

REDACTION

AUBERT Patrick

Avec la collaboration de:

MICOL Cyril
et de BILLIE

AMSTRAD

MENSUEL
JUILLET 1989

- NUMERO 9 -

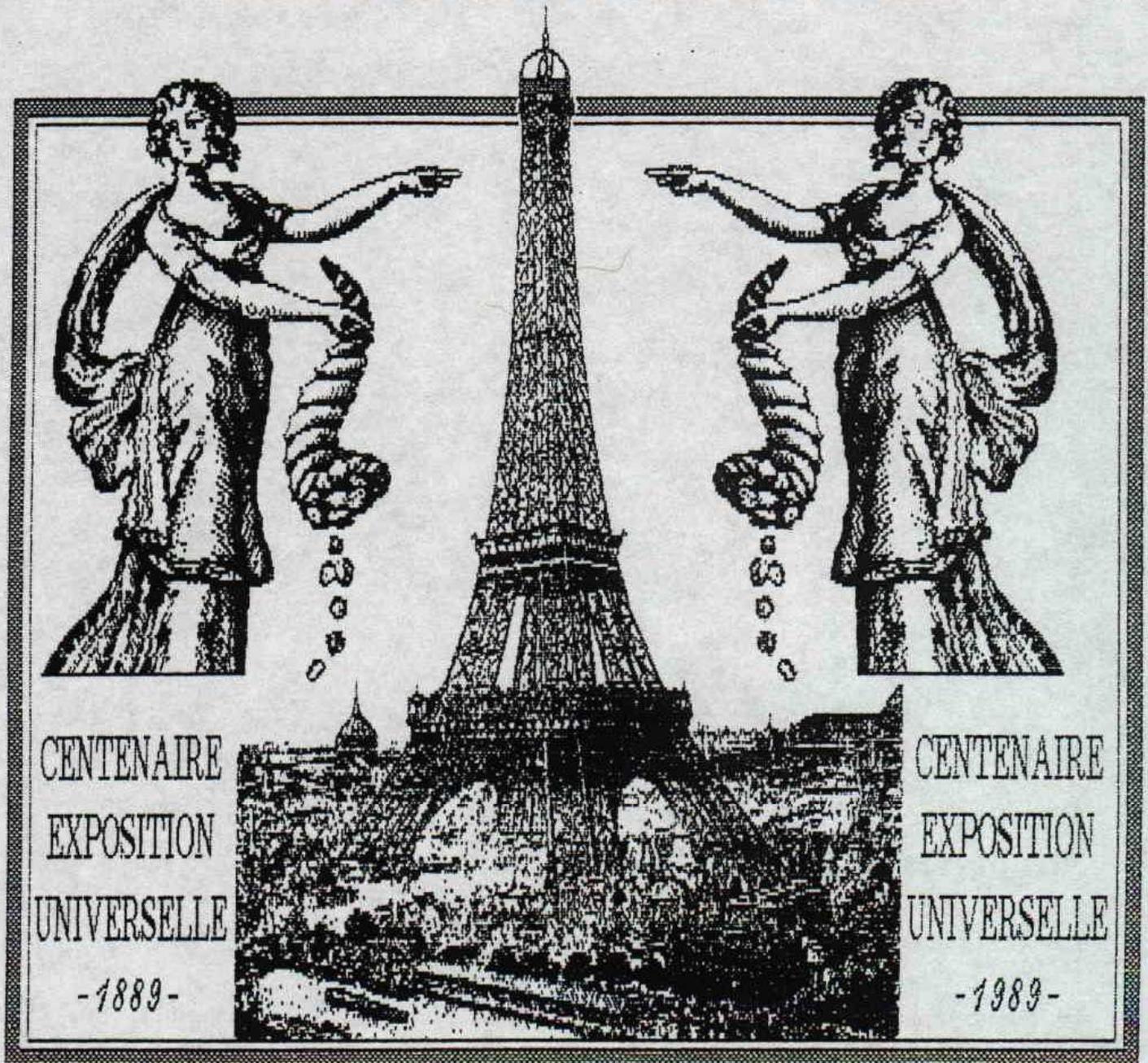
Siège : 58 rue de la Briquetterie — 17800 La Rochelle

TELEPHONE : 46.67.58.89

AMSTRAD CPC

9 PAGES ENTIEREMENT GRATUITES

AMSTRAD CPC



SOMMAIRE NUMERO 9

Page 1 Editorial - Publicité

Page 2 Logitest

Page 3 Routines

Page 4 Soundissimo

Page 5 Le Coin Pro

Page 6 Trucs & Astuces

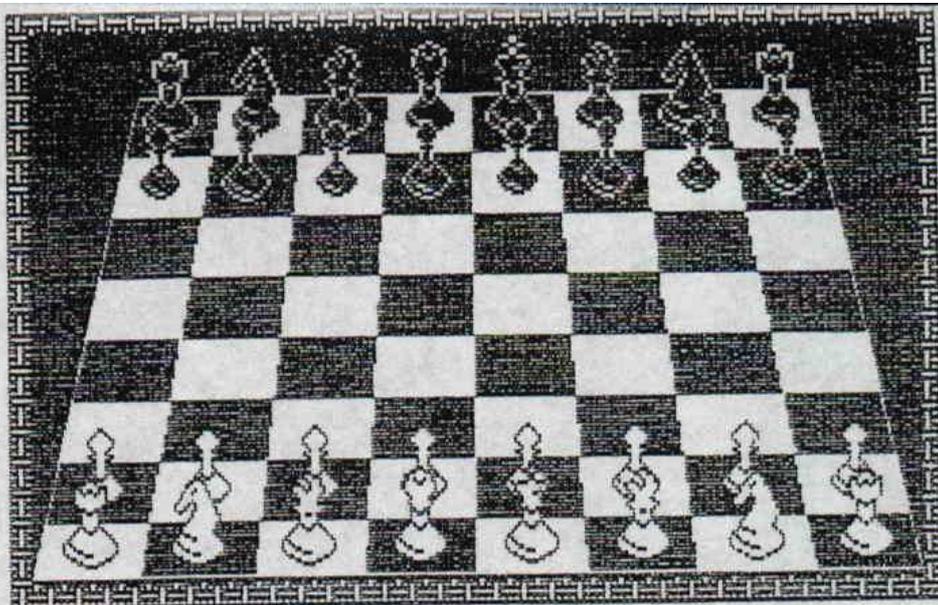
Page 7 Trucs & Astuces (suite)

Page 8 Jean Sedechozes

Page 9 Divers - Petites Annonces

LOGITEST

ECHEC et MAT TROIS JEUX D'ECHECS SUR CPC CYRUS II 3D VOICECHES MASTERCHES



Pour faire ce test, j'ai tout simplement fait jouer les trois jeux concernés les uns contre les autres. Pour cela, un deuxième CPC était nécessaire. J'ai donc organisé un mini tournoi consistant à opposer les différents programmes. Ce tournoi était organisé en trois manches, chaque programme affrontant les deux autres trois fois de suite. Seules les meilleures performances ont été retenues. Pas de grosse surprise à l'issue des tests. Voici donc les résultats, ainsi qu'un tableau des caractéristiques des trois jeux.

