01,7F,ED,49,0E,40,ED,2F2  
1100 DATA 49,76,76,00,00,00,00,135  
1110 DATA 00,00,00,00,00,00,00,000  
1120 DATA 00,00,00,00,00,00,00,000  
1130 DATA 00,00,00,00,00,00,00,0AF  
1140 DATA ED,79,21,71,52,06,7F,7E,34D  
1150 DATA FE,00,CA,AB,50,23,ED,79,44C  
1160 DATA 01,02,BC,ED,49,06,BD,7E,336  
1170 DATA ED,79,23,00,00,00,00,189  
1180 DATA 00,00,00,00,00,00,00,000
```



```

1190 DATA 00,00,00,00,00,00,00,00,00,00
1200 DATA 00,00,00,00,00,00,00,00,00,00
1210 DATA C3,75,50,3E,54,ED,79,01,381
1220 DATA 02,BC,ED,49,01,32,8D,ED,3D1
1230 DATA 49,21,71,52,06,02,C5,7E,278
1240 DATA 47,23,7E,FE,00,CA,CC,50,3CC
1250 DATA 70,C3,C0,50,21,71,52,70,397
1260 DATA C1,10,EB,76,00,00,00,00,232
1270 DATA 00,00,00,00,00,00,00,00,000
1280 DATA 00,00,00,00,00,00,00,00,000
1290 DATA 00,00,00,00,00,01,00,7F,080
1300 DATA ED,49,21,3F,52,7E,FE,00,364
1310 DATA CA,01,51,23,ED,79,C3,F5,45D
1320 DATA 50,3E,58,ED,79,0E,01,ED,348
1330 DATA 49,ED,79,06,04,C5,2B,E5,38E
1340 DATA 7E,47,2B,7E,FE,00,CA,1D,353
1350 DATA 51,70,C3,11,51,E1,70,C1,3F8
1360 DATA 10,EB,01,00,7F,ED,49,00,2B1
1370 DATA 00,00,00,00,00,00,00,00,000
1380 DATA 00,00,00,00,00,00,00,00,000
1390 DATA 00,00,00,00,00,00,00,00,000
1400 DATA 00,00,00,00,06,17,C5,06,0E8
1410 DATA 7F,0E,54,3E,44,16,55,1E,1EC
1420 DATA 57,26,5F,2E,4B,ED,79,ED,3A8
1430 DATA 51,ED,59,ED,61,ED,69,ED,528
1440 DATA 61,ED,59,ED,51,ED,79,ED,538
1450 DATA 49,00,00,00,00,00,00,00,049
1460 DATA 00,00,00,00,00,00,00,00,000
1470 DATA 00,00,00,00,00,00,00,00,000
1480 DATA 00,00,00,00,00,00,00,00,000
1490 DATA 00,00,00,00,00,00,00,00,000
1500 DATA 00,00,00,00,00,00,00,00,000
1510 DATA 00,00,00,00,00,00,00,00,000
1520 DATA 00,00,00,00,00,00,00,00,000
1530 DATA 00,C1,10,9A,00,00,00,00,16B
1540 DATA 00,00,00,00,00,00,00,00,000

```

