

numéro  
2

# QUASAR

SOMMAIRE ————— Pages ———

<i>Editorial &amp; Sommaire</i> .....	29
<i>Actus</i> .....	30
<i>Petites Annonces</i> .....	31
<i>Courrier</i> .....	32
<i>Histoires Perpendiculaires</i> .....	33
<i>Basic au Berceau</i> .....	34
<i>Initiation à l'Assembleur</i> .....	35
<i>Initiation au Turbo Pascal</i> .....	36
<i>Perfectionnement au Basic</i> .....	37
<i>Perfectionnement à l'Assembleur</i> ..	38 - 39
<i>Perfectionnement au Turbo Pascal</i> .....	40
<i>Bidouilles</i> .....	41
<i>Concours</i> .....	42

## EDITORIAL

Salut !

Voici Quasar CPC numéro 2 (1er trimestre 93) réalisé par toute l'équipe de Futurs'. Pour ceusses qui n'auraient pas vu passer le numéro 1, on se doit de préciser que ce dernier était sur papier et qu'il est toujours disponible à condition de payer les frais de port (5F) et de nous joindre une enveloppe (format A4) avec, si possible, un chèque de 35000 francs à l'ordre de Futurs' (mais non Tony, c'est une blague !).

Bref, je vous souhaite un agréable voyage dans l'univers de Quasar CPC...

La Rédac' —

Mais avant tout, quelques messages :

- Tony, t'énerve pas, je te promets de te réserver une rubrique Tests plus importante dans le prochain numéro.

- Zack, merci pour les trois musiques qu'on a pu repomper dans ton fanz. Ce sont celles qu'on a utilisées pour le menu, le Turn Disk A/B et maintenant. Eh oui, Zik était à sec. Merci aussi pour avoir diffusé mes routines assembleur et les musiques de Zik.

Cette fois-ci, je vous quitte en pensant à Poum, Sined, Zede, Franck Einstein, Septh, Miss X, Robby (vive ADPC !) mais aussi aux Logon System et à Crocus.

A tout hasard, (re)voici notre adresse :



Quasar CPC / Futurs'  
Philippe & Gilles RIMAURO  
8, chemin des Maillos  
Quartier Lédar  
09200 SAINT-GIRONS

# ACTUS

Que mettre sous ce titre prometteur... Ah oui, vous avez été très nombreux (hem !) à nous écrire à propos de la TMCV (en fait "Tu Me Crois Vraiment ?"). Eh bien, vous pouvez la chercher longtemps, cette fameuse ROM... elle n'existe pas et, je le crains, n'existera jamais ! En passant je dis un bravo à Tony qui ne s'est pas fait avoir (pour une fois). En fait, c'était notre poisson d'avril... mais à cause d'un retard de parution, le poisson n'était plus très frais.

À part ça, je n'ai pas grand chose à raconter mis à part le fait que le numéro 1 de Virtual World est sorti en Août, et que le numéro 2 ne devrait plus tarder ; il s'agit d'un fanz' pas mal du tout qui a, il faut le dire, un aspect démo ; en deux mots, je vous le conseille. Pour vous le procurer, écrivez à la la rédac de Quasar en envoyant une disquette (pas vide si possible) et une enveloppe timbrée.

Quoi de neuf dans le monde des grands ? Pas grand chose d'inattendu vous dirais-je : les logiciels pour CPC se font de plus en plus rares (seuls quelques éditeurs fidèles subsistent), Amstrad Cent Pour Cent continue tant bien que mal à nous tenir au courant de l'évolution du monde du CPC, etc...

En effet, les actus sont de plus en plus dures à trouver et, comme ce n'est pas mon style, je me vois mal vous répéter ce qui aura été dit dans d'autres journaux ou fanzines sortis avant Quasar, donc...

Ce sera tout !



OffseTasec



Peu de petites annonces pour ce numéro ci...

