

Le rédacteur  
Benoît Rivière  
8 rue Victor Hugo  
13730 Giberville  
FRANCE

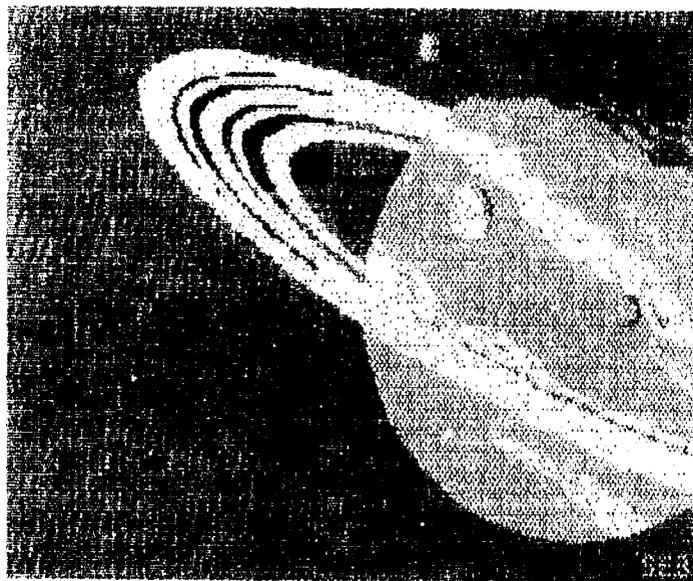
# ORIGINAL IMAGE

ORIGINAL, le *LANZING* des passionnés de l'Amstrad CPC et du PC.

**LANZING**  
**CPC**  
&  
**PC**

numéro 6 - août 1993

Apériodique gratuit 18 pages



## Édito

Bonjour tout le monde ! Nous sommes de retour avec cette fois-ci, pour le bonheur de certains et le dégoût d'autres, un dossier consacré aux mathématiques (programmes divers accompagnés d'explicatifs...). Sinon vous pourrez lire la suite de l'interminable saga des adresses des CPC, taper des programmes, résoudre vos jeux favoris...

*Bons bronzages à tous, Benoît.*

## Sommaire

- p. 2 : Initiation pratique à EXCEL 4.0
- p. 5 : Les adresses des CPC
- p. 8 : La programmation sur CPC
- p. 10 : Au banc des accusés
- p. 11 : Programmation sur PC
- p. 15 : Dossier mathématiques
- p. 24 : Au secours ! Help ! Socorro !

PC et CPC

X0000 - 6 - GRATUIT



# Microsoft Excel

## INITIATION PRATIQUE A EXCEL 4.0

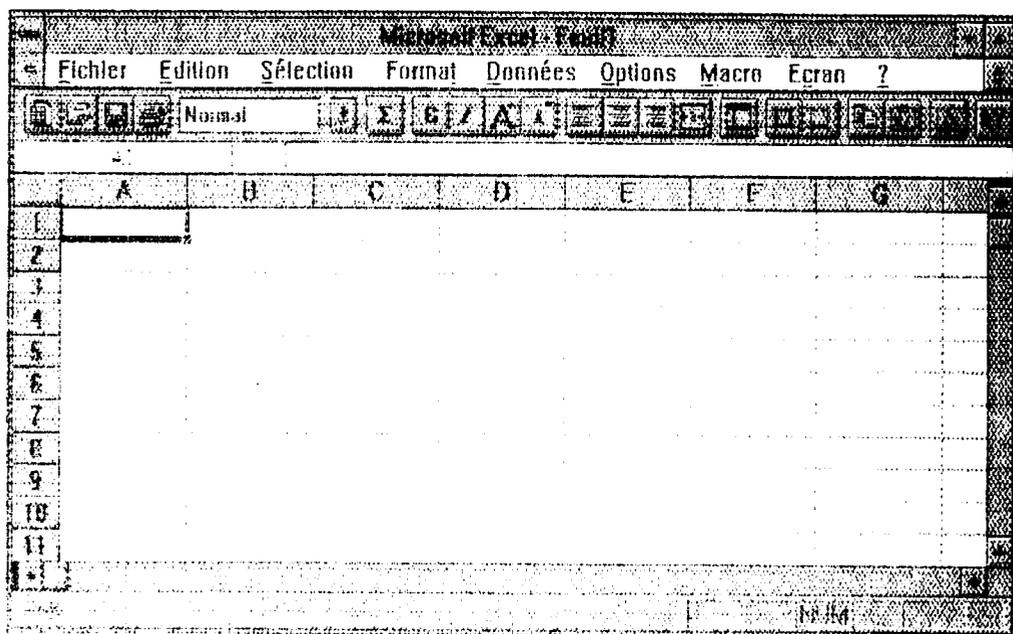
Tous ceux qui font de la gestion, ou qui ont besoin de dresser rapidement des tableaux sont amenés à utiliser un tableur. In des tous derniers tableurs sortis est EXCEL 4.0

Microsoft. On ne peut guère pas le comparer avec ses prédécesseurs (comme Multiplan ou même d'ailleurs) ; on revient à un concept d'interface

type Windows. Ainsi, la principale ressemblance qui subsiste entre Multiplan et Excel est la programmation des cellules

qui à quelques points près est identique. Sinon tout les sépare. Excel possède une barre d'icônes symbolisant les principales commandes les plus usitées, des menus déroulants

permettant une approche très intuitive du logiciel. En outre grâce à l'interface graphique de Windows, on peut



créer des graphiques de toutes sortes en toute simplicité. Ce sont tous ces points que nous allons étudier.

### 1. Tout d'abord...

Il est bien entendu que pour utiliser EXCEL, il est nécessaire qu'il ait été préalablement installé sur le disque dur. Il est également évident que Windows doit nécessairement figurer sur ce même disque. Il est fortement conseillé d'utiliser la souris (d'ailleurs pour cliquer sur l'icône c'est je pense beaucoup plus pratique).