1er Tour

CYRUS II	NOMS	3D-VOICE
OUI en 58 coups	VAINQUEUR	non
1	NIVEAU	1
2' 08"	TEMPS	24' 12"

2ème Tour

CYRUS II	NOMS	MASTER
OUI en 34 coups	VAINQUEUR	non
1	NIVEAU	1
1' 23"	TEMPS	32' 14"

3ème Tour

3D VOICE	NOMS	MASTER
OUI en 79 coups	VAINQUEUR	non
1	NIVEAU	1
47' 10"	TEMPS	1H 7' 24"

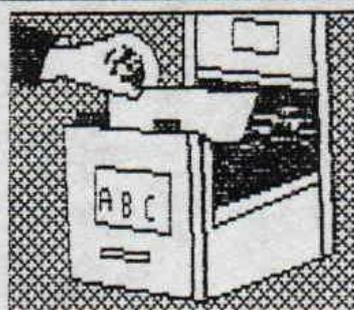
Caractéristiques	CYRUS II	3D VOICE	MASTER
Rapidité de jeu	Excellente 8/10	Assez lent 5/10	Lent 3/10
Graphisme	Le meilleur 9/10	Très bon 7/10	Passable 5/10
Déplacement des pièces sur l'échiquier	Par curseur avec touches fléchées	Coordonnées entrées au clavier (BIC3)	Coordonnées entrées au clavier (BIC3)
Signalisation de la mise en échec	Non	Par Bip sonore et affichage	Par Bip sonore et affichage
Pendules	Oui, deux	Non	Oui, deux
Affichage à l'écran des coups en cours de partie	Oui	Oui	Oui
Retour en arrière	Oui, jusqu'au début	Non	Non
Sauvegarde partie en cours	Oui, Disc et K7	Non	Oui, K7 (1)
Nombre de niveaux de jeu	12, dont un Tournoi	6	9
Résolution des problèmes	Oui	Oui	Oui
Type d'affichage échiquier	Deux type, 3d et à plat	3d	À plat seul
Joue ouverture automatique	Oui, sur 8 coups	Non	Non
Conseil du coup à jouer	Oui	Oui	Oui
Echange les camps en cours	Oui	Oui	Oui
Joue seul contre lui même	Oui	Oui	Non
Imprimante possible	Oui, 3 variantes	Non	Non
Force	Le débutant n'a aucune chance	Moyen	À conseiller aux débutants

(1) Le logiciel d'origine sauvegarde les parties en cours, uniquement sur cassettes. Dans le numéro 8 de MICRO STRAD, page 48, une routine permet de modifier ce type de sauvegarde pour l'utiliser avec des disquettes. Cette routine sera donnée dans le prochain numéro (10) de Runstrad.

ROUTINES

TRI SHELL-METZNER

L'ensemble des données est successivement décomposé en sous-ensembles de plus en plus petits (divisé par deux à chaque itération). Un élément d'un sous-ensemble est alors comparé à son correspondant du sous ensemble suivant, avec échange, s'il y a lieu et, dans ce cas, on remonte au sous ensemble précédent pour remise en ordre éventuelle et ainsi de suite jusqu'au premier sous-ensemble. Le travail est terminé lorsque le sous-ensemble ne comporte plus qu'un élément. Par cette méthode, le N^e élément d'un sous-ensemble est toujours inférieur N^e élément du sous-ensemble suivant. Pour utiliser



LE TRI

cet algorithme pour faire un tri, il suffit de ranger les deux premiers éléments. On insère ensuite chacun des éléments de la suite en appliquant la procédure décrite.

Cette méthode est évidemment plus efficace que la précédente, mais elle reste largement insuffisante pour des fichiers en dessous d'une centaine de fiches.

```
10 REM TRI PAR SELECTION
20 REM DU TABLEAU D$
30 FOR I=1 TO M-1
40 FOR J=I+1 TO M-1
50 IF D$(J)<D$(I) THEN AA$=D$
  $(J);D$(J)=D$(I);D$(I)=AA$
60 NEXT J
70 NEXT I
80 REM RETURN SI VOUS
  ENTREZ PAR GOSUB
90 REM DANS CE SOUS
  PROGRAMME
```

```
10 REM TRI SHELL-METZNER
20 REM DU TABLEAU D$
30 P=M
40 P=INT(P/2)
50 IF P<1 THEN RETURN:REM TRI
  TERMINE
60 DEB=1;FIN=M-P
70 R=DEB
80 C=R+P:IF D$(R)<D$(C) GOTO
  100
90 AA$=D$(R);D$(R)=D$(C);D$(C)
  =AA$:R=R-P:IF R>0 GOTO 80
100 DEB=DEB+1:IF DEB>FIN
  THEN 40 ELSE GOTO 70
```



```
10 REM TRI PAR DICHOTOMIE
20 REM DU TABLEAU D$
30 IF D$(1)>D$(2) THEN AA$=
  D$(2);D$(2)=D$(1);D$(1)=
  AA$:GOTO 40
40 FOR N=3 TO M
50 R=0:D=N;G=0;Z=D-G:C=1
60 E=INT(Z/2)
70 R=R+E*C
80 IF D$(N)<D$(R) THEN D=R;
  C=-1;GOTO 100
90 G=R:C=1
100 Z=D-G:IF Z>1 THEN 60
110 IF C=1 THEN R=R+1
120 AA$=D$(N)
130 FOR I=N TO 1 STEP -1
140 IF R=I THEN D$(I)=AA$:
  GOTO 160
150 D$(I)=D$(I-1):NEXT I
160 NEXT N
170 REM RETURN
```

COMMENTAIRES :
Il est possible de trier des variables NUMERIQUES entières et décimales. Il suffit pour cela de définir les tableau. Exp:

```
DIM H(1200) ou DIM HX(1200)
  décimales          entières
```

Une autre solution consiste à définir les types des variables en début de programme. Exp:
DEFINT H (entièrè)
DEFREAL H (décimale ou réelle)
Si vous voulez tester ces programmes, sachez que la variable M utilisée dans les routines représente le nombre de fiches que vous dimensionnez dans vos programmes.