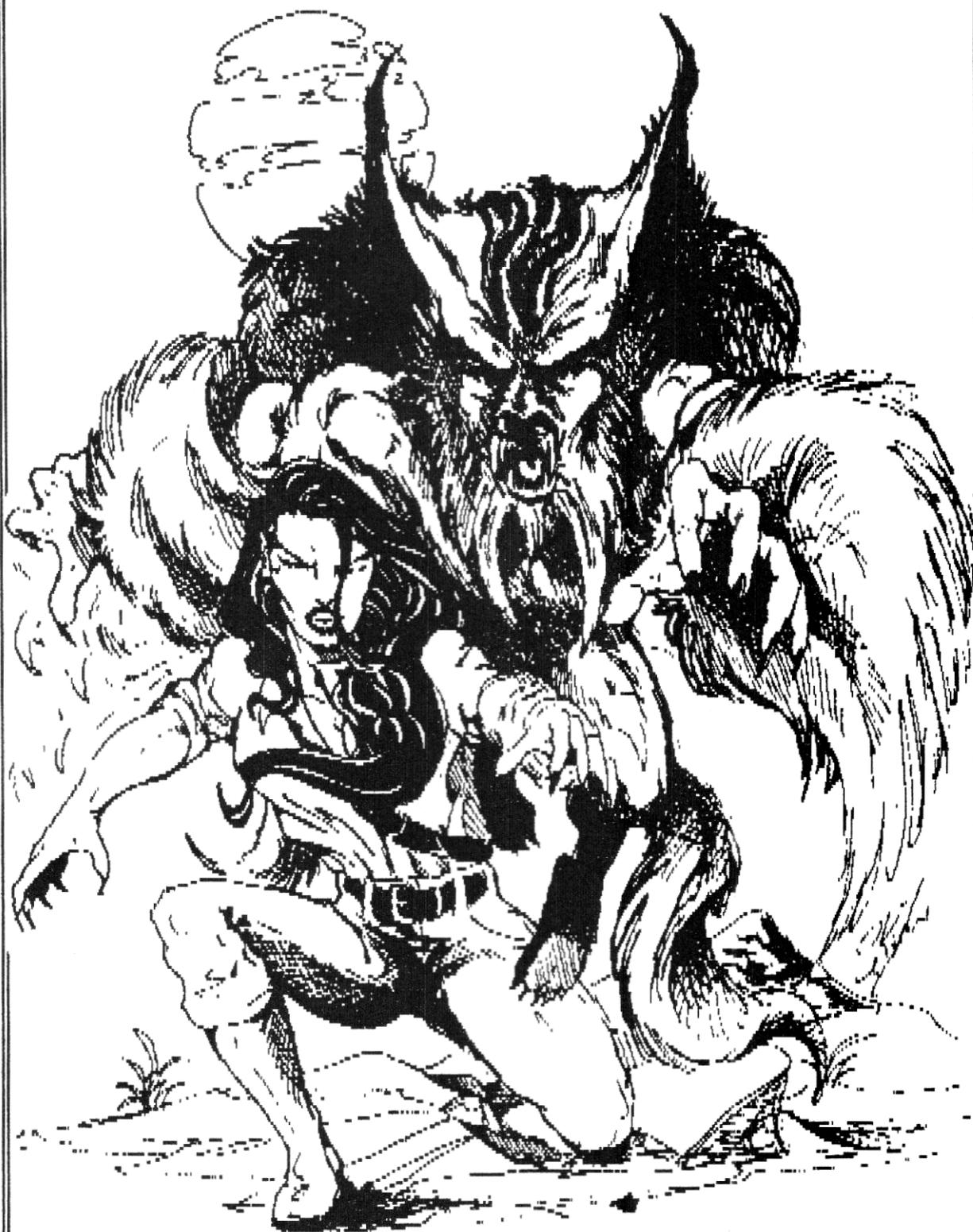
Recherche THE MUSIC MACHINE à un prix raisonnable, contacter la rédac'.

Achèterais la Bible et/ou les Clefs pour Amstrad, écrire à la rédac'.

Cherche contacts pour échanger des routines, programmes, ou autre, contacter Offset.

Voilà, c'est tout ; c'est bref et précis !

 Offset



# Courrier

Voilà un article qui va être bien court à taper. A ce jour, nous n'avons reçu que deux petites lettres ; les voici :

**QUESTION :** Salut Offset et Zik ! Je vous écris pour vous faire part d'un problème auquel je ne trouve pas de solution. En fait, c'est simple, je voudrais réaliser un programme qui transforme les nombres arabes en chiffres romains pour un jeu que je suis en train de concevoir...

→ Plotting

**REPONSE :** No problem, tu trouveras le prog qui réalise cela sur la face B, le fichier se nomme "CHIFFRES.BAS". Au fait, tu n'as pas précisé dans quel langage tu le désirais, j'espère que le Basic te conviendra.

**QUESTION :** Hug ! Je voudrais tout d'abord vous dire que votre fanz est hyper ! Mais le concours est bien trop hard ! Ensuite, ma question : comment modifier la taille de l'écran comme c'est souvent le cas dans les jeux ou les démos ?

→ Momo la terreur

**REPONSE :** Pour modifier les paramètres de l'écran, il faut s'adresser au CRTC6845 du CPC (le contrôleur vidéo). Le registre 1 modifie la taille X de l'écran (40 en standard) et le 6 la taille en Y (25 en standard). Pour aller plus loin, le registre 2 change la position X (46 en standard) et le 7 la position Y (30 en standard). Le mode d'adressage est le suivant :