### 2. Le tableau

Une fois l'icône désignant Excel cliquée, vous voyez apparaître un tableau vierge (voir la figure ci-dessus). Pour se familiariser avec l'utilisation du tableur, nous allons commencer par des opérations élémentaires, comme entrer du texte, changer de police de caractères... puis compliquer progressivement.

## 1°) Le titre

Placez-vous en cellule A1, tapez les mots suivants : "Le commerce extérieur de la France en chiffres".

Puis, cliquez de A1 jusqu'à L1, vous obtenez une zone en vidéo-inversee. Appuyez sur ALT+F ou cliquez sur **Format** avec la souris, puis **Position/Centrer sur plusieurs colonnes** : le titre doit se positionner au milieu de l'écran.

Enfin, **Format/Police/Police** : Times New Roman, **Effet** : souligné, **Style de police** : normal, **Taille** : 18, votre titre sera écrit à l'aide de caractères extraits de la police Times New Roman de taille 18 et souligné.

## 2°) Le tableau

Entrez les données du tableau de la page suivante, dans la zone A3:L4, sans vous préoccuper de la présentation etapez les mots en anglais EXCEL, les caractères sont exécutés en caractères appropriés :

Cliquez la zone A3:L4, **Format/Format automatique/Format tableau** (écranque 2), vous avez obtenu un tableau de données correctement alignés. Toujours dans la même zone cellulaire, **Format/Position/Position centre/vertical centre**, les mots et chiffres seront centrés.

Positionnez-vous en A1, puis exécutez **Format/Largeur de colonne** : 10. Restez dans la même cellule **Format/Police/MS Sans Serif** 10, puis **Format/Position** : cliquez la case "Boussole" (à l'opposé du multiplicateur).

Cliquez sur les lettres de colonnes de B à L, afin de les mettre en vidéo inversee. **Format/Largeur colonne** : 6.

## 3°) Le graphique

Cliquez la zone B5:L4 du tableau de données : de façon à le mettre en vidéo inversee. Sélectionnez la

deuxième icône en partant de la droite, à l'aide de la souris, puis cliquez la zone E6:L20 (ces cellules recevront le graphique).

Sélectionnez **Suivant/Histogrammes/Suivant/8/Suivant/Suivant**. **Titre du graphique** : Le commerce extérieur Français. **Titre des axes : valeur (Y)** : en millions de francs/ok.

Cliquez sur la dernière icône, en bas à gauche, ouvrez une fenêtre, avec le curseur juste sous les mois, puis tapez : sources INSEE.

Pressez le bouton droit de la souris, puis cliquez sur **Police/Italique/Taille 9/MS sans serif**. Appuyez à nouveau sur le bouton droit de la souris et sélectionnez tour à tour : **Motif/Contour/Couleur** : blanc.

## 4°) Intégration d'images

Par exemple, si vous possédez Lotus SmartPics, réduisez EXCEL en icône (en cliquant sur la petite flèche orientée vers le bas, en haut à droite de la fenêtre Microsoft Excel - Feuille.xls).

Cliquez SmartPics, ou à défaut faites **Fichier/Executer/lotus\smartpics\smartpic/ok** dans le gestionnaire de programmes.

Choisissez **Files/toons1.syl/Symbol : 8 of 12/Copy to clipboard/Exit**.

Réagrandissez EXCEL. Positionnez-vous en A6, puis **Edition/Coller** : l'image s'affiche alors. Redimensionnez la figurine sur la zone A6:D16.

## 5°) Quelques statistiques

En E22, entrez : Quelques statistiques, puis **Format/Police/Times New Roman/16**

En cellule C24, tapez Moyenne. Faites de même en cellules C25, C26, H24, H25, en entrant respectivement : Somme, Nombre, Maximum, Minimum.

Normalisez la zone B4:L4, puis **Sélection/Définir un nom/tableau/ok**.

Il ne vous reste plus qu'à taper les formules correspondant aux libellés, c'est à dire : (moyenne tableau) en E24, (somme tableau) en E25, (le tableau) en F26, (maxi tableau) en H24 et (min tableau) en H25.

Ensuite **Fichier/Imprimer.../Ok** après un éventuel **Fichier\Aperçu avant impression**.