```

1550 DATA 00,00,00,00,00,00,00,00,000
1560 DATA 00,00,00,00,00,00,00,00,000
1570 DATA 00,00,00,00,00,00,00,00,000
1580 DATA 00,00,00,00,06,7F,3E,5C,11F
1590 DATA ED,79,0E,01,ED,49,3E,52,33B
1600 DATA ED,79,01,0E,F4,ED,49,01,3A0
1610 DATA C0,F6,ED,49,AF,ED,79,01,502
1620 DATA 92,F7,ED,49,01,45,F6,ED,4E8
1630 DATA 49,06,F4,ED,78,01,82,F7,422
1640 DATA ED,49,01,00,F6,ED,49,17,37A
1650 DATA DA,3F,50,F3,2A,3C,52,22,336
1660 DATA 38,00,FB,01,06,BC,ED,49,32C
1670 DATA 01,19,8D,ED,49,01,07,BC,2D1
1680 DATA ED,49,01,1E,8D,ED,49,01,349
1690 DATA 02,BC,ED,49,01,2E,8D,ED,3CD
1700 DATA 49,01,01,BC,ED,49,01,28,266
1710 DATA 8D,ED,49,C9,00,00,00,46,302
1720 DATA 44,56,46,53,55,52,40,4B,265
1730 DATA 4B,43,4B,53,55,52,40,46,259
1740 DATA 44,56,46,54,54,54,54,44,274
1750 DATA 5E,5C,5D,4D,4A,4C,45,4F,28E
1760 DATA 43,4A,47,4D,4A,4C,45,44,240
1770 DATA 5E,5C,5D,54,54,54,54,00,267
1780 DATA 00,44,32,55,33,57,33,5F,1E7
1790 DATA 34,53,34,5F,34,57,35,55,22F
1800 DATA 35,44,35,54,35,5E,34,4A,213
1810 DATA 34,43,34,4A,33,5E,33,54,20D
1820 DATA 32,5C,31,4C,31,4E,30,43,1FD
1830 DATA 30,4E,30,4C,2F,5C,2F,54,208
1840 DATA 2F,44,2F,46,30,53,30,5B,1F6
1850 DATA 30,43,31,5B,31,53,32,46,1FB
1860 DATA 33,44,33,54,34,58,34,4D,20B
1870 DATA 34,4F,33,4D,33,58,32,54,214
1880 DATA 31,56,31,52,30,59,30,4A,20D
1890 DATA 30,59,31,52,31,56,32,54,219
1900 DATA 33,00,00,00,00,00,00,00,033

```



**NDOffset :**

Tiens ! Voilà de la place libre !

Je vais en profiter pour vous dire comment il faut s'y prendre pour obtenir 255 vies au début de chaque niveau de Moktar. Je sais bien que ce jeu est (très) facile et ne nécessite pas autant de vies mais je vous donne tout de même le poke. Sous disco, recherchez la chaîne hexa 3E,05,21,10,00,32,48,01 et remplacez le 05 par FF. Avec la Multiface II, c'est plus simple : faites un Poke &2202,&FF.

N'étant pas franchement spécialiste dans la recherche des vies infinies dans le jeu, et me demandant encore comment j'ai fait pour trouver ces quelques pokes (j'exagère un peu), je me mets à genoux et vous supplie de nous envoyer le résultat de vos élucubrations...

# Histoires Perpendiculaires

Alors, que je vous prévienne tout de suite, dans cette rubrique vous rencontrerez un peu de tout, c'est normal !

Tout d'abord, je vais vous expliquer un jeu. Pour commencer, munissez-vous d'un copain et de deux pièces de monnaie identiques. Maintenant, jouez... Ben quoi ? Ah oui ! Comment jouer ? Eh bien, chaque joueur va présenter une face de la pièce à l'autre (attention ! Chaque joueur doit agir en même temps que l'autre !). Ainsi, il y aura différentes combinaisons possibles. Chaque joueur portant un numéro (1 et 2), voilà ce qui arrivera dans chacun des cas :

- Deux fois pile : 9 euros au joueur 1 (-9 au joueur 2),
- Deux fois face : 1 euro au joueur 1 (-1 au joueur 2),
- Mixte : 5 euros au joueur 2 (-5 au joueur 1).

Allez, jouez, vous allez voir que c'est équitable. Il est évident qu'il faut faire au moins une dizaine de coups et effectuer le total des sommes.

Et maintenant, si je vous disais que l'on peut, si l'on est le joueur 2, gagner pratiquement chaque partie ? Ca vous intéresse ?

Grâce à un calcul mathématique de statistiques, on constate que si le joueur 2 propose pile à la fréquence  $3/10$ , il gagne en moyenne 0,8 euros par coups. Je vous sens ruminer alors voici un exemple : sur trente tours, il faut que le joueur 2 présente pile 9 fois ( $3/10 = 9/30$ ), et il pourra alors ainsi gagner un maximum de 24 euros. Si en adoptant cette technique vous en venez à perdre, c'est vraiment que vous le faites exprès !

Au fait, pour ceusses qui sont intéressés par la démonstration mathématique, il n'y a qu'à m'écrire !

Ensuite, de quoi vais-je bien pouvoir vous parler ? Ca y est, j'ai une idée, je vais vous parler météo ! Eh bien on peut pas dire qu'il fait très beau ! Vous n'en avez rien à faire ? Bon, changeons de sujet !

Bof ! Pfff ! Je pense finalement que je vais m'arrêter là pour aujourd'hui ! Tchao !

OffseT