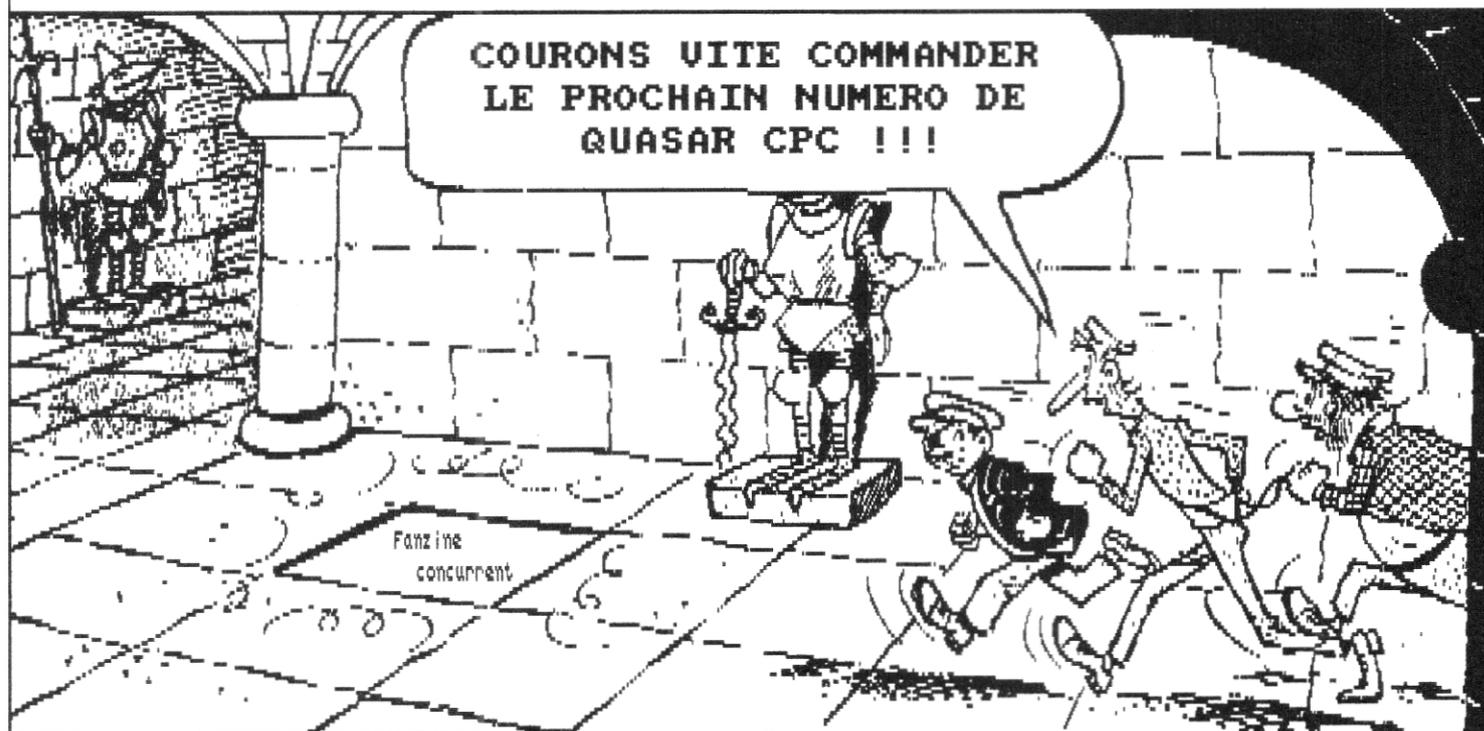
```
OUT &BC00,registre  
OUT &BD00,valeur
```

J'espère avoir répondu à ta question, mais si tu veux plus de détails sur le CRTC c'est possible.

Voilà pour le courrier de ce mois-ci, à la prochaine !

  
Offset

**PUB :** pour en savoir plus sur le hard du CPC, lisez Amstrad Cent Pour Cent !



# Histoires Perpendiculaires

C'est parti pour un long délire ! Je ne sais pas si vous l'avez remarqué mais la rédac' de Quasar s'est étoffée (on prononce Kouazar pour les nouveaux !), en effet, dans le numéro 1, il n'y avait que Zik et moi, mais maintenant nous sommes trois avec Tony qui est venu en renfort, et bientôt quatre avec Basik qui va nous rejoindre pour le numéro trois !

Ainsi, on compte dans notre groupe (FUTURS'), un codeur (qui a fait toutes les routines de ce fanz, moi), un musicien (qui a fait presque toutes les musiques de ce fanz, Zik) et un testeur (qui a réalisé tous les tests, Tony). Relisez bien tout ça, vous avez tout compris, il nous manque un GRAPHISTE ! HELP US !

Bref, changeons de sujet, voilà ce que je vous propose : "Qu'est-ce qu'être civilisé ?", ça ne vous va pas ? Bon alors "La Passion éloigne-t-elle de la réalité ?", ça ne va pas non plus ? Bon, arrêtons ce délire ici, mais, j'y pense, le délire est-il une forme de passion ? A moins que ce ne soit une forme de civilisation ? Agaga ! Bip, beep, paf, pif...

Fotsef (Boum !)

Offset est KO et c'est donc Tony qui prend la relève pour vous présenter en exclusivité mondiale et même interstellaire : son TEST DE (TRES) VIEUX JEUX !

C'est parti !

LA POURRITURE COMPLETE alias RAMBO : Ce jeu étant tellement nul, nul, nul, je vais vous commenter son impression face à moi quand j'en ai eu marre de le voir face à moi aussi nul face à moi... Après

avoir mis de l'énergie infinie (je vous dirai comment si ça vous intéresse mais c'est tellement nul, nul, nul...), je suis parti en fonçant droit devant moi. Au bout de 30 secondes, je vois un tank ; je lui tire dessus, rien ne se passe. Je vais dessus : tiens, "ça" m'attendais. Je me dis "J'ai fini le niveau 1". Mais non, après un barratin d'une dizaine de mots, il m'affiche "END". Voilà un jeu qui se finit en 30 secondes. Je le

répète : c'est nul, nul, nul, nul, nul, nul à l'infini...

Mes Notes :

GRAPHISME : 11/20  
ANIMATION : 10/20  
MUSIQUE : 13/20  
BRUITS : 01/20  
RICHESSSE : 00/20  
JOUABILITE : 01/20

NOTE GLOBALE : 06/20 !

Je vous quitte dégoûté. Ce jeu est trop 99 99 99 99.

Tony équeuré !



# Basic au Berceau

Gut neues Jahr! Haben Sie ein glückliches Weihnachtsfest? Ich... oups! Parlez-vous allemand? Non !!?