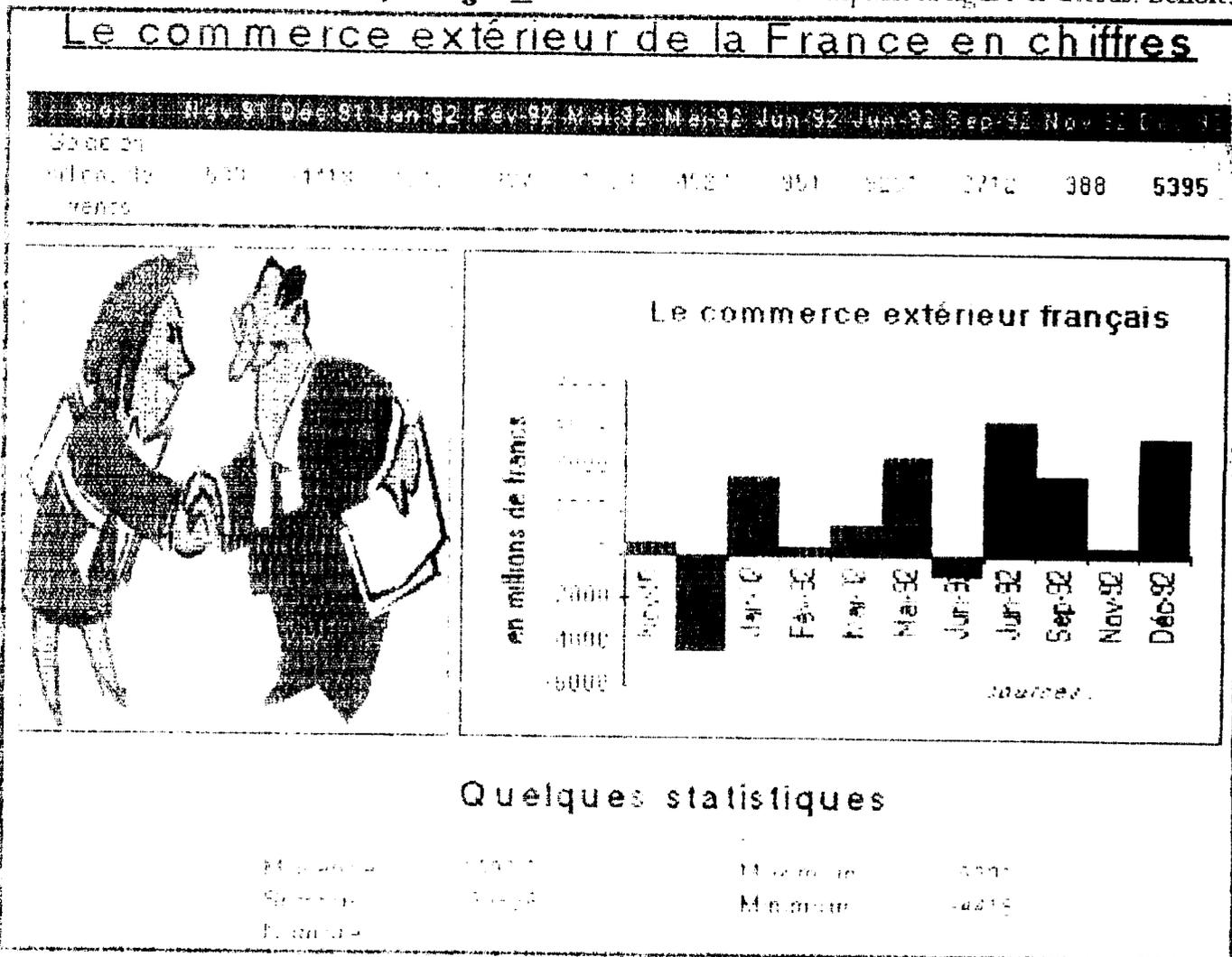
## 7°) Sauvegarde

## 5°) Impression

**Fichier/Enregistrer sous :** comext.

**Fichier/Mise en page/Orientation/Portrait.**  
Papier **Taille A4 (210x297 mm)**. Marges **Gauche**

Et voilà c'est fini, vous venez d'apprendre le principal. Vous pouvez apprécier le résultat de votre labeur en contemplant la figure ci-dessus. Benoît.



**Ecrire**  **Haut**  **Bas**  **Centrer**  :  
horizontalement et **Verticalement** Echelle 100%

Les autres cases ne doivent pas être cochées, saisissez les en tête et pied de page.

Vérifiez que votre imprimante (ou standard imprimante) est bien sélectionnée, en cliquant sur **configurer** puis **Ok**.

**NB :** Les lettres en caractères gras séparées par des barres de division "/" désignent des options de menus. Les lettres soulignées indiquent leur raccourci clavier. Ainsi **Format/Position/Centrer** sur plusieurs colonnes signifie qu'il faut sélectionner l'option Format de la barre de menu, puis position, et choisir de centrer, soit à la souris, soit en frappant successivement, au clavier, Alt+F, P, Alt+C.

# DOSSIER SPECIAL :

## les adresses des CPC

Voici la suite tant attendue de notre dossier consacré aux adresses de nos vieux CPC. Ces adresses sont classées par thème : texte, son, imprimante... Le nom des routines est indiqué (si connu), puis leur utilité, éventuellement leur équivalence Basic, et enfin leurs conditions d'entrée (E) et les sorties (S).

L'utilisation de ces routines à partir de l'assembleur est fort simple. Il suffit de procéder comme suit :

```
ld a,1
call #bc0e
```

Explication : on charge 1 dans le registre A, puis on appelle la routine de changement de résolution d'écran (MODE en Basic).

A suivre... Benoît.

J.B. : Les adresses relatives aux opérations arithmétiques ne seront pas évoquées dans cette rubrique puisqu'elles ont l'objet d'une étude dans le dossier Mathématiques qui figure plus en avant.

## *Routines texte et graphique (suite et fin)*

BC23 : calcul de l'adresse de l'octet à gauche de l'adresse courante, en tenant compte d'éventuels scrollings.

E : HL contient l'adresse courante.

S : HL contient la nouvelle adresse.

BC26 : calcul de l'adresse de l'octet en dessous de l'adresse courante, en tenant compte d'éventuels scrollings.

E : HL contient l'adresse courante.

S : HL contient la nouvelle adresse.

BC29 : calcul de l'adresse de l'octet au dessus de l'adresse courante en tenant compte d'éventuels scrollings.

E : HL contient l'adresse courante.

S : HL contient la nouvelle adresse.

BC2C : SCR INK ENCODE  
encode une encre pour couvrir tous les points (pixels) d'un octet.

E : A contient le numéro d'encre.

S : A encre encodée, F est modifié.

BC2F : décode une encre en numero d'encre.

BC32 : positionne les couleurs d'une encre (INK en Basic).

E : A contient le numéro de l'encre, B la première couleur, et C la deuxième.

S : AF, BC, DE, et HL sont modifiés.

BC35 : demande les couleurs de l'encre courante.

BC38 : positionne les couleurs du bord (BORDER en Basic).  
 E : B contient la première couleur, et C la deuxième.  
 S : AF, BC, DE, et HL sont modifiés.

BC3B : lecture de la couleur du bord.

BC3E : positionne la vitesse de clignotement des couleurs (SPEED INK en Basic).

E : H contient la première durée, et L la seconde.  
 S : AF et HL sont modifiés.