# TESTS

Récemment, un ami (ou copain si vous préférez) a acheté Le dernier né de L'éditeur Titus : Moktar. D'où le pourquoi du comment du parce que de ce test...

Moktar est un jeu dans le style de Rick Dangerous, c'est dire un jeu de plate-forme avec un scrolling multi-directionnel qui a une particularité dont je m'en irai (au revoir !) vous causer plus tard. Vous êtes Moktar et vous devez aller sauver la Zoubida. Les sept niveaux (en écran à la taille standard) sont truffés de pièges (dont certains sont assez vicieux) et d'ennemis qui sont différents de niveaux en levels. Vous devrez (si vous achetez le jeu !) actionner des manettes, ruser pour trouver le passage le plus sûr (qui est souvent le plus long). On peut accéder directement à un niveau grâce à un code trouvé durant le jeu. Comme je suis gentil (hof !), je vais vous en donner (ou plutôt prêter) un : 5772.

Revenons au scrolling multi-directionnel. Le-dit truc multi-chose est contrôlable dans les quatre directions bien connues (sans blague !), et tellement bien connues que je ne les citerai pas (ces directions sont bien sûr gaut, hauche, broite et das) !

Vous avez tous remarqué que je suis revenu (bonjour !) à la ligne ! Cela veut simplement dire que je change de sujet (non ?! Si !) ! Ce sujet est l'animation qui est bien réalisée (sauf quand les sprites clignotent et quand leur vitesse est divisée par leur nombre !). Elle est assez fluide (quoique le héros patine, traditionnellement).



Les graphismes, en mode 0, sont clairs, colorés et surtout (et donc) beaux. Les musiques, qui sont au nombre de trois, sont agréables et accompagnent le joueur tout au long de son aventure. Ce jeu à la difficulté savamment dosée, fait partie de ceux auxquels on ne peut s'arrêter de jouer avant d'en être venu à bout.

Mes notes sont :

- GRAPHISME	: 17/20
- ANIMATION	: 14/20
- MUSIQUE	: 16/20
- BRUITAGES	: ----
- RICHESSE	: 18/20
- JOUABILITE	: 17/20
- NOTE GLOBALE	: 16.40

Zik

# Basic au Berceau

Nous avons décidé d'insérer cette rubrique dans Quasar CPC pour enseigner Le Basic à ceux qui veulent se lancer dans La programmation (c'est haut !) sans ouvrir Le manuel de Leur CPC (Bouh ! Les fainéants !).

Je vais commencer par vous expliquer comment afficher un mot (une phrase ou voire même une lettre). Il vous faut taper la commande PRINT suivie de ce que vous voulez afficher entre deux guillemets. Ainsi, si je veux afficher "Bonjour !", je taperai la ligne de commande suivante : PRINT"Bonjour !". Cette commande sera bien sûr suivie d'une validation (appui sur la touche RETURN ou ENTER). On peut aussi afficher le contenu d'une variable mais il faut lui attribuer un contenu. Une variable sans extension contient des nombres réels (exemple, des variables comme a, b, c, gaga ou cekevouvoulez) et une variable avec l'extension "\$" contient une chaîne de caractères (exemple, des variables comme a\$, b\$, c\$, flop\$ ou cometuveu\$). Il y a quatre façons d'attribuer une valeur à une variable :

- une attribution directe (a=12 ; a\$="Bonjour"),
- une attribution par saisie (INPUT a ; INPUT a\$),
- une attribution par données (par DATA et READ),
- une attribution par l'assembleur (par CALL ou RSX).

Donc, pour afficher "Bonjour !" en passant par une variable, on tapera la ligne suivante : a\$="Bonjour !" (+ validation) puis PRINT a\$ (+ validation). Il est gênant de devoir valider après chaque instruction, on met donc le signe ";" pour les accorder entre elles. La ligne précédente deviendra ainsi : a\$="Bonjour";PRINT a\$. On peut aussi créer un programme en faisant précéder les commandes d'un numéro qui est le numéro de ligne ; le programme sera exécuté grâce à la commande RUN. L'intérêt d'un programme est que les commandes qui le composent restent en mémoire (sauf initialisation) et peuvent être visualisées à l'aide de la commande LIST.

Maintenant, nous allons voir comment demander au gars qui est devant le clavier de taper quelque chose qui sera stocké dans une variable. Il faut pour cela utiliser la commande INPUT qui s'emploie de la manière suivante :