Pour trier les données stockées dans un fichier, il est nécessaire de charger celles-ci en mémoire (dans un tableau), d'en faire le tri suivant un ou plusieurs critères, puis de réenregistrer le tableau dans le fichier.

Il existe plusieurs méthodes de tri (par sélection, par échange, par dichotomie, etc...), leur efficacité reposant sur la longueur et la complexité du programme et surtout sur le temps d'exécution. Les moins efficaces nécessitent un temps proportionnel au carré du nombre d'éléments à trier (N * N) les plus rapides à N * log(N).

Dans cet article, nous allons voir trois méthodes de tri complètement différentes. Vous pourrez comparer les performances: Le tri par sélection. Le tri par dichotomie. Le tri par la méthode de Shell-Metzner.

D'après les résultats sur les trois échantillons de nos exemples, le tri par dichotomie dont on dit le plus grand bien, n'est pas d'une grande efficacité lorsque le fichier s'agrandit et il est alors préférable de choisir la méthode de Shell-Metzner.

TRI par SELECTION

Dans cette méthode, on prend un élément de référence. On le compare successivement à chacun des suivants. Si l'élément de référence est plus petit, c'est bon. Sinon, on l'échange avec le suivant. Arrivé au dernier suivant, élément de référence contient le plus petit élément du tableau. On prend alors son suivant comme élément de référence, et ce, jusqu'au dernier du tableau.

Cette méthode, facile à programmer, ne peut être utilisée que pour de très petits fichiers (100 à 150 fiches maximum).

TRI par DICHOTOMIE

Le principe est le suivant: On considère que N éléments sont rangés dans l'ordre voulu (croissant pour l'exemple). On souhaite insérer un nouvel élément. Pour cela, nous prenons l'élément du milieu et nous le comparons à l'élément nouveau. Si le nouvel élément est plus petit, sa place sera dans le sous-ensemble gauche, sinon dans le sous-ensemble droit. On recommence l'algorithme jusqu'à ce que la gamme couverte par le sous-ensemble corresponde à 1. Nous connaissons alors la place à laquelle il faut ranger le nouvel élément. Nous devons alors décaler tous les éléments qui suivent, d'une position, et ranger dans la place libérée le nouvel élément.

TABEAU COMPARTATIF	Nbre d'éléments	30	150	300
	Sélection	3,9 s	79 s	295 s
	Dichotomie	3,9 s	55 s	194 s
	Shell-metzner	1,7 s	19,8 s	33,8 s

SOUND'S / MOSOUND'S

Et voici un petit morceau intitulé "Computer Disco". Ce programme nous a été envoyé par Nicol Cyril

```

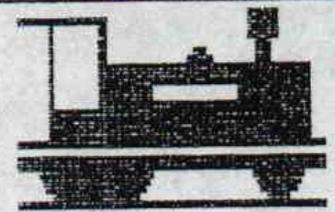
1 BORDER 26:MODE 0:CLG 15
2 DATA 1,1,3,4,5,4,3,1,0,1,3,4,5,4,3,0
3 DATA 1,1,3,4,5,4,3,1,0,1,3,5,4,0,3,0
4 DIM n(32):RESTORE 2:FOR i=1 TO 32:READ
n(i):NEXT i
5 DATA 179,159,142,127,119,106,95,89,80
6 DIM g(10):RESTORE 5:FOR i=1 TO 9:READ
g(i):NEXT i
7 DATA 5,3,1,2,5,3,1,2
8 DIM t(16):RESTORE 7:FOR i=1 TO 8:READ
t(i):NEXT i
9 DATA 1,2,3,1,1,2,3,2,1,2,3,1,1,2,3
10 DATA 3,1,2,3,1,1,2,3,2,1,2,3,1,1,4,1,4
11 DIM b(32):RESTORE 9:FOR i=1 TO 32:REA
D b(i):NEXT i
12 DATA 1911,30,0,1,60,10,0,5
13 DIM h(4),v(4):RESTORE 12:FOR i=1 TO 4:
READ h(i),v(i):NEXT i
14 ENV 1,1,15,1,15,-1,2:ENV 2,8,-2,1:ENV 3,
1,15,1,15,-1,4
15 ENV 4,3,5,1,15,-1,1:ENV 5,1,13,1,3,-1,2,1
0,-1,8
16 ENV 6,1,15,1,3,0,1,15,-1,12
17 TEM=18
18 FOR i=1 TO 9:SOUND 7,g(i),5,6:NEXT i
19 FOR i=1 TO 32:SOUND 4,h(b(i)),tem,0,b(i)
,,v(b(i))
20 SOUND 3,g(t(1)),tem/2,4:SOUND 3,0,tem/
2,0:NEXT i
21 CLG 14
22 FOR i=1 TO 8
23 a=g(t(i))
24 IF a=179 THEN g(4)=134 ELSE g(4)=127
25 a=a*2
26 FOR j=1 TO 32
27 SOUND 34,a,tem/2,0,5:SOUND 20,h(b(j)),
tem,0,b(j),v(b(j))
28 SOUND 1,g(t(i)+n(j)-1),tem,0,6*SGN(n(j)
)):SOUND 2,a/2,tem/2,0,5
29 NEXT j
30 IF <>1 THEN 34
31 FOR i=1 TO 8
32 SOUND 1,g(t(1)+n(i)-1),tem,0,6*SGN(n(i)
)
33 NEXT i:END
34 FOR i=1 TO 16:SOUND 1,g(t(1)+n(i)-1),te
m,0,6*SGN(n(i))
35 SOUND 6,0,tem,0:NEXT i
36 FOR i=1 TO 16:SOUND 1,g(t(1)+n(i)-1),te
m,0,6*SGN(n(i))
37 SOUND 6,0,tem,0:NEXT i
38 FOR j=1 TO 2:FOR i=1 TO 32:SOUND 4,h(
b(i)),tem,0,b(i),v(b(i))