BC41 : lecture de la vitesse de clignotement des encres.

BC44 : colorie un rectangle exprimé en caractère dans une encre déterminée.

BC47 : colorie un rectangle exprimé en adresse écran dans une encre déterminée.

BC4A : inversion de la couleur d'un caractère.

BC4D : SCR HW ROLL  
 déplace (SCROLLING) l'écran entier de huit points (1 caractère = 1 ligne) vers le haut ou le bas, avec effacement de la nouvelle ligne.  
 E : A contient le masque (couleur papier) pour remplir la nouvelle ligne. B contient le sens de défilement (O pour le bas, et FF pour le haut ou BWD).  
 S : AF, BC, DE et HL sont modifiés.

BC50 : déplace une partie de l'écran de huit points (1 caractère) vers le haut ou le bas, avec recopie de la dernière ligne.

E : A contient le masque pour remplir la nouvelle ligne. B contient le sens de défilement (O pour le bas, et FF pour le haut). H contient le numéro de la colonne de gauche, D le numéro de la colonne de droite, L le numéro de la ligne du haut, et E celui de la ligne du bas.  
 S : AF, BC, DE, et HL sont modifiés.

BC53 : conversion d'une matrice de caractère en un ensemble de points approprié au Mode utilisé.

BC56 : fonction inverse de la précédente.

BC59 : positionne le mode graphique en cours.

E : A contient le mode :

0 : normal NEW = INK,  
 1 : XOR NEW = INK XOR OLD,  
 2 : AND NEW = INK AND OLD,  
 3 : OR NEW = INK OR OLD.

NEW = mettre pixel final, OLD = mettre pixel actuel, INK = INK PLOTTE.

S : AF, BC, DE, et HL sont modifiés.

BC5C : écriture d'un point à l'écran sans se soucier du Mode graphique.

BC5F : dessine une ligne horizontale.

BC62 : dessine une ligne verticale.

BD19 : attend jusqu'à ce que le CRT envoie un signal indiquant la fin du balayage vertical (retour du spot) de l'écran (FRAME en Basic). Cette routine permet d'éviter des effets optiques désagréables.

E : aucune.

S : aucune.

BD1C : positionnement d'un mode dans la VGA.

BD1F : positionnement de l'OFFSET de la mémoire écran.

BD22 : met toutes les encres dans la même couleur (effet d'effacement).

E : DE contient l'adresse d'un vecteur d'encre.

S : AF est modifié.

BD25 : positionne la couleur de toutes les encres et du bord.

## Routines musique et son

BCA7 : RESET du générateur. Cette routine arrête tous les sons et efface toutes les queues d'attente, réinitialise les enveloppes (volume, fréquence).

E : aucune.

S : AF, BC, DE et HL sont modifiés.

BCAA : ajoute un son dans une queue. (SOUND en Basic). Le générateur pouvant traiter plusieurs sons sans s'occuper du processeur, cette routine a pour but de continuer le chargement des différents sons quand cela est possible.

E : HL contient l'adresse du programme sonore de 9 octets qui doit se trouver dans les 32K de mémoire vive centrale (entre 4000 et C000).

S : si le son a pu être ajouté à la queue sonore, carry est vrai et HL est modifié. Si toutes les queues sonores sont remplies et que le son n'a pu être ajouté à aucune d'entre elles, carry est faux et HL est préservé. De toute façon, AF, BC, DE et IX sont modifiés et les autres registres sont préservés.

Octet 0 : canaux et synchronisation.

Octet 1 : enveloppe d'amplitude à utiliser.

Octet 2 : enveloppe de son à utiliser.

Octets 3,4 : période de ton.

Octet 5 : période de son.

Octet 6 : amplitude de départ.

Octets 7,8 : durée, ou nombre de répétitions.

Configuration de l'octet 0 : Bit 0 : canal A

Bit 1 : canal B

Bit 2 : canal C

Bit 3 : synchronisé avec canal A

Bit 4 : synchronisé avec canal B

Bit 5 : synchronisé avec canal C

Bit 6 : son tenu

Bit 7 : queue vide

BCAD : vérifie s'il y a de la place dans la queue. (SQ en Basic).  
 E : A contient le numéro du canal testé (bit de test) : A est égal à 1 pour tester le canal A, à 2 pour le canal B, et 4 pour le canal C.  
 S : A contient l'état du canal testé (bit de test) :  
     B2 B1 B0 : nombre de places libres dans la queue,  
     B3 : rendez-vous avec le canal A,  
     B4 : rendez-vous avec le canal B,  
     B5 : rendez-vous avec le canal C,  
     B6 : attente au début de la queue,  
     B7 : canal en train de jouer.  
 B, C, D, E, H, L et flags altérés.

BCB0 : prépare le son pour qu'il se produise quand une queue sonore se vide.  
 E : A : bit de liaison (événement-canal), HL : adresse d'événement.  
     Bit 0 : canal de lien A  
     Bit 1 : canal de lien B  
     Bit 2 : canal de lien C  
 S : A, B, C, D, E, H, L et flags altérés.

BCB3 : permet de rétablir les sons arrêtés sur chaque canal.  
 E : A : canaux à relâcher  
     Bit 0 : relâche canal A  
     Bit 1 : relâche canal B  
     Bit 2 : relâche canal C  
 S : A, B, C, D, E, H, L et flags altérés.

BCB6 : arrêt de tous les sons. Les sons sont automatiquement relancés par CANAL SON (BCAA), RELACHE SON (BCB3), ET CONTINUE SON (BCB9).  
 E : aucune.  
 S : carry : vrai si son actif, sinon faux.  
     A, B, C, D, E, H, L et flags altérés.

BCB9 : démarre tous les sons arrêtés par la routine précédente.  
 E : aucune.  
 S : A, B, C, D, E, H, L et flags altérés.