```
INPUT a
ou
INPUT a$
```



ou, si l'on veut afficher une phrase :

```
INPUT"Entrez un nombre : ",a
ou
INPUT"Entrez une phrase : ",a$
```

Voici un petit programme d'exemple :

```
10 CLS ' Efface l'écran
20 INPUT"Quel est votre age ";age ' Devinez !
30 INPUT"Entrez l'année ; ",annee ' Mais encore ?!
40 naissance=annee-age ' Comme en Maths !
50 PRINT"Vous etes né(e) en "naissance ' Affichage
RUN
```



Si cette rubrique vous intéresse, écrivez-le à l'adresse du fanzine (vous pouvez nous donner des conseils). Par rapport au nombre de lettres et au nombre de lecteurs, nous verrons si nous la gardons ou pas...

Zik



# Initiation à l'Assembleur

Chassez que ve j'ai pour prendre au berceau ! Autrement dit, je considère que vous n'avez jamais vu quelque Listing Assembleur que ce soit, et que vous n'en connaissez aucune commande ! OK ? C'est parti !

Il est primordial que je commence par vous expliquer la commande LD (prononcer comme Load). Vous devez savoir qu'en Assembleur, on ne travaille pas avec des variables mais avec des registres. La différence est que ceux-ci ne peuvent contenir que des nombres et que leur nombre est limité, leur capacité aussi. Ainsi, vous devez également faire la différence entre les registres simples et les doubles.

Les simples ne peuvent contenir que des nombres compris entre 0 et 255 alors que les doubles peuvent aller jusqu'à 65535. Mais là où ça se corse, c'est quand un registre double est formé par deux registres simples. Non, ne hurlez pas ! Voici des exemples : prenons les registres simples B et C ; ils formeront le registre double BC. Et comme je suis gentil, je vous donne la liste de "tous" les registres doubles existants : AF, BC, DE, HL, IX, IY, etc...

Pour ce premier cours, nous ne nous servons que de A et HL qui sont les plus utilisés. Revenons donc à LD : si vous faites un LD A,10 cela correspond un A=10 en Basic. Simple non ? De même, LD HL,65535 mettra 65535 dans HL. Oui, mais là c'est un peu plus complexe, et il vaut mieux raisonner en hexadécimal (base 16). Ainsi, 65535=&FFFF et on lit alors très simplement que H contient &FF, tout comme L (&FF=255). De même, si l'on émet un LD HL,&01B3, H=&01 et L=&B3. Compris ? Parfait ! Maintenant que vous connaissez LD par coeur, je vais vous fournir deux vecteurs (ou routines système). L'un équivalent à "PRINT CHR\$(A)", c'est le &BB5A et l'autre équivalent à "LOCATE H,L", c'est le &BB75. Non ! On ne se sert pas de LD pour les exécuter, mais de la commande CALL. C'est très simple, il faut un "CALL vecteur" et ça marche ! Alors, voici un petit programme qui affiche le code ascii 65 aux coordonnées 1,1 :