```

```

39 SOUND 3,0,tem,0:NEXT i,j
40 RESTORE 48:FOR i=1 TO
63:READ t,d:SOUND 3,h(t),d
*tem,0,t,,v(t)
41 SOUND 4,0,d*tem,0:NEXT
42 FOR i=1 TO 32:SOUND 4,h
(h(i)),tem,0,b(i),v(b(i))
43 SOUND 3,0,tem,0:NEXT i
44 FOR i=1 TO 16:SOUND 1,g
(t(1)+n(i)-1),tem,0,6*SGN(n
(i))
45 SOUND 4,g(t(1)+n(i)-1)/
2,tem,0,6*SGN(n(i)):SOUND 2
,0,tem,0:NEXT
46 IF z=0 THEN z=1:GOTO 22
47 END
48 DATA 1,3,1,1,1,3,1,1,1,3,1,1
49 DATA 1,1,3,1,1,1,3,1
50 DATA 1,3,1,1,1,3,1,1,1,3,1,1
51 DATA 1,1,3,1,1,1,3,1
52 DATA 1,3,1,1,1,3,1,1,1,3,1,1
53 DATA 1,1,3,1,1,1,3,1
54 DATA 1,3,1,1,3,4,1,3,1,1,3
,4,1,3,1
55 DATA 1,3,4,1,3,1,1,3,1,3,1
,3,1,3,1
56 DATA 1,1,5,2,0,5,3,1,1,1,1
,1,5,2,0,5,3,1,1,1
57 DATA 1,0,5,1,1,1,0,5,1,1,1
,0,5,2,0,5,2,1,2,1,2,1,5,4,0,5

```



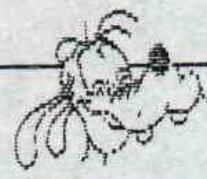
Bruit du train.....

```

10 ENV 1,3,5,1,1,0,1,3,-5,1:ENV 2,15,1,1,1,
0,50,15,-1,1:ENT 2,14,-1,1,1,0,50,14,1,
1
20 FOR A=1 TO 2:SOUND 1,67,0,0,2,2:SOUND
2,56,0,0,2:FOR T=1 TO 800:NEXT T,A
30 FOR I=1 TO 10
40 SOUND 1,0,0,0,1,0,10:FOR T=1 TO 50:
NEXT:SOUND 1,0,0,0,1,0,I+5
50 FOR T=1 TO 100:NEXT T,I
60 FOR I=1 TO 20:SOUND 1,0,0,0,1,0,10
70 FOR T=1 TO 100-I:NEXT T,I
80 FOR I=1 TO 20:SOUND 1,0,0,0,1,0,15:SO
UND 1,0,0,0,1,0,10:NEXT
90 FOR I=1 TO 500:NEXT:GOTO 10
100 END=

```

Billie



Electra 2000

Gestion de Fichiers-Agenda-Calendar
 Calculatrice-Carte de France et des
 Départements-Tous les utilitaires
 présents à l'écran en permanence
 Aide Constante à l'écran-Gestion des
 Erreurs-Manuel d'utilisation 20 pages
 TRI-MASQUES-RECHERCHE MULTICRITERE
 FICHIERS de 128 ko PAR FACE DE DISC
 DISQUETTE UNIQUEMENT-CPC 6128
 PRIX 100.00 fr ou 70.00 si envoi disc

OFFRE D'EMPLOI

Ce petit journal vous a intéressé,
 vous pourriez peut être
 collaborer, quelque soit votre
 domaine, graphisme, programmation
 Prenez contact avec la rédaction

LE COIN PRO

BANKMAN ? C'est pas compliqué ? Regardez...

Certains d'entre vous qui ont achetés un CPC 6128, et qui n'avaient pas les connaissances informatiques nécessaires, ont déchantés lorsqu'ils ont compris (ou pas compris) pourquoi il n'était pas possible, malgré les 128k de mémoire RAM (Random Access Memory) de créer des programmes d'une occupation supérieur à 42K (théorique). Ils ont fait la découverte de Bankmanager, vendu avec le CPC et qui se trouve sur l'une des disquettes système C/PM plus et qui permet l'accès aux 64k supplémentaires. Certains ont retenus qu'il était possible avec Bankman de travailler sur 5 écrans différents en même temps. Cette possibilité, fort intéressante, est cependant réservée pour des applications assez spécifiques. Nous allons donc la laisser de côté pour étudier l'autre possibilité de Bankman, et qui permet de mettre dans ces 64k, des fichiers de données alphanumériques. Nous avons vu dans le numéro 8 page 8, ce qu'était une donnée alphanumérique. Pour appréhender parfaitement toutes les nouvelles possibilités, ainsi que les nouvelles commandes disponibles, nous allons écrire ensemble un programme de gestion de fichiers qui sera d'une simplicité remarquable, écrit en basic mais néanmoins performant. Rien ne vous empêche de l'habiller selon vos goûts et vos moyens. Pour notre exemple, il nous faudra prendre plusieurs décisions avant de nous lancer dans la programmation proprement dite. Voyons cela. Que fera le programme ? Il va tout simplement gérer un répertoire d'adresses et de téléphones. Notez que vous pouvez de votre côté choisir autre chose, les adaptations seront faciles. Quelles sont les rubriques qui vont composer notre création ? Disons, 1 - Le nom, 2 - Le(s) prénom(s), 3 - L'adresse, 4 - Le code postal, 5 - La ville, 6 - Le numéro de téléphone. Maintenant, regardez le numéro 3 du journal. Nous allons travailler avec bankman de la même façon qu'en accès direct. Nous pouvons considérer nos 64k comme une disquette virtuelle. Il nous faut réserver la place des rubriques qui vont composer une fiche. Après cette opération, nous saurons exactement le nombre de fiches qu'il sera possible d'entrer dans le fichier. Retenez cependant une chose importante. Si notre fichier est limité par rapport à un fichier similaire en accès direct (presque 3 fois plus petit) sa rapidité en mode recherche est 250 fois supérieur. Un petit inconvénient toutefois, puisque le fichier est chargé en entier dans la mémoire à chaque nouvelle utilisation, vous perdez selon sa taille de 3 à 25 secondes, ce qui est négligeable en comparaison du temps gagné lors des recherches.

```
Le listing sous Bankman
10 REM GESTION FICHER SOUS BANKMAN
20 REM N'OUBLIEZ PAS DE LANCER LE
30 REM PROGRAMME BANKMAN AVANT TOUT
40 REM DEBUT
50 OPENOUT"XXX":MEMORY HIMEM-1:CLOSEOUT
60 DIM RUB$(6):'6 RUBRIQUES
70 DATA NOM,PRENOMS,ADRESSE,CODE-POSTAL,VILLE,TELE
  PHONE
80 REM LECTURE DATAS
90 FOR I=1 TO 6:READ RUB$(I):NEXT
100 L0%=101:'LONGUEUR D'UNE FICHE
110 REM INITIALISATION DES RUBRIQUES
120 R1$=STRING$(20,32):'NOM 20 CARACT.
130 R2$=STRING$(15,32):'PRENOMS 15 CARACT.
140 R3$=STRING$(40,32):'ADRESSE 35 CARACT.
150 R4$=STRING$(5,32):'C-P 5 CARACT.
160 R5$=STRING$(20,32):'VILLE 15 CARACT.
170 R6$=STRING$(11,32):'TEL 11 CARACT.
180 R%=0:'VARIABLE SYSTEME. INITIALISE
190 FIN%=648:'NOMBRE MAXI DE FICHES
200 BANKOPEN,101
210 REM MENU
220 MODE 2:'DEFINISEZ LES ENCREES VOUS-MEME
230 LOCATE 35,2:PRINT"MENU"
240 LOCATE 5,6:PRINT"1 - LECTURE FICHER"
250 LOCATE 5,8:PRINT"2 - ECRITURE FICHER"
260 LOCATE 5,10:PRINT"3 - SAISIE DES FICHES"
270 LOCATE 5,12:PRINT"4 - RECHERCHE DE FICHES"
280 LOCATE 5,14:PRINT"5 - SUPPRIMER UNE FICHE"
290 LOCATE 5,16:PRINT"6 - CATALOGUE DISC"
300 LOCATE 5,18:PRINT"7 - FIN TRAVAIL"
310 LOCATE 5,22:PRINT"ENTREZ CHOIX (RETURN)"
320 CLEAR INPUT:'VIDE LE TAMPON CLAVIER
330 I$=INKEY$:IF I$="" THEN 330
340 I=VAL(I$):IF I<1 OR I>7 THEN 330
350 ON I GOSUB 1000,2000,3000,4000,5000,6000,7000
360 GOTO 220
```