BCB0 : établit une des 15 enveloppes d'amplitude programmable.  
 E : A numéro d'enveloppe, HL adresse d'amplitude.  
 S : HL contient adresse du bloc de données plus 16 si enveloppe correcte, sinon préserve.  
     A, B, C : préserve si enveloppe incorrecte, sinon altérés.  
     D, E : altérés.  
     carry : vrai si enveloppe correcte, sinon faux.  
     autres flags : altérés.

Octet 0 : nombre de parties de l'enveloppe  
 Octets 1,2,3 : première partie de l'enveloppe  
 Octets 4,5,6 : deuxième partie de l'enveloppe  
 Octets 7,8,9 : troisième partie de l'enveloppe  
 Octets 10,11,12 : quatrième partie de l'enveloppe  
 Octets 13,14,15 : cinquième partie de l'enveloppe

BCBF : établit une des 15 enveloppes de fréquence (ton) programmable.  
 E : A : numéro d'enveloppe, HL adresse du bloc de données de ton.  
 S : HL contient l'adresse du bloc de données plus 16 si enveloppe correcte, sinon préservé.  
     A, B, C : préserve : enveloppe incorrecte.

carry : vrai si enveloppe correcte, sinon faux.

autres flags : altérés

Octet 0 : nombre de parties de l'enveloppe  
 Octet 1,2,3 : première partie de l'enveloppe  
 Octet 4,5,6 : deuxième partie de l'enveloppe  
 Octet 7,8,9 : troisième partie de l'enveloppe  
 Octet 10,11,12 : quatrième partie de l'enveloppe  
 Octet 13,14,15 : cinquième partie de l'enveloppe

BCC2 : fournit l'adresse d'une enveloppe d'amplitude.

E : A numéro d'enveloppe. (A doit être compris entre 1 et 15).

S : A altéré.

BC longueur de l'enveloppe en octets si enveloppe correcte, sinon préservé.

HL adresse de l'enveloppe si enveloppe correcte, sinon altéré.

carry : vrai si enveloppe correcte, sinon faux.

autres flags : altérés.

BCC5 : fournit l'adresse d'une enveloppe de ton.

E : A numéro d'enveloppe. (A doit être compris entre 1 et 15).

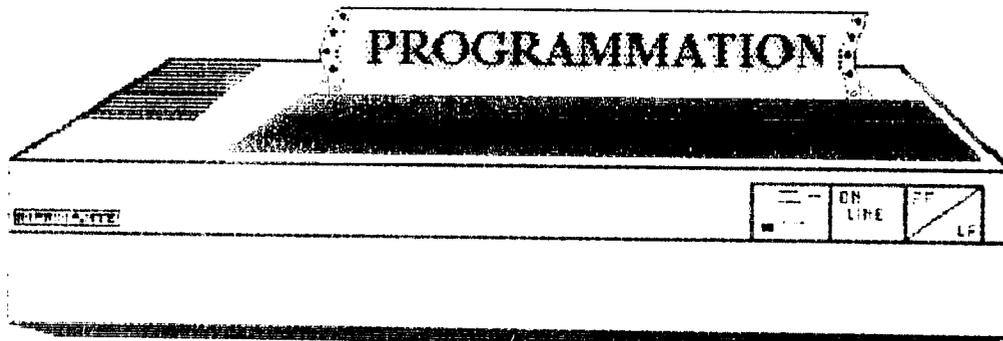
S : A altéré.

BC longueur de l'enveloppe si enveloppe correcte, sinon préservé.

HL adresse de l'enveloppe si enveloppe correcte, sinon altéré.

carry : vrai si enveloppe correcte, sinon faux.

autres flags : altérés.



## Assembleur Z80 sur CPC

Après la simulation du CPC6512, je vous propose un programme imitant un reset. Pour simuler un 6128, tapez POKE &7000,1, pour un 464, tapez POKE &7000,0.

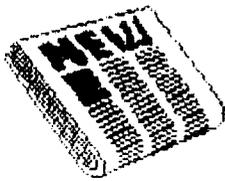
Le code source est publié plus loin afin que vous puissiez l'analyser, et le modifier.

Pour exécuter ce programme, il faut taper le listing de codes. Sauvez-le sous le nom de "RESET.LM", puis faites-le tourner. S'il n'y a pas d'erreurs, il sauvera un fichier nommé "RESET.LM". Ensuite, recopiez le chargeur (le petit programme), enregistrez-le en l'appelant "RESET.BAS". Enfin, RUN "RESET.BAS". Benoit.

```

10 'RESET
20 '
30 'écrit par Benoit RIVIERE
40 'le 27/06/90
50 '
60 'août 1990 I.P.L. Informatique
70 '
80 MEMORY &7FFF
90 LOAD"reset.lm",&8000
100 CALL &8000
110 PRINT
120 INPUT "",a$
130 PRINT"Syntax error"
140 PRINT"Ready"
150 GOTO 120
160 'avant de faire RUN :
170 'pour un 464 ne rien faire
180 'pour un 6128 taper POKE &7000,1
  
```

suite page 13



# AU BANG DES ACCUSÉS

## CHUCK YEAGER'S AIR COMBAT : LA SIMULATION QUI NE VIEILLIT PAS...

Charles Yeager (Chuck pour les intimes comme moi) est le premier pilote à franchir le mur du son le 14 octobre 1947 à bord du Bell X-1, avion supersonique qui mène à 1079 km/h et à 12 800 mètres d'altitude. Ainsi, le papy en connaît un peu plus sur les avions par rapport à vous.