```
ORG &5000 ; Implantation du
           ; prog en &5000
LD HL,&0101 ; H=1 et L=1
CALL &BB75 ; LOCATE H,L
LD A,65 ; A=65
CALL &BB5A ; PRINT CHR$(A)
RET ; Retour au Basic
```



Et pour les ceusses qui n'ont pas ouvert la doc de leur Assembleur, je dis même comment lancer cette routine : sous Basic, faites un CALL &5000 ; il est évident que le programme a été assemblé...

Salut et la prochaine ! On parlera des data (DB et DW), des labels et des boucles (JP, DJNZ).

 Offset

# Initiation au Turbo Pascal

Je sais bien que Le Turbo Pascal est une denrée rare sur CPC, mais je pense qu'il ne faut pas pour autant Le délaïsser. Contrairement à ce que vous êtes en train de marmorer Le Turbo Pascal peut présenter un grand intérêt ; comme vous ne programmerez pas toute votre vie sur CPC, il est fort utile de maîtriser un langage quasi universel. Ainsi, vous pourrez dériver sur ST, AMIGA ou PC sans trop de problèmes !

Ça y est, convaincu ? Alors on y va ! Primo, vous devez savoir que tout programme Turbo commence par la commande "PROGRAM nom;" et que chaque instruction doit se terminer par un ";". De plus, on doit déclarer ses variables en début de programme grâce à la commande "VAR variable:type;". Ah ! Le type, oui, voici les plus simples : INTEGER (entier), REAL (réel), CHAR (caractère), STRING x (chaîne de caractères de longueur x) et BOOLEAN (bouléen). Après la déclaration des variables il vous faut un "BEGIN", pas de ";" ici (ça n'est pas considéré comme une instruction). Ensuite, c'est parti pour le programme ! Mais attention, à la fin, il faut un "END."

Ouf, après ces quelques notions de base, je m'en vais vous expliquer quelques commandes :

- "WRITE(ce qu'il y a à écrire);", cette instruction fonctionne comme le "PRINT" du Basic mis à part que les chaînes de caractères seront placées entre "".

- "GOTOXY(coordonnée X, coordonnée Y);", alors là, pas de problème, c'est comme "LOCATE" !

- "IF condition THEN ... ELSE ...;", ici no comment hormis le fait qu'il ne doit pas y avoir de ";" entre le "IF" et le "THEN" et, s'il y a un "ELSE", il n'en faut pas non plus avant.

- "REPEAT ... UNTIL condition;", bon, je suis obligé de tout vous expliquer vu que ça n'a pas d'équivalent en Basic. Le Turbo va exécuter le morceau de programme situé entre les deux bornes jusqu'à ce que la condition suivant le "UNTIL" soit vraie.

- "KEYPRESSED;", c'est une fonction qui rend vrai si une touche est pressée et faux dans le cas contraire.

- "CLRSCR;", c'est exactement comme le CLS du Basic.

Ce sera tout pour les commandes ce numéro-ci. Par contre, vous devez savoir que pour les attributions de variable on ne joue pas avec le "=" du Basic qui est réservé aux conditions, mais plutôt avec un étrange ":= " qui se lit "reçoit" (ex : pif:= 'paf';).



Avant de vous quitter, je vous conseille de jeter un coup d'oeil sur le listing de ce cours qui fait rebondir une lettre sur les bords de l'écran en se servant exclusivement de ce que j'ai expliqué dans ce cours.

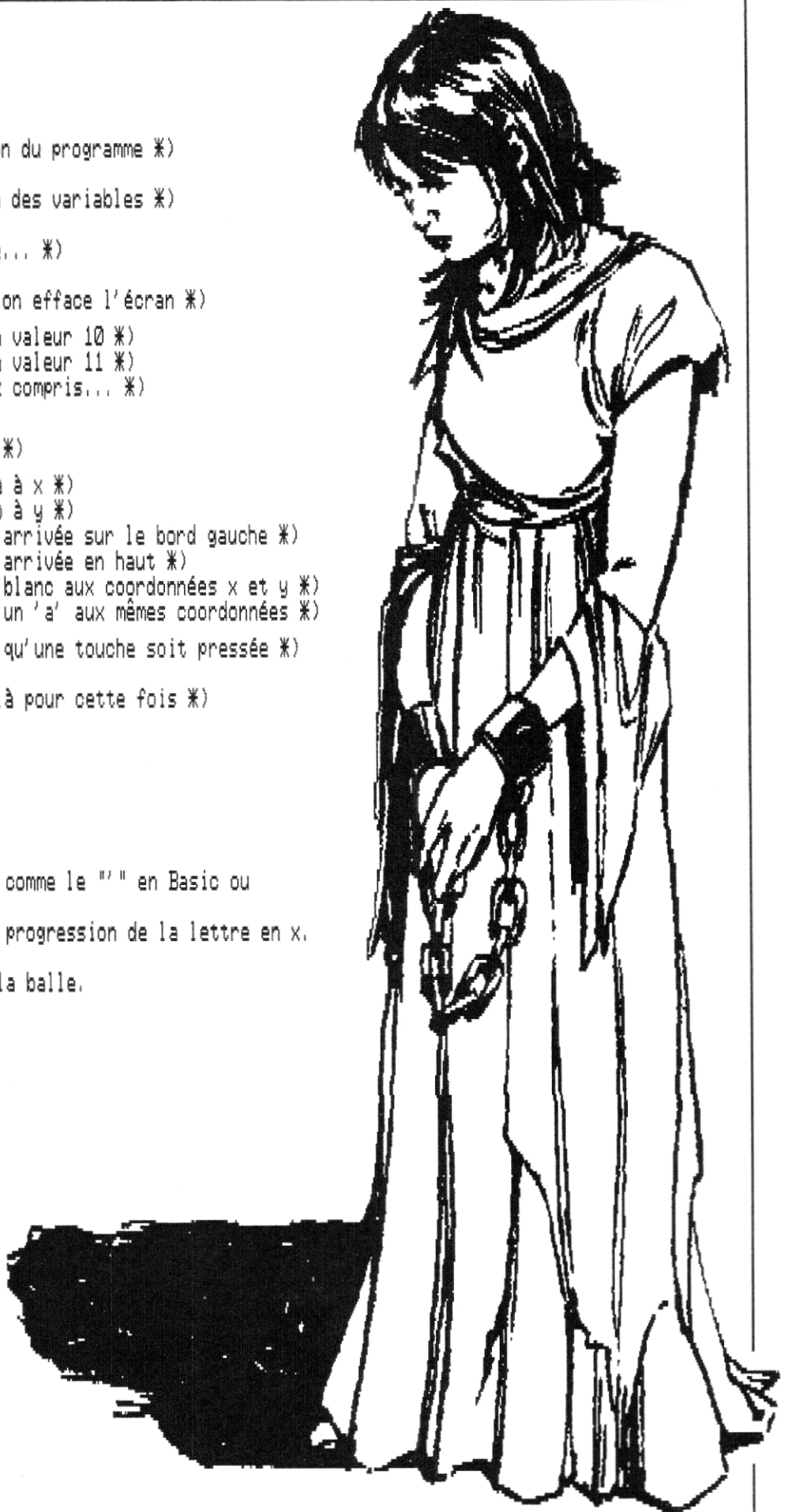
Offset