Ca sera tout pour le moment: nous allons voir certaines instructions spéciales à Bankman.

LIGNE 101 - L0% contient la somme des caractères de nos 6 rubriques.

LIGNE 110 à 170 - Initialise les variables réservées aux rubriques avec un nombre d'espace égal aux nombre de caractères que devront recevoir les variables.

LIGNE 180 - R% est initialisé à zéro. Ceci est indispensable car Bankman va utiliser cette variable entière pour retourner certaines valeurs qui servent à contrôler le résultat de certaines opérations comme par exemple lors de recherches dans les banks, si la recherche est négative, R% sera à -1, si la recherche est positive, R% recevra le No de la fiche contenant les renseignements recherchés.

Rendez-vous pour la suite dans le No 10.

TRUCS ASTUCES

La couleur sur CPC vous connaissez ???
Avec ce Truc vous allez voir 36 chandelles
8 en Mode 2, 16 en Mode 1, 27 en Mode 0

Ce mois-ci nous allons voir un truc qui nous permettra d'écrire en MODE 2 avec 8 couleurs, en MODE 1 avec 16 couleurs et en MODE 0 avec 27 couleurs, soit la TOTALITE des couleurs du CPC. La routine a été écrite sur un 6128 et l'auteur ne dit pas si la compatibilité avec un 464 ou un 664 est assurée.

Ce truc est réalisé pour le mode texte, il y a de forte chances pour que cela ne marche pas correctement en mode graphique. La routine utilise le système des RSX, aussi lors de l'utilisation de celles-ci n'oubliez pas la barre !



LE PRINCIPE

L'écran est divisé en 4 zones, A, B, C, D. Vous définissez vous même la couleur de chaque zone, sachant que, en mode 2 vous disposez de 4 zones et que tous ce que vous allez écrire dans la zone 1, le sera avec la couleur que vous avez sélectionné pour cette zone. Il est normal de constater que si vous définissez une encre bleue par exemple, pour les zones A et B, et que vous faites la même chose avec une encre vert sur les zones C et D, vous aurez alors l'impression que votre écran ne comporte plus que deux zones, la première en bleu et la deuxième en vert. La bordure de l'écran est elle aussi divisée en quatre zone selon le même principe que l'écran, ce qui vous donne la possibilité d'obtenir des effets spéciaux fantastiques.



LA TECHNIQUE

Le truc se divise en deux programmes. Le premier est chargé de créer les RSX ainsi que les routines binaires nécessaires. Pour des raisons de mises en pages acrobatiques, au cours desquelles je me suis foulé le poignet gauche et passé deux ongles, ce programme se trouve sur la page suivante. Le deuxième programme (sur cette page) est une demo, qui a pour but, mais vous vous en doutez car vous n'êtes pas tombés de la dernière pluie, de vous montrer les capacités du truc en question.