Vous avez la possibilité de jouer avec des avions réels ou fictifs. Soit vous créez votre propre escadron, soit vous choisissez des avions intéressants ou vous en voyez dans les journaux ou dans ceux de vos ennemis. Ensuite de leur donner soit des missions historiques bien définies concernant la Seconde Guerre Mondiale, la Guerre de Corée, ou la Guerre du Vietnam. Mais la

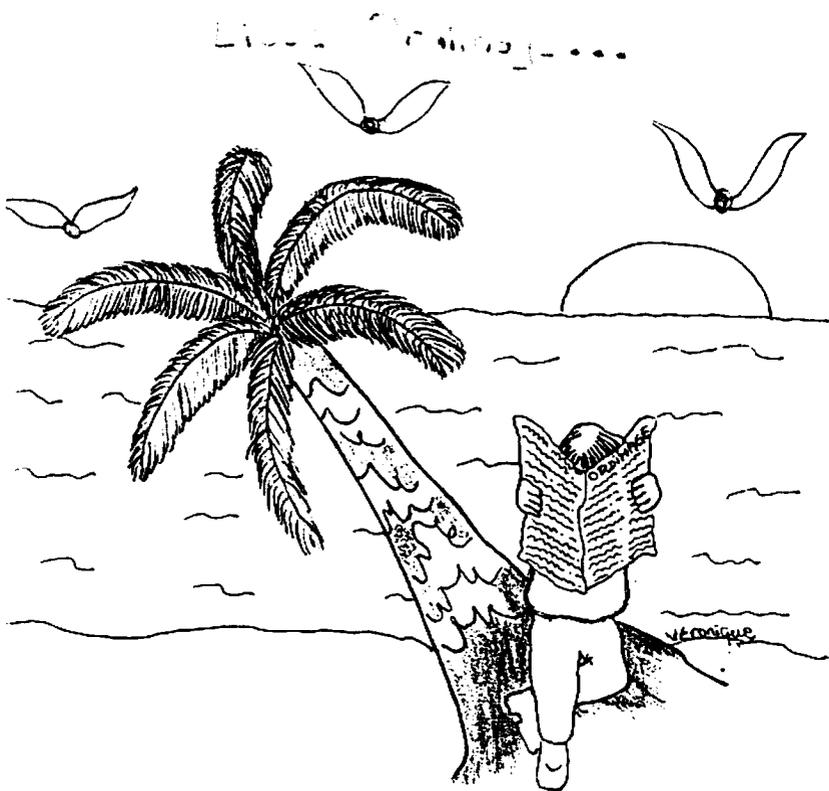
première option est la plus intéressante car nous pouvons dire un avion performant PHANTOM et sélectionner la hélice années 40 et nous avons simplement que et de pose double de nous y compris les munitions in-



vous prenez comme le MIG 29 pour nos avions datant des années 60. C'est tout démonté plus, on dit qu'on a la capacité d'avoir des avions et de

ne pas pouvoir s'écraser la tronche sur le plancher des vaches (Meuh !!). Aussi, on peut observer son avion sous toutes les coutures possibles et inimaginables. Mais, bien qu'on ait des avions assez performants ou des bombardiers, ces appareils datent de plusieurs années (Ils ne sont même plus utilisés par l'Armée de l'Air Américaine). On aurait aimé avoir des F-14 ou des F-16. Néanmoins, ce jeu reste intéressant, surtout avec son option recorder grâce à laquelle vous pouvez voir le film de votre mission sous toutes les coutures avec l'utilisation d'un zoom grossissant jusqu'à 32 fois.

En somme, Chuck Yeager's Air Combat, malgré son grand âge, reste un jeu de référence. A.M.



...et parlez dans un monde de rêves.

# Programmation en Turbo Pascal 6.0

La dernière fois nous avons abordé l'aspect ludique de la programmation en composant un jeu de Mastermind. Nous allons maintenant nous attaquer à un sujet plus sérieux mais non moins intéressant. Vous avez peut-être déjà tenté d'écrire un menu (dans un fichier batch) proposant le chargement de vos principaux logiciels après l'initialisation de l'ordinateur. Néanmoins, vous avez sans doute été confronté à un problème plus ou moins insoluble : comment saisir le choix de l'utilisateur ?

La solution consiste donc à écrire une routine de saisie. En outre le programme que je vous propose affiche la date et l'heure. L'utilisation de cette routine est simple. Une fois copiée et sauvée sous le nom de 'ATTOUCH.PAS' et compilée sous le nom de 'ATTOUCH.EXE', il suffit de l'insérer dans vos fichiers 'batch'. Sa syntaxe est extrêmement simple puisqu'il n'existe aucune condition d'entrée (aucun paramètre).

Cependant, le programme restitue un code ASCII correspondant à la touche frappée que l'on peut utiliser avec la commande DOS ERRORLEVEL. Benoît.