Soit... ce soir je vais vous parler des différents types de variable en Basic. Il existe 3 types de variable :

- Le type entier, variables suivies de l'extension "%",  
ex : a%=125

- Le type réel, variables suivies de l'extension "!" ou de rien en standard,

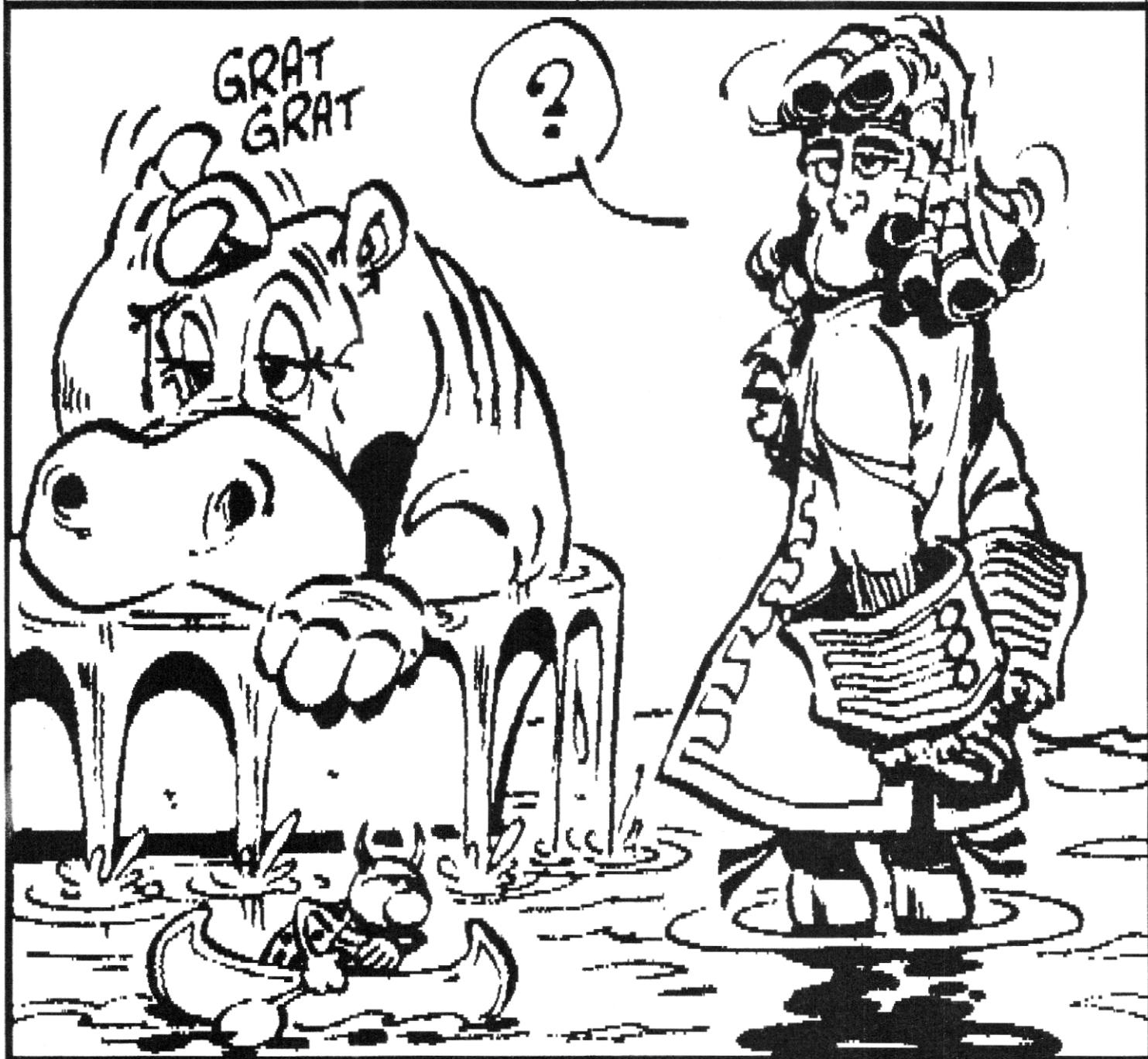
ex : a!=1.2 ou a=1.2

- Le type chaîne de caractères, variables suivies de l'extension "\$",

ex : a\$="Bonjour !"

Et voilà, c'est fini pour ce numéro. La prochaine fois, je parlerai de la reconfiguration du type de variable en standard !

Zik



# Initiation à l'Assembleur

Cette fois-ci, je vais vous parler, comme promis au premier numéro par Offset, des boucles et des datas.