Les RSX et le mode d'emploi

Les RSX sont les suivantes:
 ION : Active le système multicolore
 IOFF : L'inverse de ION
 IMODE,x: Même utilisation que MODE sous Basic
 IBORDER,a,b,c,d
 Change la couleur de la bordure de l'écran. Les variables a,b,c,d peuvent être des constantes. Les valeurs sont données dans le tableau A.
 IINK,x,a,b,c,d :
 Change la couleur des zones de l'écran. Mêmes remarques que précédemment pour a,b,c,d. La valeur de x, qui peut être soit 1, soit 0, agit comme un interrupteur, sur 1, les messages sont visibles, sur 0, les messages sont de la même couleur que l'ancre du fond.
 Tapez le programme "Mode multicolore No2" de la page suivante. Le programme test la valeur des datas et si vous avez fait une erreur, corrigez-la et relancez, non sans avoir au préalable, mis une disquette dans le lecteur, car le programme sauvegarde un fichier binaire sous le nom de "MULTCOLO.BIN". Pour exécuter le programme il suffit de le lancer par
 LOAD"MULTCOLO.BIN",&A000 : CALL &A000

```
10 REM DEMO
20 MEMORY &9FFF
30 LOAD"MULTCOLO",&A000
40 CALL &A000
50 ION:IMODE,0
60 IBORDER,4,28,31,14
70 IINK,1,7,11,5,29
80 FOR i=1 TO 25
90 PRINT STRING$(20,65);
100 NEXT i:PRINT CHR$(7);
110 CALL &BB18:IMODE,1
111 IBORDER,28,11,15,4
112 IINK,1,2,12,9,25
120 FOR i=1 TO 25
130 PRINT STRING$(40,66);
140 NEXT i:PRINT CHR$(7);
150 CALL &BB18:IMODE,2
151 IBORDER,14,16,22,8
152 IINK,1,30,20,10,1
160 FOR i=1 TO 25
170 PRINT STRING$(80,67);
```

```
180 NEXT i:PRINT CHR$(7);
190 FOR i=1 TO 100
200 a=RND*31:b=RND*31
210 c=RND*31:d=RND*31
220 IBORDER,a,b,c,d
230 IINK,1,d,a,c,b
240 NEXT i
250 CALL &BB18:PRINT
CHR$(7);
260 IINK,0,d,a,c,b
270 CALL &BB18:PRINT
CHR$(7);
280 IINK,1,d,a,c,b
290 IOFF:IMODE 2:END
```



TRUCS & ASTUCES Suite et fin.

TABLEAU A Correspondance Couleurs

LA ROUTINE MULTICOULEUR N'UTILISE PAS LES VALEURS NORMALES POUR CODER LES COULEURS, AUSSI LE TABLEAU -A- VOUS DONNE LA CORRESPONDANCE.

A SAVOIR: NE PAS OUBLIER DE FAIRE ION POUR ACTIVER LA ROUTINE, ET IOFF POUR LA DESACTIVER, IMODE,x SE COMPORTE DE LA MEME FACON QUE SON HOMOLOGUE MODE DE BASIC.

POUR INTEGRER LA ROUTINE A UN PROGRAMME BASIC DE VOTRE CRU MEMORY &9FFF: LOAD"MULTICOLO.BIN",&A000: CALL &A000

LE CALL &A000 INITIALISE LES RSX.

Ce truc nous est communiqué par un lecteur. Il s'agit du jeune MARIEN E. Merol à lui...

NORMAL	ROUT	NORMAL	ROUT	NORMAL	ROUT
0	20	9	22	18	18
1	4	10	6	19	2
2	21	11	23	20	19
3	28	12	30	21	26
4	24	13	0	22	25
5	29	14	31	23	3
6	12	15	14	24	11
7	5	16	7	25	3
8	13	17	15	26	11



```

1  *** MODE MULTICOULEURS no2 ***
2  * COMMUNIQUE PAR E.MARIEN *
3  *POUR RUNSTRAD LE:5/5/89 C.C *
4  *****
5  MEMORY &9FFF:a=&A000
6  FOR e=0 TO 70
7  FOR b=0 TO 7
8  READ n:POKE (a+b),n:c=c+n
9  NEXT:READ d:IF c=d THEN 11
10 PRINT CHR$(7);"ERREUR DATA ligne";e+13:END
11 a=a+8:c=0:NEXT
12 SAVE"MULTICOLO.BIN",b,&A000,&238
13 DATA 1,10,160,33,47,162,205,209,827
14 DATA 188,201,27,160,195,83,160,195,1209
15 DATA 126,160,195,70,160,195,50,160,1116
16 DATA 195,168,160,77,79,68,197,73,1017
17 DATA 78,203,79,70,198,79,206,66,979
18 DATA 79,82,68,69,210,0,0,0,508
19 DATA 0,0,58,250,161,167,192,61,889
20 DATA 50,250,161,33,251,161,6,129,1041
21 DATA 17,194,160,195,215,188,33,251,1253
22 DATA 161,175,50,250,161,205,221,188,1411
23 DATA 195,2,188,254,1,192,205,70,1107
24 DATA 160,221,126,0,205,14,188,205,1119
25 DATA 50,160,17,4,4,237,83,19,574
26 DATA 162,237,83,21,162,17,10,10,702
27 DATA 237,83,23,162,237,83,25,162,1012
28 DATA 62,1,205,144,187,201,254,5,1059
29 DATA 192,221,126,8,33,19,162,254,1015
30 DATA 0,40,9,221,70,8,17,4,369
31 DATA 0,25,16,253,221,126,6,119,766
32 DATA 35,221,126,4,119,35,221,126,887
33 DATA 2,119,35,221,126,0,119,201,823
34 DATA 254,4,192,33,14,162,221,126,1006
35 DATA 5,119,35,221,126,4,119,35,665
36 DATA 221,126,2,119,35,221,126,0,850
37 DATA 119,201,243,221,229,229,245,217,1704
38 DATA 197,221,33,14,162,14,0,203,844
39 DATA 225,237,73,221,78,0,203,241,1278
40 DATA 237,73,237,73,221,33,19,162,1055
41 DATA 175,79,237,73,221,78,0,203,1066
42 DATA 241,237,73,221,35,221,35,221,1284

```

```

43 DATA 35,221,35,60,254,6,32,233,876
44 DATA 193,217,62,2,50,13,162,33,732
45 DATA 4,162,6,129,17,15,161,205,699
46 DATA 224,188,241,225,221,225,201,243,1768
47 DATA 229,221,229,245,33,13,162,53,1185
48 DATA 40,6,241,221,225,225,251,201,1410
49 DATA 217,197,221,33,15,162,14,0,859
50 DATA 203,225,237,73,221,78,0,203,1240
51 DATA 241,237,73,237,73,221,33,19,1154
52 DATA 162,175,79,237,73,221,78,1,1026
53 DATA 203,241,237,73,221,35,221,35,1266
54 DATA 221,35,221,35,60,254,6,32,864
55 DATA 233,193,217,33,4,162,205,230,1277
56 DATA 188,33,4,162,6,129,17,105,644
57 DATA 161,205,224,188,241,221,225,225,1690
58 DATA 201,243,221,229,229,245,217,197,1782
59 DATA 221,33,16,162,14,0,203,225,374
60 DATA 237,73,221,78,0,203,241,237,1290
61 DATA 73,237,73,221,33,19,162,175,993
62 DATA 79,237,73,221,78,2,203,241,1124
63 DATA 237,73,221,35,221,35,221,35,1078
64 DATA 221,35,60,254,6,32,233,193,1034
65 DATA 217,33,4,162,205,230,188,33,1072
66 DATA 4,162,6,129,17,193,161,205,367
67 DATA 224,188,241,225,221,225,201,243,1768
68 DATA 229,221,229,245,217,197,221,33,1592
69 DATA 17,162,14,0,203,225,237,73,931
70 DATA 221,78,0,203,241,237,73,237,1290
71 DATA 73,221,33,19,162,175,79,257,999
72 DATA 73,221,78,3,203,241,237,73,1129
73 DATA 221,35,221,35,221,35,221,35,1024
74 DATA 60,254,6,32,233,193,217,33,1028
75 DATA 4,162,205,230,188,241,221,225,1476
76 DATA 225,201,0,0,0,0,0,0,426
77 DATA 129,194,160,255,0,0,241,0,979
78 DATA 0,129,183,161,82,0,11,12,578
79 DATA 13,14,0,4,4,4,4,10,53
80 DATA 10,10,10,0,0,0,0,0,30
81 DATA 0,0,0,0,0,0,0,0,0
82 DATA 0,0,0,0,0,0,0,252,252
83 DATA 166,12,162,255,255,255,255,1611