## Le programme :

```
PROGRAM attouche;
{écrit le 22.07.93 par Benoit Riviere}
{©1993 I.E.T. Inf. Marketing}

uses crt, ioin;
var
  p, err : integer;
  ch      : char;

const
  jour : array[0..6] of string[3] =
    ('JAN', 'FEB', 'MAR', 'AVR', 'MAY', 'JUN', 'JUL');
  mois : array[0..11] of string[3] =
    ('JAN', 'FEB', 'MAR', 'AVR', 'MAY', 'JUN', 'JUL',
     'AUG', 'SEP', 'OCT', 'NOV', 'DEC');

procedure affiche;
var
  d, j, m, a, heu, min, sec, sec2 : word;
  [a, heu, min, sec] : string;
begin
  {capture de l'heure et de la date}
  getdate(a, j, d);
  gettime(heu, min, sec, sec2);

  {formatage et transfert en chaîne}
```

```
de la date et de l'heure}
  if d>4 then d:=6;
  d:=d-1;
  m:=m-1;

  str(j, j2);
  if j=1 then j2:=j2+'er';
  str(a, a2);
  str(heu, heu2);
  if heu<10 then heu2:=' '+heu2;
  str(min, min2);
  if min<10 then min2:='0'+min2;
  str(sec, sec22);
  if sec<10 then sec22:='0'+sec22;

  {affichage de l'heure et de la
  date sous le format français}
  gotoxy(55, 1);
  writeln(heu2+' '+min2+' '+sec22+'
'+
  jour[d]+' '+j2+' '+mois[m]+' '+a2);
end; {affiche}

procedure att;
begin
  repeat
    affiche;
    delay(200);
  until keypressed;
  ch:=upcase(readkey);
```







```

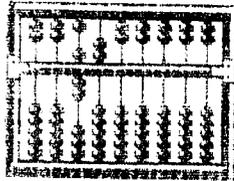
#8000      ....RESET.....
#8000      .
#8000      . adresse implantation #8000 .
#8000      .
#8000      . 27/06/90 Benoit RIVIERE .
#8000      . (c)I.P.L. Informatique .
#8000      .
#8000      .....
#8000      id a,1
#8002 CD0EBC call #bc0e .....MODE 1
#8005 3E00 id a,0
#8007 0601 id b,1
#8009 0E01 id c,1
#800B CD32BC call #bc32 .....INK 0,1,1
#800E 3E01 id a,1
#8010 0618 id b,24
#8012 0E18 id c,24
#8014 CD32BC call #bc32 .....INK 1,24,24
#8017 3E01 id a,1
#8019 CD90BB call #bb90 .....PEN 1
#801C 3E00 loc a,0
#801E CD96BB call #bb96 .....PAPER 0
#8021 0601 id b,1
#8023 0E01 id c,1
#8025 CD39BC call #bc38 .....BORDER 1,1
#8028
#802B 3A0070 id a, #7000:
#802B FE01 op 1 .....: #7000:1 alors CPC6128
#802D DA80B1 jp c,cpc464
#8030 CD90B1 call cpc6128
#8033 C9 ret
#8034 2601 ams128k : id h,1 .....toutes les proc textes
#8036 2E02 id 1,2
#8038 CD75BB call #bb75
#803B 213A81 id hl,txt6
#803E 0642 id b,66
#8040 CDB480 call boucie
#8043 C9 ret
#8044 2601 cpc85 : id h,1
#8046 2E04 id 1,4
#8048 CD75BB call #bb75
#804E 215A81 id hl,txt7
#804E 0650 id b,80
#8050 CDB480 call boucie
#8053 C9 ret
#8054 2601 basic2 : id h,1
#8056 2E07 id 1,7
#8058 CD75BB call #bb75
#805E 216181 id hl,txt8
#805E 0614 id b,20
#8060 CDB480 call boucie
#8063 C9 ret
#8064 2601 ams64k : id h,1
#8066 2E02 id 1,2
#8068 CD75BB call #bb75
#806E 216D60 id hl,txt1
#806E 0642 id b,66
#8070 CDB480 call boucie
#8073 C9 ret
#8074 2601 cpc84 : id h,1
#8076 2E04 id 1,4
#8078 CD75BB call #bb75
#807B 21DD80 id hl,txt2
#807E 0650 id b,80
#8080 CDB480 call boucie
#8083 C9 ret
#8084 2601 loco : id h,1
#8086 2E05 id 1,5
#8088 CD75BB call #bb75
#808B 210481 id hl,txt3
#808E 0650 id b,80
#8090 CDB480 call boucie
#8093 C9 ret
#8094 2601 basic1 : id h,1
#8096 2E07 id 1,7
#8098 CD75BB call #bb75
#809B 212B81 id hl,txt4
#809E 0614 id b,20
#80A0 CDB480 call boucie
#80A3 C9 ret
#80A4 2601 ready : id h,1
#80A6 2E09 id 1,9
#80A8 CD75BB call #bb75
#80AB 213581 id hl,txt5
#80AE 060A id b,10
#80B0 CDB480 call boucie
#80B3 C9 ret
#80B4 7E boucie : id a,(hl)
#80B5 CD5ABB call #bb5a
#80B8 23 inc hl
#80B9 05 dec b
#80BA 10F8 djnz boucie
#80BC C9 ret
#80BD 20416D73 txt1 : " Amstrad 64K Microcomputer (f1)"
#80DD 202431 txt2 : " #1984 Amstrad Consumer Electronics plc"
#8104 202026 txt3 : " and Locomotive Software Ltd."
#812B 2042 txt4 : " BASIC 1.0"
#8135 52 txt5 : "Ready"
#813A .....cpc6128
#813A 20416D73 txt6 : " Amstrad 128K Microcomputer (v3)"
#815A 202431 txt7 : " #1985 Amstrad Consumer Electronics plc"
#8181 2042 txt8 : " BASIC 1.1"
#818B C9 ret
#818C CD6480 cpc464 : call ams64k
#819F CD7480 call cpc84
#8192 CD8460 call loco
#8195 CD9480 call basic1
#8198 CDA480 call ready
#819B C9 ret
#819C CD3460 cpc6128 : call ams128k
#819F CD4480 call cpc85
#81A2 CD8480 call loco
#81A5 CD5480 call basic2
#81A8 CDA480 call ready
#81AB C9 ret

```

# Dossier : les mathématiques et l'informatique

**L'**homme a toujours cherché à fabriquer des outils afin d'alléger ses tâches quotidiennes. C'est ainsi qu'il inventa la charrue, le tracteur... De même, dans des domaines aussi variés que les sciences, la géographie, le dessin... les mathématiques, l'homme s'est empressé de développer des techniques afin de faciliter ses calculs.

Une de ses premières inventions dans le domaine du calcul, est le boulier. Beaucoup plus tard, au XVII<sup>ème</sup> siècle, Blaise Pascal inventa, à l'âge de dix-neuf ans, une calculatrice mécanique capable d'effectuer des opérations aussi élémentaires que l'addition et la multiplication au moyen de d'engrenages, et ceci afin d'automatiser les travaux de son père alors impliqué à la perception.