Commençons par les boucles. On utilise l'instruction JP (jump) pour faire un saut dans un programme. Je m'explique : si, par exemple, on veut qu'un programme recommence au début quand son exécution est terminée, on place l'instruction JP DEBUT en fin de programme ; DEBUT étant un label placé au début du prog. dans notre exemple (DEBUT ayant été pris au hasard). Mais, j'y pense, on ne vous a pas parlé des labels ! Bon, je vais vous expliquer. Un label est un mot (ou une lettre) quelconque qui est placé en face d'une commande donnée, il sert à ne pas avoir à préciser l'adresse mémoire où se trouve l'instruction, mais seulement préciser le nom donné au label concerné. Pour mieux vous illustrer cela, vous n'avez qu'à regarder le programme qui doit suivre (on ne sait jamais).

```
ORG &5000
LD HL,TEXTE
BOUCLE LD A,(HL)
CP 0
RET Z
CALL &BB5A
INC HL
JP BOUCLE

TEXTE "Bonjour !",0
```



Je change de paragraphe pour aérer un peu. En observant le prog. vous avez pu voir l'instruction RET Z (eh oui !). Vous devez connaître RET mais pas Z (qu'est-ce qu'il va encore nous sortir !?). Ce fameux Z signifie "si le flag est égal à zéro tu fais l'instruction". Le flag est modifié par certaines commandes dont CP (qui signifie "compare" au registre A), ce qui veut dire en simplifiant (dans notre prog.) : ComPare A à 0, si A=0 alors RETourne au Basic sinon on continue.



Il me reste à vous parler des datas. C'est le même principe qu'en Basic, on réserve une zone mémoire pour y stocker des données, sauf que DATA est remplacé par DB (ou DW pour des tableaux d'adresses) et que la commande READ du Basic est remplacée par des LD. LD HL,TEXTE fait pointer HL sur les données (ici le texte), on lit celles-ci grâce à LD A,(HL) et on incrémente le pointeur (ici INC HL) pour pointer sur la donnée suivante. Vous pouvez déduire que le texte doit être terminé par un 0 et que ce programme correspond au PRINT du Basic (zai oublié de le dire !).

Zik

# Initiation au Turbo Pascal

Salut tout le monde ! Comme promis la dernière fois, je vais continuer l'énumération des principales commandes du Turbo 3. Ainsi, ce numéro sera consacré aux boucles. Vous connaissez déjà le REPEAT UNTIL ; il ne me reste donc plus qu'à vous parler des deux autres boucles du Turbo Pascal.

Tout d'abord, il y a le classique : "FOR x:=v1 TO v2 DO" qui va exécuter l'instruction le suivant autant de fois qu'il le faut pour que v1 soit égal à v2. Il y a peu de différences par rapport à son homologue Basic, mis à part que l'instruction STEP n'existe pas ; il n'est donc pas possible de faire avancer un compteur de 2 en 2 ou autre. En revanche, il est possible de décrémenter le compteur (STEP-1) en remplaçant le TO par un DOWNTO. Mais cet inconvénient est pallié par le fait que la variable utilisée peut être de n'importe quel type ; il est donc possible de réaliser des boucles du type "FOR x:='a' TO 'z' DO" voire "FOR y:='z' DOWNTO 'a' DO". Je pense que ce sera tout pour cette boucle.

Ensuite, il y a le "WHILE condition DO" (qui existe aussi en Basic). Son fonctionnement est parfaitement identique au Basic, c'est pourquoi je ne m'étendrai pas dessus.

Attention ! En effet, l'utilisation de ces boucles diffère légèrement par rapport à celle du REPEAT UNTIL, car vous aurez constaté qu'il n'y a pas de bornes permettant de délimiter l'action des boucles (NEXT et WEND en Basic). Il est donc obligatoire de mettre la partie concernée par la boucle entre BEGIN...END. Votre routine aura donc l'aspect suivant :

```
FOR x:=v1 TO v2 DO
BEGIN
  Instructions
END;
```



Il n'est toutefois pas nécessaire de placer ce BEGIN...END si la boucle ne concerne qu'une seule instruction (note : deux instructions sont toujours séparées par un point-virgule, cf numéro 1).

J'espère que vous avez tout compris car il paraît que je ne suis pas clair... Au cas où, je vous ai mis un programme d'exemple sur la face B (le prog se nomme

BOUCLES.PAS mais vous devrez tout d'abord le copier sur une disquette système).

La prochaine fois, j'évoquerai les fonctions de gestion du clavier.



Offset



## BOUCLES.PAS

```
(* Programme d'exemple d'utilisation des boucles : *)
(* Rechercher des nombres premiers entre 1 et 100. *)
```

```
Program boucles; (* Déclaration du programme *)
```

```
Var x,y:integer; (* Déclaration des variables *)
    a:boolean;
```

```
Begin (* Début du programme *)
```

```
a:=true;
```

```
For x:=1 To 100 Do
```

```
Begin (* Début de boucle *)
```

```
a:=true;
```

```
For y:=2 To x-1 Do
```

```
If x Mod y = 0 Then a:=false;
```

```
If a Then Writeln(x:3,' est un nombre premier...');
```

```
End; (* Fin de boucle *)
```

```
End. (* Fin du programme *)
```

```
(* NOTA : *)
```

```
(* La commande MOD fonctionne comme en Basic. *)
```

```
(* On peut optimiser ; j'attends vos propositions. *)
```

# Perfectionnement au Basic

La gestion des fichiers. C'est de cela dont nous allons parler comme promis. Voyons tout d'abord les commandes d'ouverture et de fermeture des fichiers. Suivant que vous vouliez ouvrir un fichier ASCII en lecture ou en écriture, vous utiliserez deux jeux de commandes distincts. Tout d'abord, je vais traiter l'ouverture en écriture. OPENOUT"nomfich" sert à créer un fichier et à l'ouvrir en écriture. PRINT#9 et WRITE#9 inscrivent dans le fichier les données spécifiées après la dite instruction. Si vous voulez voir la différence entre ces deux commandes tapez ceci sous Basic :

```
WRITE"Bonjour"
```

puis

```
PRINT"Bonjour"
```

...et constatez vous même la différence. Pour sauver le fichier, il faut le fermer, et pour cela, il suffit d'utiliser CLOSEOUT.