```



Les cours du Prof Jean SEDECHOZES

Salut, je suis le professeur Jean SEDECHOZES, Maître de conférence à l'Institut du bidouillage, Directeur du Centre d'Initiation Appliqué, Ancien Interne de la Faculté des T.E.T.G. (Trucs En Tous Genres), Chef de Recherche du Congrès ASC II, Diplômé des Hautes Ecoles de la Disquette Formatée, Prix Nobel de l'AMSDOS.

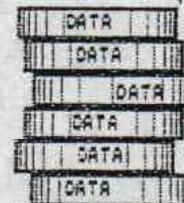
BASIC et ses DATAS

Des DATAS pourquoi faire ?... Ce n'est pas très compliqué! Les DATAS sont utilisées pour introduire au sein d'un programme une certaine quantité de données constantes, qui peuvent être, soit Numériques, (des TVA ou des coordonnées x,y pour un affichage par LOCATE), soit Alphanumériques, (les noms de vos amis, les mois de l'année). En général, les datas sont utilisées lorsque le nombre de données est important. Dans ce cas, l'utilisation des datas fait gagner de la place en octet. Lorsque qu'un programme est lancé, il s'exécute en séquence, lignes par lignes, (sauf dans le cas d'un Goto ou d'un Gosub), lorsque le programme arrive sur une ligne commençant par DATA (exp: 1200 DATA 44,45,47), il ignore totalement les données qui suivent la data en question, il se contente de regarder si après la dernière données, il n'y aurait pas le symbol de séparation d'instructions (les 2 points :) car si oui, cela signifierait qu'il y a une instruction à exécuter, dans le cas contraire, il passe à la ligne suivante.

Comme pour toutes les instructions ou les commandes de Basic, l'instruction DATA doit respecter certaines règles. La syntaxe d'abord, peu importe que le mot data soit écrit en majuscules ou en minuscules car l'interpréteur transforme les mots clés en majuscules automatiquement. Ne faites pas comme quelqu'un qui, lorsqu'il avait plusieurs données à mettre sur une ligne de data, écrivait le mot data au pluriel (DATAS) vous imaginez d'où les Syntax Error. Il est possible, et même fortement recommandé, lorsque vos données sont nombreuses, d'en mettre plusieurs à la suite sur une même ligne de data. Ceci économise de la place en mémoire, (moins de lignes à taper). Si vous procédez de cette façon, il y aura alors certaines règles de syntaxe à respecter. Les données, s'il y en a plusieurs sur une même ligne, doivent être séparées par une VIRGULE. (exp: DATA 1,3,5,7), à l'exception de la première et de la dernière. Il ne faut rien mettre entre le mot data et la première donnée. Rien non plus après la dernière. Même s'il est possible de mettre une instruction derrière une ligne de data, ceci n'est pas conseillé. Les GUILLEMETS. Vous avez certainement remarqué que certains programmeurs mettent des guillemets autour de certaines données, et d'autres, non. Une explication s'impose tout particulièrement, car il s'agit là d'une possibilité du Basic Locomotive. Je ne connais pas tous les Basics existants (il y en a autant qu'il existe de marque d'ordinateurs), mais je peux vous dire que seul à ma

Je sais pas trop...
Regarde ligne 6128?

Tiens, encore
des DATAS. Où je
les pose ?



connaissance, le Basic Locomotive permet ce genre de truc. De quoi s'agit-t'il? Et bien, normalement, lorsque vous écrivez une donnée ALPHANUMERIQUE sur une ligne de data, la logique voudrait que cette donnée soit, comme c'est le cas lors de son attribution à une variable, entourées de guillemets (exp: A\$="Janvier"). Or sur les CPC, cette syntaxe n'est pas obligatoire et vous n'êtes pas obligés d'entourer une donnée alpha de guillemets. Oui mais alors comment le programme peut faire la différence entre une donnée Numérique, et une donnée Alphanumérique. Et bien c'est ce que nous verrons en détail dans le numéro 10. A bientôt...

```
10 MODE 2:CAT:INPUT"Nom du Fichier: ",no$
20 CLS:LINE INPUT"chaîne recherchée: ",a$
30 OPENIN no$
40 WHILE NOT EOF
50 LINE INPUT#9,c$
60 IF INSTR(c$,a$)<>0 THEN PRINT c$
70 WEND
80 CLOSEIN .
```

Et voici à la demande de J.P et de Marc, un petit programme chargé de retrouver une chaîne de caractères dans un fichier, préalablement sauvegardé en ASCII sur une disquette.