```

Program Exemple;      (* Introduction du programme *)
Var a,b,x,y:Integer;  (* Déclaration des variables *)
Begin                (* On commence... *)
Clrscr;              (* Comme CLS, on efface l'écran *)
x:=10;               (* x reçoit la valeur 10 *)
y:=11;               (* y reçoit la valeur 11 *)
a:=1;                (* Vous m'avez compris... *)
b:=1;                (* Idem *)
Repeat              (* Répéter... *)
  x:=x+a;            (* On ajoute a à x *)
  y:=y+b;            (* On ajoute b à y *)
  If x<2 Then a:=1; (* On teste l'arrivée sur le bord gauche *)
  If y<2 Then b:=1; (* On teste l'arrivée en haut *)
  Gotoxy(x,y);Write('a'); (* On affiche blanc aux coordonnées x et y *)
  Gotoxy(x,y);Write(' '); (* On affiche un 'a' aux mêmes coordonnées *)
Until Keypressed;   (* Jusqu'à ce qu'une touche soit pressée *)
End.                 (* On arrête là pour cette fois *)

```

NOTA : entre "(\*", c'est les remarques comme le ";" en Basic ou le ";" en Assembleur.  
 a est la variable qui génère la progression de la lettre en x.  
 b c'est pareil mais pour y.  
 x et y sont les coordonnées de la balle.



# Concours

L'autre soir, je me posais une question : Y a-t-il beaucoup de cpcistes qui connaissent réellement leur ordi ?

Afin d'y trouver une réponse, je vous propose ce concours histoire de voir de quoi vous êtes capable. Je vous pose donc une série de questions, répondez sur papier libre, et envoyez-le nous. Le meilleur d'entre vous gagnera un câble de téléchargement, alors pensez bien à préciser votre adresse (qui a dit hof ?).

Voici les questions dont il est question :

- 1/ Fastoche : quel est le format des disquettes Amstrad CPC ?
- 2/ Bof : citez quatre commandes Basic qui ne fonctionnent pas sur 464.
- 3/ Aïe ! A quoi servent les registres 12 et 13 du CRTC ?
- 4/ Dur : comment commande-t-on au Gate Array de changer la couleur du border en cours de balayage (Basic ou Assembleur, comme vous voulez) ?

5/ A quoi sert le FDC ?

6/ Quel est le modèle du CRTC sur CPC ?

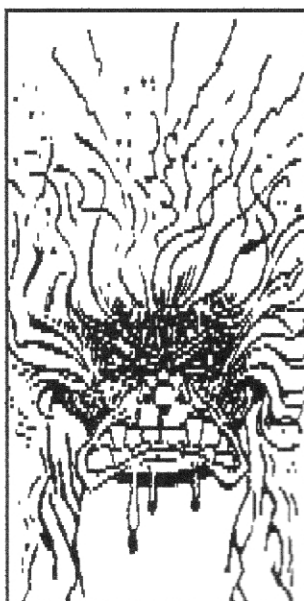
7/ Que signifient les appellations CRTC, FDC et PSG ?

8/ Pour vous départager : c'est quoi une rupture d'écran (non, ce n'est pas quand on y enfonce une massue) ?

Bonne Chance !

Mais attention, ce concours n'est valable que jusqu'au quinze Avril de l'an de grâce deux mille six cent quarante et un. Les réponses seront analysées en présence de Maître Offset, huissier de jus passé (ne soyons pas vulgaire...).

 Offset





# Perfectionnement au Basic

Cette rubrique a pour but d'approfondir votre connaissance du Basic. Je commencerai par vous parler des tableaux de variables, puis de La gestion des fichiers ASCII et enfin de La manipulation des ports du CRTC et du Gate Array.

Nous allons donc parler des tableaux de variables. Vous pouvez dès lors remarquer qu'il y a un programme d'exemple que je vous commenterai. Mais tout d'abord : un tableau de variables qu'est-ce que cette bête ? Et bien c'est tout simplement (Hum !) un ensemble de variables ordonnées et désignées par des nombres. Je m'explique : si vous faites un "DIM a(10)" vous venez de créer un tableau nommé "a" qui contient 11 variables numérotées de 0 à 10. C'est bien beau de savoir déclarer un tableau de variables, mais faut savoir le remplir de données. Et moah, je vais vous l'expliquer (si, si !) : par exemple, pour mettre le nombre 153 dans la case x de ce tableau, il faut tapoter sur votre cher clavier : a(x)=153. Simple isn't it ?

Voici comment se présente un tel tableau :

Cases(x)	0	1	2	3	4	5	6	7	8	9	10
Contenus											

Ce type de tableau s'appelle un tableau à 1 dimension. Mais on peut aussi faire un tableau à 2 dimensions dont voici le schéma.

Cases	1	2	3	4	5	6	7	8	Etc.
1									
2									
3									
Etc.									

Ce type de tableau se déclare de la manière suivante :

DIM a(x,y) : - x représente le nombre de cases horizontalement (1ère dimension),  
- y représente le nombre de cases verticalement (2ème dimension).

Pour attribuer une valeur à une case, c'est comme pour un tableau à 1 dimension, sauf qu'il faut aussi préciser le numéro de la case de la 2ème dimension ; c'est-à-dire comme suit : a(x,y)=valeur à attribuer.

Je ne vous représenterai pas un tableau à 3 dimensions, et je vous laisse réfléchir à un tableau 13 dimensions, qui est le plus complexe qu'accepte le Basic.

Je vais maintenant commenter le programme. Celui-ci prend des codes ASCII au hasard et les stocke dans un tableau à 2 dimensions (1.110 à 140). La 1ère dimension de ce tableau représente les colonnes de l'écran et la 2ème dimension représente le nombre de lignes. Le programme affiche les codes ASCII à l'écran à l'aide du tableau de variables ainsi constitué (1.180 à 220) puis, suivant les touches du clavier enfoncées, il déplace un pointeur et réaffiche le décor en se servant du tableau (1.260 à 370).



Dans le prochain numéro, je parlerai de la gestion des fichiers ASCII.

Zik

# Perfectionnement à l'Assembleur

Dans cette rubrique je vais tenter de vous expliquer comment on peut utiliser 100% des capacités de son CPC. Vous vous en doutez certainement, ce ne sera pas en passant par les routines système qui sont souvent bien trop lentes...

Pour ce premier numéro, je vais vous expliquer comment il faut s'y prendre pour réaliser des rasters comme on en voit dans toutes les bonnes démos.

J'espère que vous n'en êtes plus à penser qu'il s'agit là de sprites car, dans ce cas, je ne peux plus rien pour vous... Mais non je blague, si je suis là, ce n'est pas pour vous démoraliser, mais au contraire pour vous aider à faire haver vos copains qui se vantent sans cesse de leur console ou de leur ST.

Ne trouvez-vous pas qu'il est triste de devoir travailler en mode 0 pour disposer de 16 couleurs alors que l'on peut facilement en avoir 27 en mode 2 (et même plus sur CPC+)? Eh bien, la solution, c'est les rasters. La technique consiste à changer la couleur de telle ou telle encre durant le balayage de l'écran; ainsi, on peut facilement donner l'illusion d'avoir des tonnes de couleurs affichées en même temps!

La petite merveille qui permet cela, c'est le GATE ARRAY (Gai Taré pour les intimes). Il permet entre autre (je dis bien entre autre!) de gérer les banks, les roms et les couleurs. Il est entièrement contrôlé à partir du port &7F et sa programmation se fait en deux temps.

La première étape consiste à signaler quelle encre on désire modifier. Là, c'est très simple, on envoie directement le numéro de l'encre à l'aide du OUT (C), registre. Il y a tout de même une exception: pour le border, on doit envoyer la valeur 16.

Ensuite, pour spécifier la couleur, c'est un peu plus complexe. Celle-ci doit lui être envoyée en numérotation hardware (voir tableau). De plus, pour compléter les bits (eh oui, notre Gai Taré ne s'occupe pas que des couleurs!) on doit y ajouter 64. Compris? Parfait!

Je vais maintenant vous détailler le fonctionnement du OUT, ce qui est essentiel pour la suite. On doit tout d'abord mettre dans B le port auquel on veut téléphoner, et dans un autre registre simple, ce qu'on veut lui dire. Ainsi, vous aurez un programme d'allure suivante:

```
LD BC, &7F00
OUT (C), C
LD C, 64+11
OUT (C), C
```

Ce programme, comme vous l'avez remarqué, met l'encre 0 en blanc brillant. Malgré les apparences, il n'y a pas d'erreur; le OUT (C) peut en effet paraître étrange (un OUT (B) aurait été plus logique), mais cette curiosité est la conséquence d'un choix de câblage particulier du Z80 dans le CPC. Enfin, on fait avec...

Maintenant, quelques précisions annexes. Pour ne pas être dérangé, je vous conseille de geler les interruptions, mais pas avec DI car les HALT seraient alors perturbés. Personnellement, je poke un EI, RET en &38. J'allais z'oublier! Pour que votre raster soit toujours au même endroit sur l'écran, il est important d'attendre que le balayage vidéo revienne en haut de l'écran, et ainsi de positionner votre raster en fonction de la synchronisation verticale. C'est le PPI qui permet cela, mais ça, c'est une autre histoire. Si vous n'avez pas envie de vous lancer là dedans, utilisez simplement le vecteur &BD19.

Voilà le programme d'exemple et entraînez-vous, car la prochaine fois, on mettra les rasters en mouvement...

 Offset