Pour relire le fichier sauvé, on doit l'ouvrir en lecture. Je ne dirai qu'un mot : OPENIN"nomfich". Bon, maintenant qu'il est ouvert, il faudrait peut-être le lire non ? Et pour ce faire, on emploie INPUT#9 qui lit la donnée courante dans le fichier, la stocke dans la variable spécifiée, et pointe sur la donnée suivante. Il vaut mieux ensuite le fermer grâce à CLOSEIN.

Il est impossible d'ouvrir deux fichiers en écriture ou en lecture ; mais, en revanche, on peut ouvrir un fichier en écriture et un autre en lecture simultanément.

Voyons, voyons... Ah oui ! EOF (End Of File) ! Voilà, cette commande (EOF) renvoie 1 lorsque toutes les données du fichier courant ont été lues et 0 dans tous les autres cas. La plupart du temps, on l'utilise de la manière suivante :

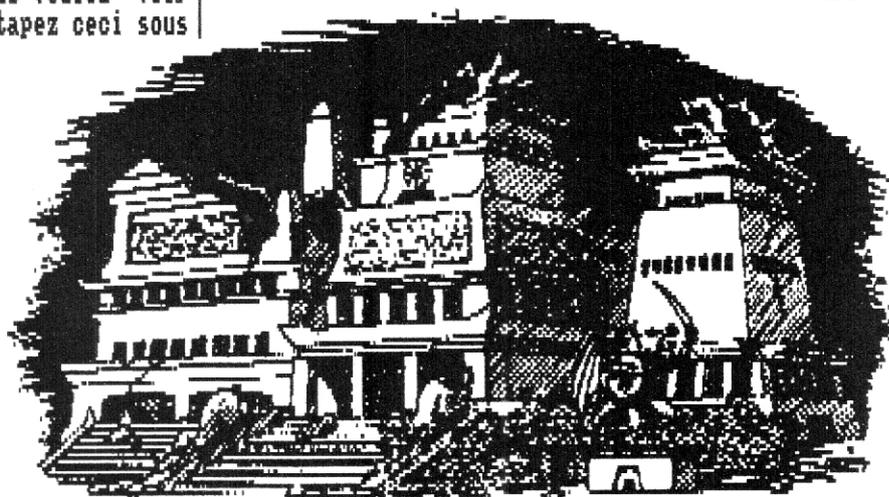
```
10 WHILE NOT EOF  
20 INPUT#9,variable  
30 WEND
```



Ainsi, toutes les données du fichier seront lues.

Un prog. d'exemple nommé "FICHIER.BAS" se trouve normalement sur la face B de la disquette. Si vous n'avez pas compris quelque chose, écrivez-nous...

Zik



FICHIER.BAS

```
5 MODE 2  
10 OPENIN"donnee"  
20 INPUT#9,a  
30 CLOSEIN  
40 IF a=0 THEN GOTO 60 ELSE GOTO 140  
45  
50 ' ** Saisie des donnees **  
55  
60 OPENOUT"donnee"  
70 INPUT"Entrez un nombre : ",nbr  
80 INPUT"Entrez un mot : ",mot$  
90 WRITE#9,1,nbr,mot$  
100 INPUT"Voulez-vous continuer la saisie (O/N) ";a$  
110 IF UPPER$(a$)="O" THEN GOTO 70  
120 CLOSEOUT:END  
125  
130 ' ** Lecture et affichage des donnees du fichier **  
135  
140 OPENIN"donnee"  
150 WHILE NOT EOF  
160 INPUT#9,a,nbr,mot$  
170 PRINT"Nombre : ";nbr  
180 PRINT"Mot : ";mot$  
190 WEND:CLOSEIN  
200 OPENOUT"donnee"  
210 WRITE#9,0  
220 CLOSEOUT  
230 END
```

# Perfectionnement à l'Assembleur

Comme promis en Août, je vais donc vous expliquer comment on doit s'y prendre pour faire bouger des rasters à l'écran. Partant du principe que l'affichage d'un raster à l'écran n'a plus de secret pour vous, sa mise en mouvement est fort simple. Il s'agit tout simplement de jongler entre le raster et un tableau. Ainsi, on fait une table pour le raster, puis on la sauve dans le tableau qui est ensuite envoyé à l'écran. Je m'explique : soit un raster stocké dans un tableau et un tableau dans lequel sera stocké le raster ! C'est plus clair comme ça non ?

Bref, voici donc l'algorithme qu'il faut suivre :

- Début
- Attendre la synchro.
- Pointer sur le tableau
- Afficher le tableau
- Effacer le tableau
- Y sauver le raster à sa nouvelle position
- Reprendre au début.

Comme il n'y a rien de plus clair qu'un programme d'exemple (surtout quand c'est moi qui fais les cours), vous en trouverez un entièrement commenté sur la disquette de Quasar CPC.

Mon programme d'exemple est en fait une variante de l'intro d'affichage du menu principal de Quasar. Mais il va de soi que si vous avez compris le système, rien ne vous empêche de faire un raster qui rebondit ou même, qui suit les courbes de sinus, ou qui tient compte de l'énergie cinétique et, pourquoi pas, une animation sur le principe des oscillateurs harmoniques...