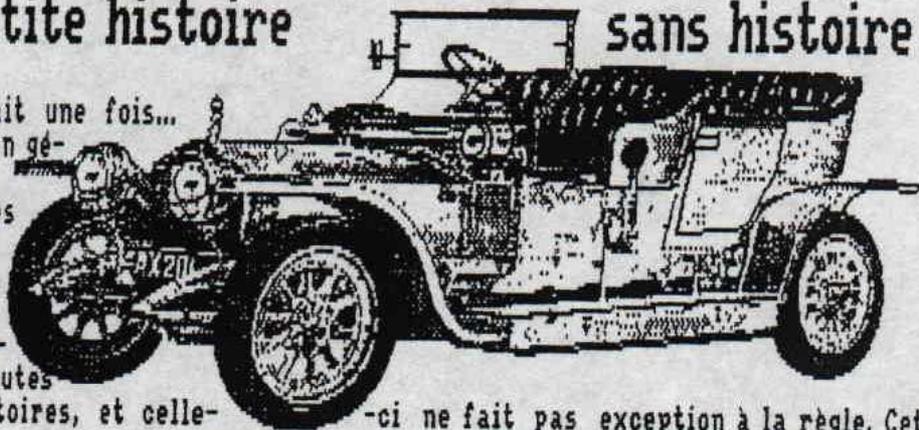
D'un usage extrêmement simple, ce programme vous demande, après avoir affiché le catalogue de la disquette, de donner le nom du fichier où se trouve la chaîne à rechercher. Donnez ensuite le contenu de la chaîne, objet de vos recherches et le tour est joué. Vous remarquerez l'usage qui est fait de la boucle (<WHILE> <WEND>) ainsi que INSTR(c\$,a\$).

D I V E R S

Petite histoire

sans histoire

Il était une fois...
C'est en gé-
néral
par ces
quatre
mots
que
commen-
cent toutes



les histoires, et celle-ci ne fait pas exception à la règle. Cette histoire commence il y a quelques années en arrière. Lorsqu'un type bien sous tous rapports, décide d'écrire des jeux de rôle, (très en vogue à cette époque) Il prend donc son ordinateur d'une main, son stylo de l'autre, le papier entre les dents, et il commence la programmation de jeux fabuleux, mais hélas en texte seulement, car à cette époque, il ne possède qu'un CBM 4032 qui lui, ne fait pas de beaux dessins, mais il a une très grosse mémoire, pensez donc, 32 K ce n'est pas rien. Nous en sommes là, lorsque soudain, notre brave programmeur se trouve à cours d'idées pour donner des noms aux nombreux personnages qui peuplent les contrées fantastiques de ses jeux. Mais voilà, si notre type n'a pas d'idées pour trouver les noms, il sait programmer, alors il va se mettre au boulot et programmer un programme (c'est évident non ?) qui lui, (le programme) va trouver les noms de ces personnages. Et oh! Merveille, les noms pleuvent, pas tous utilisables hélas, que faire d'un nom comme FGRIKWE ou NNRSIN, mais dans le lot, il y en a de très bon comme KURKU TYMOR ESWARK DAVAAR et d'autres notre programmeur était comblé de joie. Le temps à passé, notre programmeur a laissé de côté son merveilleux CBM pour loucher de plus en plus sur une machine nouvelle, un Amstrad. Seulement voilà, en abandonnant son CBM, il a aussi abandonné tous les fabuleux jeux de rôles, car sur Amstrad il y en avait plein et même des très bons. Mais ce que notre programmeur n'a pas abandonné, c'est ce petit programme qui trouvait des noms. Il l'a même adapté sur son Amstrad et il l'a mis de côté, sur une disquette, au cas où... Et puis un jour, pas très éloigné, puisque c'était en novembre 1988, notre programmeur décide de ne plus faire de programme, il veut faire autre chose. Et pourquoi pas un petit journal ? Va pour un journal, c'est passionnant ça, un journal. Quel nom il va lui donner à ce journal? Et il se souvient du petit programme qui trouve des noms tout seul, alors il le charge dans la vaste mémoire de son CPC, il fait RUN et met l'imprimante en marche. Une heure après, 500 et quelques noms étaient sur le papier. Notre programmeur qui ne l'était plus d'ailleurs, prend son courage à deux main, et il commence le tri de tous ces noms. Et RUNSTRAD était né. Vous avez tous compris je crois, que cette histoire est authentique. C'est vraiment de cette façon que le nom du journal a été trouvé. J'ai fait un tri et j'ai gardé celui-là. Pourquoi? Il y en avait d'autres pas mal non plus, en définitif je pense que c'est le mot RUN qui l'a emporté sur tout le reste. Mais pourquoi cette histoire? Certains d'entre vous ne le savent pas encore, mais un nouveau magazine a fait son apparition dans les kiosques, et, comble de l'horreur, il porte lui aussi le nom de RUNSTRAD. Or comme je vous le dis dans mon histoire, ce nom a été trouvé par un programme. L'histoire s'est-elle reproduite ailleurs? J'en doute fort. Toujours est-il qu'il n'était absolument impossible de copier un nom qui n'existait pas encore. Alors ???

Petites Annonces

Cherche: LIVRE "LA PRATIQUE DES IMPRIMANTES DE M. ARCHAMBAULT ainsi que DISC OXFORD PAD. A VENDRE: 2X 81. + documentation Le tout en 7-8-E. Prix 200,00 Fr Téléphone: 68.63.59.69

Ach ou Ech contre utilitaires ou jeux: REVUES et LIVRES sur CPC 6128 Roger LORRAIN, Route de Frétte-rans, 71270 PIERRE de BRESSE

SI VOUS ETES INTERESSE PAR UNE INTERFACE RS 232 et par un MODEM REACITEL Contactez la rédaction

VENTES & ECHANGES de SOFTS ORIGINALS AVEC MODE D'EMPLOI UNIQUEMENT Contactez la rédaction

VENDS INTERFACE RS 232 C Absolument NEUVE, JAMAIS utilisée et MAGNETO cassettes COMPUTONE en EXCELLENT ETAT avec cordon spécial Amstrad, ou échange le tout contre SOURIS OU SCANNER OART. BERNARD CAPPI 225 Picas, Jardin St Jacques 66000 PERPIGNANT, TEL:68.34.56.91 heures de bureau

Je recherche toujours un Correcteur Orthographique adapté au CPC 6128 et à un des traitements de textes tournant sur cette bécane... VOTRE PRIX SERA LE MIEN Contactez la rédaction bon sang

Si la CONCEPTION ASSISTEE PAR ORDINATEUR VOUS INTERESSE et si vous envisagiez l'achat d'un bon logiciel, puissant, performant NE CHERCHEZ PLUS J'AI CE LOGICIEL, IL S'AGIT DE C.A.O. de chez LOGICIELS Vendu avec son mode d'emploi de 50 pages. VALEUR ACTUELLE 400F VENDU 200,00. Pour plus amples informations contactez la rédac.



Rédaction
46.67.58.89
de 10 à 02h du matin

OP