```

; Exemple de rasters par Offset
; pour Quasar CPC numéro 1 (1992)
;
; Programme réalisé avec MAXAM,
; si vous avez DAMS les DS n
; doivent être remplacés par des DEFS n,0.

```

```
Org &5000
```

```

di          ; Disable Interrupt
ld hl,(&38) ; On sauve les vieilles
ld (inter),hl ; Interruptions
ld hl,&c9fb ; On met les nouvelles
ld (&38),hl ; EI et RET
ei          ; Enable Interrupt

Prog ld b,&f5 ; On sélectionne le Port B du PPI
Synch in a,(c) ; On pique son contenu
rra ; On teste si le Bit0=1
jp nc,synch ; Si bit0=0 alors on attend la fin
; du balayage
halt ; On attend que le balayage arrive
halt ; à peu près au milieu de l'écran
halt ; Patience...

```

```

halt ; Ouf, le dernier !
ds 20 ; On se calle en début de ligne...
ld b,&7f ; On sélection le Gate Array
ld hl,raster ; HL pointe sur la table de couleurs
Boucle ld a,(hl) ; On charge la couleur dans A
op 255 ; Si A=255 alors...
jp z,key ; On saute au test clavier
ld c,16 ; On sélectionne le border
out (c),c ; Et hop là !
out (c),a ; On le met à la couleur
ld c,0 ; On sélectionne l'encre 0
out (c),c ; Et hop là !
out (c),a ; On la met elle aussi à la couleur
inc hl ; On pointe sur la couleur suivante
ds 32 ; On attend la fin de la ligne...
jp boucle ; C'est reparti pour un tour !

```

```
; Test de la barre espace
```



```

Key ld bc,&f40e ; No comment ici,
out (c),c ; c'est pas les rasters...
ld bc,&f6c0
out (c),c
xor a
out (c),a
ld bc,&f792
out (c),c
ld bc,&f645
out (c),c
ld b,&f4
in a,(c)
ld bc,&f782
out (c),c
ld bc,&f600
out (c),c
rla
jp c,prog ; Bref, si pas espace alors
; on repart pour un tour...

di ; Disable Interrupt
ld hl,(inter) ; On récupère les anciennes
ld (&38),hl ; interruptions
ei ; Enable Interrupt
ret ; Retour à la case départ

```

```
; Les data...
```

```

Inter dw 0000
Raster db 20+64, 4+64, 21+64, 23+64, 31+64, 19+64, 11+64
db 3+64, 10+64, 14+64, 12+64, 28+64, 20+64
db 255

```

# Perfectionnement au Turbo Pascal

Comme je l'ai entendu dire autour de moi, ce qu'il vous manque en Turbo Pascal, c'est la manipulation des fichiers et des tableaux de variables. Donc, je vais de ce pas commencer par vous éclairer sur la gestion des fichiers en Turbo 3.

Je vais tout d'abord définir ce qu'est un fichier : c'est un ensemble de données de même type dont l'unité est une fiche. Bon, ça c'est le barratin classique, rien de bien compliqué. Donc, pour les contemporains du diplodocus, je vais encore affiner cette définition : un fichier existe, une fois ouvert, en mémoire (tampon ou buffer) et sur la disquette (si, si, je vous assure !).

Trêve de généralités simplistes. En Turbo, ils (les fichiers) s'utilisent au moyen d'un nom (de fichier) et d'une variable de type fichier (nom logique). Cette variable se déclarera à l'aide du type "FILE OF type" (où "type" est Integer, String, etc.) et l'instruction "ASSIGN(nom logique, nom physique)" permettra de la relier au fichier physique (sur la disquette). Je sais, je suis pas clair, mais c'est comme ça.

Maintenant que vous savez mettre en liaison la variable et son fichier, faudrait peut-être voir à ouvrir ce petit fichier. Eh bien, c'est très simple : deux instructions sont à votre disposition. D'abord, il y a "REWRITE(nom logique)" qui ouvre le fichier en écriture en l'initialisant totalement (c'est comme le OPENOUT du Basic). Ensuite, quelque chose de déjà plus intéressant, la commande "RESET(nom logique)". Alors, cette instruction, c'est ce que j'appellerais le luxe et ce, pour deux raisons : primo cette commande ouvre le fichier en lecture (comme OPENIN), deuzio elle l'ouvre aussi en écriture sans en effacer le contenu. Ainsi, si vous faites un RESET vous pourrez rajouter des données ou même changer seulement une fiche sans toucher au reste du fichier ! Il y a tout de même une restriction ; le fichier doit déjà exister sur la disquette, cette commande ne le créera pas si ce n'est pas le cas. Et, une dernière commande à connaître, le "CLOSE(nom logique)" qui ferme le fichier.

C'est bien beau de savoir configurer, ouvrir et fermer un fichier, mais vous aimeriez sans doute savoir y écrire et y lire non ? Eh bien c'est très simple.



Vous connaissez déjà les deux commandes qui réalisent ces opérations : "WRITE(nom logique, donnée)", c'est l'équivalent du PRINT#9 en Basic, et "READ(nom logique, variable)" qui réagira comme INPUT#9, variable.

Bon, je vous ai tout dit, du moins pour les notions de base sur les fichiers. Car leur utilisation en Turbo est beaucoup plus souple qu'en Basic. Pas loin de cet article, vous devriez trouver un petit programme d'exemple qui utilise les quelques commandes que je vous ai expliquées, plus quelques autres dont le fonctionnement y est établi en remarque.

La prochaine fois, soit on approfondira sur les fichiers, soit on attaquera les tableaux, tout dépendra de votre courrier... s'il y en a...

 Offset