La prochaine fois, on parlera de la gestion des écrans overscan et peut-être même du mode d'interruption vectorisé (IM 2).

 Offset



Programme d'exemple de mise en mouvement d'un raster avec un tableau.  
De plus ce programme restitue des couleurs (à modifier dans "Image") pour donner l'illusion qu'une image s'affiche sous le raster...  
En deux mots c'est pratiquement le même programme que celui de l'intro du menu principal de Quasar.

Note...vous remarquerez que le tableau est stocké à l'envers étrange non ?

```
Org &8000
Ent
Nolist
Long Equ 12 ; Long = nbre de lignes du raster + 1
```

Adieu les interruptions !!!

```
di ; Tout ceci sert à installer un
ld hl,(&38) ; EI, RET
ld (inter),hl ; en &38 pour éviter d'utiliser
ld hl,&c9fb ; des HALT et pouvoir donc se
ld (&38),hl ; synchroniser sur les
; interruptions

Prog ld b,&f5 ; Synchronisation verticale
Synchro in a,c ; On attend le début du
; balayage vidéo...
rra ; puis on se barre !
jp nc,synchro ; On attend le début de la ligne
ds 36 ; suivante (synchro horizontale)
ld hl,image ; On sélectionne le papier
ld bc,&7f00 ; Et hop !
out (c),c ; On envoie les couleurs
outi ; de l'image
; Encre 0
inc b ; Hop !
inc c ; Encre 1
out (c),c ; Hop !
outi ; C'est pas fini !
inc b ; Patience...
inc c ; Encre 2
out (c),c ; Hop !
outi ; La fin approche...
inc b ; Encre 3
inc c ; Hop la !
out (c),c ; C'est fini...
outi ; On remet b comme il faut...
inc b ;

ld hl,tableau ; On pointe sur le tableau
ld a,(hl) ; Hop !
cp 0 ; Si 0 Alors fin raster donc...
jp z,suite ; On passe à la suite !
ld c,16 ; Sélection du Border
out (c),c ; Hop !
out (c),a ; Modif du border
ld c,0 ; Sélection de l'encre 0
out (c),c ; Modif de l'encre 0
```

```
out (c),a ; Hop !
inc hl ; On passe à la ligne suivante
ds 32 ; Patience...
jp loop2 ; C'est reparti pour un tour...

Suite ld a,64+20 ; On remet tout à zéro
ld c,1 ; Encre 1
out (c),c ; Hop !
out (c),a ; Zéro
inc c ; Encre 2
out (c),c ; Hop !
out (c),a ; Zéro
inc c ; Encre 3
out (c),c ; Hop !
out (c),a ; Zéro
xor a ; On se prépare pour
ld hl,(courant) ; réinitialiser le tableau
ld b,long ; juste ou il le faut...
Loop3 ld (hl),20+64 ; C'est tout bete
dec hl ; Une petite boucle
djnz loop3 ; s'occupe de tout !
ld hl,(courant) ; On passe à la ligne
inc hl ; suivante...
ld (courant),hl ; On sauve pour savoir ou
cp (hl) ; on en est au prochain coup !
jp z,fin ; Si fin tableau bye !
ld de,raster ; Sinon on y sauve le raster...
ld b,long ; Ne me demandez pas pourquoi
Loop1 ld a,(de) ; je n'ai pas fait avec LDDR ou
ld (hl),a ; LDIR car ce serait trop long
inc de ; à expliquer...
dec hl ; Vous comprendrez bien par
djnz loop1 ; vous meme...
jp prog ; On repart pour une VBL !
Fin ei ; Ici, pas de miracle
ld hl,(inter) ; On restitue le système
ld (&38),hl ; proprement avant de
ret ; partir...
```

; Datas

```
; Couleurs de l'image à restituer (MODE 1)
Image db 64+20,64+30,64+10,64+14
```

```
; Raster à faire défiler...Attention !
; Il doit commencer par zéro et les
; couleurs doivent être mise à l'envers !
```

```
Raster db 0
db 64+20,64+4,64+21,64+23,64+31,64+19
db 64+11,64+10,64+14,64+12,64+28
```

```
; Le tableau ou l'on stockera le raster
```

```
Tableau db 0
ds 310,64+20
db 0
```

```
; Sauvegarde Interruptions
Inter dw 0000
```

```
; Pointeur du tableau
Courant dw tableau+long-1
```

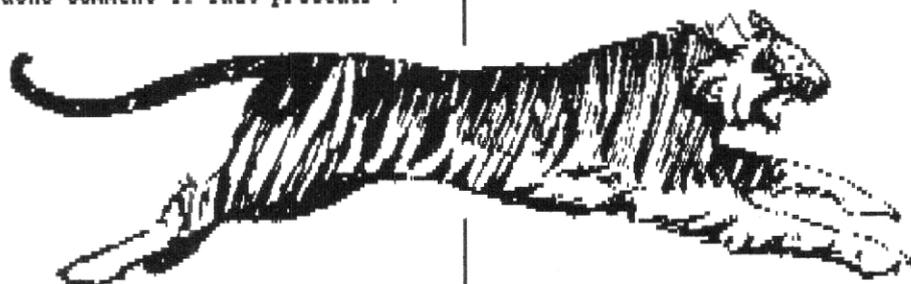
# Perfectionnement au Turbo Pascal

C'est parti pour les tableaux ! Vous allez enfin pouvoir donner une nouvelle dimension à vos programmes en Turbo Pascal. Sur le plan pratique, l'utilisation des tableaux en Turbo Pascal 3 est exactement la même qu'en Basic, si ce n'est qu'il ne sont pas auto-initialisés à zéro (il faudra donc l'initialiser manuellement). Le seul problème réside dans leur déclaration ; voici donc comment il faut procéder :

Pour de plus amples explications, il ne vous reste plus qu'à voir le programme TABLEAUX.PAS de la face B (il vous faudra le copier sur une disquette système avant de l'utiliser). Pour Avril, je vous prépare un petit article sur le mode graphique et l'accès direct aux vecteurs du CP/M.



Offset



```
Var Tab:Array[a] Of b;
```

- a représente les indices, c'est-à-dire une liste de constantes. Par exemple 1..500 équivaut à DIM(500) en basic.
- b représente le type du tableau. On peut déclarer un tableau de n'importe quel type (Char, String, Integer, Real, Boolean, etc.)

Vous aurez sans doute remarqué la souplesse d'une telle déclaration ; en effet, contrairement au Basic, on peut déclarer les indices comme on le désire (de 152 à 341 par exemple : 152..341 à la place de a). De plus, les indices peuvent être de n'importe quel type, même un type défini par l'utilisateur...

Pour vous éclairer un peu plus, voici quelques exemples de tableaux :

```
Tb:Array['A'..'Z'] Of b  
Tb:Array['q'..'y'] Of b  
Tb:Array[100..510] Of b  
Etc.
```

Les tableaux à deux, trois, ou n dimensions se déclarent comme en Basic, en séparant les dimensions (dans la déclaration du tableau) par des virgules.

## TABLEAUX.PAS

```
(* Voici un petit programme qui ne *)  
(* sert strictement à rien mais qui *)  
(* illustre bien le fonctionnement *)  
(* des tableaux en Turbo Pascal... *)
```

```
Program Exemple;  
Var tab:Array[1..80,1..24] Of char;  
x,y:integer;
```

```
Procedure Rempli;  
Begin  
Clrscr;  
Write('Patientez SVP');  
For x:=1 To 24 Do  
Begin  
Write(' ');  
For y:=1 To 80 Do  
tab[y,x]:=chr(round(random*200)+32);  
End;  
End;
```

```
Procedure Affiche;  
Begin  
Clrscr;  
For x:=1 To 24 Do  
For y:=1 To 80 Do  
Write(tab[y,x]);  
End;
```

```
Begin  
Rempli;  
Affiche;  
Repeat until keypressed;  
End.
```

# Bidouilles

Une nouvelle rubrique dans Quasar ! Ici, nous traiterons un peu de tout ; notre objectif sera d'échanger des routines sympas !



Pour l'entrée en scène de la rubrique, il faut avouer que je manque un peu d'idées ; c'est pourquoi cet article sera entièrement consacré aux... je vous le donne en cent... je vous le donne en mille... aux bidouilles ! Etonnant non?!? Eh bien non, cet article ne sera pas consacré aux bidouilles pour la simple et bonne raison que je n'ai reçu aucune bidouille, et que mon objectif n'est pas de vous soumettre les miennes, car vous pourrez peu à peu les découvrir dans les divers cours de programmation de Quasar.

Donc, un simple petit poke à faire avec la Multiface pour faire plaisir à ce pauvre (petit) Bouchon (avec un B majuscule s'il vous plait !). Il concerne Rick Dangerous premier du nom :

- POKE &88C3,&00 : donne les vies infinies,
- POKE &9F35,&00 : donne le tir infini,
- POKE &9F90,&00 : donne les bombes infinies.

Voilà, ce sera tout, je sais que ça n'est pas terrible, mais j'espère que ça vous conviendra tout de même...

Envoyez-moi tout un tas de bidouilles pour qu'on puisse s'éclater la prochaine fois ; j'accepte tout, Basic, Assembleur du Z80, Turbo Pascal, etc.. L'essentiel, c'est que je comprenne votre truc afin de pouvoir l'expliquer !



Offset

# Concours

Et voilà ! Une horrible tache d'encre sous mon joli titre ! Zik ! Qui t'a dit de mettre une tache dans la fonte du fanzine ?? A m'énerver comme ça, j'en oublie pourquoi je suis là...

Ah ! Le fameux concours du numéro 1. Eh bien, comme vous n'avez pas été vraiment très nombreux à participer (je n'ai pas reçu une seule (bonne) réponse), je ne vais pas en donner les résultats. S.V.P., écrivez-nous ! (ceci est un appel désespéré !).

Non, je sais très bien pourquoi je n'ai pas eu de réponses : c'était trop difficile ! En effet, j'ai eu tout de même quelques échos.

Bon, je n'ai plus grand chose à vous dire, sinon que les questions du concours sont quelque part sur la D7.

J'attends vos réponses avec impatience... au fait, pour ceusses qui n'ont pas vu le premier numéro, le sujet du concours c'était : "Les CPCistes connaissent-ils bien leur CPC ?". Et les prix : pour le premier, un câble de téléchargement ; du second au dernier, une lettre de réponse ! Il va de soi que nous ne répondrons qu'à ceux qui nous auront donné les bonnes réponses (Je sens qu'il n'y en aura pas tant que ça...).



Offset'ador !