```

Program Exemple;
Var fichier;File Of String 20 ;      (* Déclaration des variables *)
    numero:Integer;
    nom:String 20 ;

Procédure Liste;                      (* Procédure de listage *)
Var pointeur:Integer;
Begin
Writeln('Voici le contenu du fichier : ');
Seek(fichier,0);

(* Positionnement du pointeur sur la fiche 0 avec *)
(* Seek(nom logique, position) *)

While Not Eof(fichier) Do

(* Eof(nom logique) fonctionne comme le Eof du Basic, c'est à dire renvoie *)
(* VRAI si on est en fin de fichier et FAUX dans le cas contraire *)

    Begin
    pointeur:=Filepos(fichier);

(* La variable pointeur reçoit le numéro de la fiche courante *)
(* avec Filepos(nom logique) *)

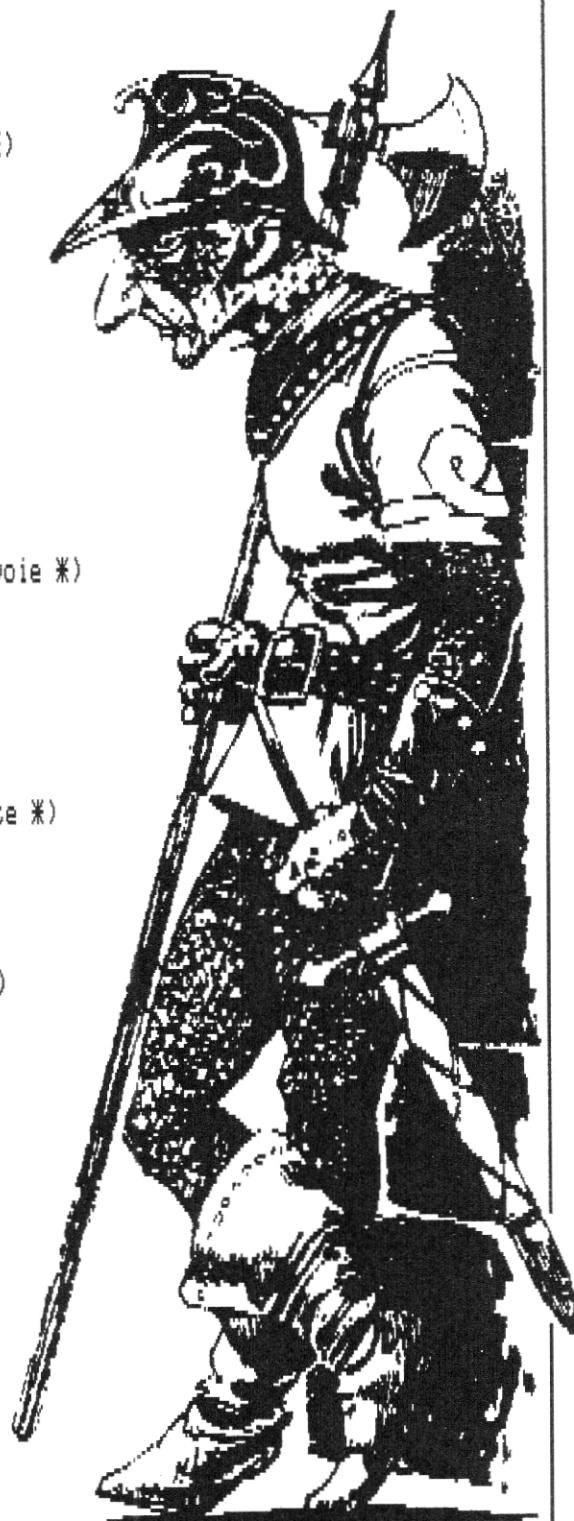
    Read(fichier,nom);                (* Lecture de la fiche courante *)
    Writeln('Fiche ',pointeur,' : ',nom);
    End;
End;

Begin
Assign(fichier,'NOMS.DAT');          (* Configuration du fichier *)
Clrscr;
Writeln('Création du fichier...');
Rewrite(fichier);                   (* création du fichier *)
Repeat
    Write('Entrez un nom (FIN pour arreter) : ');Readln(nom);
    If nom<Until nom='FIN';
Writeln('Fermeture du fichier...');
Close(fichier);
Clrscr;
Writeln('Ouverture du fichier avec RESET...');
Reset(fichier);                     (* Ouverture du fichier *)
Liste;
Write('Entrez un numéro de fiche à modifier : ');Readln(numero);
If numero
    Else Begin
        Seek(fichier,numero);
        Write('Entrez le nouveau nom : ');
        Readln(nom);
        Write(fichier,nom);
        End;

(* Filesize(fichier) renvoie le numéro de la dernière fiche du fichier *)
(* Seek vous connaissez, et Write aussi, pas de problème *)

Liste;
Writeln('Fermeture du fichier...');
Close(fichier);                     (* Fermeture du fichier *)
Writeln('Exemple fini. ');
End.

```



NOTA : Ce que j'ai dénommé fiche s'appelle plus généralement enregistrement.