*boulier - ancêtre de la calculatrice*

Même si nos ordinateurs sont bien loin de cet ancêtre de la calculatrice, il convient tout de même de souligner que l'ordinateur a révolutionné le monde scientifique contemporain. Sans les supercalculateurs (du style CRAY) les cartes météorologiques ne pourraient être établies, les missions spatiales seraient inexistantes... En bref, l'informatique est l'une des bases fondamentales des mathématiques actuelles.



*Blaise Pascal (1623-1662)*

Dans le dossier qui suit, nous aborderons successivement les mathématiques vues par les philosophes (de façon très concise, rassurez-vous), puis nous étendrons plus amplement sur la programmation à l'aide d'exemples divers dans des langages informatiques très couramment utilisés (le Basic, et le Turbo Pascal (le nom de ce langage a été donné en hommage à Blaise Pascal)). Benoît.

$$\forall x \in ]-\infty; +\infty[$$

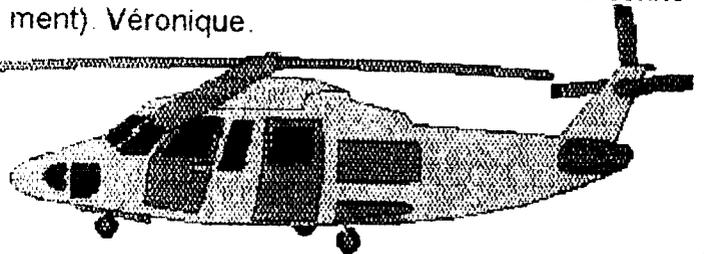
## Mathématiques et Philosophie

Les mathématiques sont, avec la philosophie, l'une des disciplines les plus anciennes. C'est donc tout naturellement qu'elles se sont rejointes. Certains philosophes étaient persuadés que l'on pouvait accéder à la réalité par des examens rationnels, en tant que ceux-ci mettent en oeuvre la pensée.

D'autres confèrent aux sciences mathématiques une place particulière dans le monde. Rappelons ici les études de Blaise Pascal, philosophe et mathématicien qui fit l'expérience au Puy de Dôme afin de prouver l'existence du vide et de l'apesanteur.

Cependant on peut évoquer la philosophie platonienne. Dans la République, l'auteur distingue le monde sensible (composé du monde des images

qui renvoie à l'imagination, et du monde des objets faisant appel à la foi, la croyance), du monde intelligible (divisé en deux sciences : mathématiques, qui portent sur les objets et font appel à la raison, et la dialectique qui renvoie à l'intelligence), montrant ainsi la limite des mathématiques (sciences hypothético-déductives (je déduis quelque chose d'une hypothèse), qui ne se préoccupent guère de la vérité, mais plutôt de la cohérence de son raisonnement). Véronique.



# Trigonométrie

Nous allons tout d'abord parler de trigonométrie. Certains connaissent peut-être des fonctions comme arctangente, cosécante... Cependant, elles sont inexistantes dans beaucoup de langages, en Basic Locomotive (CPC) entre autres. Voici donc ces fonctions.

## Fonctions trigonométriques

| FONCTIONS  |                         | FONCTIONS INVERSES |                                   |
|------------|-------------------------|--------------------|-----------------------------------|
| sinus      | SIN(x)                  | arcsinus           | DEF FNasin(x)=ATN(x/SQR(1-x^2))   |
| cosinus    | COS(x)                  | arccosinus         | DEF FNacos(x)=ATN(SQR(1-x^2)/x)   |
| tangente   | TAN(x)                  | arctangente        | ATAN(x)                           |
| secante    | DEF FNsec(x)=1/COS(x)   | arsecante          | DEF FNasec(x)=ATN(SQR(x^2-1))     |
| cosécante  | DEF FNcosec(x)=1/SIN(x) | arccosécante       | DEF FNacosec(x)=ATN(1/SQR(x^2-1)) |
| cotangente | DEF FNcote(x)=1/TAN(x)  | arccotangente      | DEF FNacotan(x)=ATN(1/x)          |

## Fonctions hyperboliques

| FONCTIONS     |                                                                                                    | FONCTIONS INVERSES |                                   |
|---------------|----------------------------------------------------------------------------------------------------|--------------------|-----------------------------------|
| sinus h.      | DEF FNsinh(x)=(EXP(x)-EXP(-x))/2                                                                   | arcsinus hy        | DEF FNasinh(x)=LOG(x+SQR(x^2+1))  |
| cosinus h.    | DEF FNcosh(x)=(EXP(x)+EXP(-x))/2                                                                   | arccosinus h.      | DEF FNacosh(x)=LOG(x+SQR(x^2-1))  |
| tangente h.   | DEF FNtanh(x)=(EXP(x)-EXP(-x))/(EXP(x)+EXP(-x))<br>OU<br>DEF FNtanh(x)=(1-EXP(-2*x))/(1+EXP(-2*x)) | arctangente h.     | DEF FNatanh(x)=LOG((1+x)/(1-x))/2 |
| secante h.    | DEF FNhsec(x)=1/FNhcosh(x)                                                                         |                    |                                   |
| cosécante h.  | DEF FNhcosec(x)=1/FNhsinh(x)                                                                       |                    |                                   |
| cotangente h. | DEF FNhcote(x)=1/FNhsec(x)                                                                         |                    |                                   |

N.B. : Dans les tableaux précédents, SQR signifie racine carré, '^' se prononce "exposant de" ; x désigne un angle.

**ATTENTION** : Contrairement aux apparences les fonctions LOG(x) et EXP(x) ne désignent en rien les logarithme et exponentiel décimaux, mais les logarithme et exponentiel népériens abrégés en ln(x) et e<sup>x</sup>.

## Fonctions angulaires

Pour traduire de degré en radian : DEF FNradian(x)=x\*PI/180, et de radian en degré : DEF FNdegre(x)=x\*180/PI

NB Pour utiliser ces fonctions, il suffit de faire comme suit :

```
10 DEF FNdegre(x)=180/PI*x
```

```
20 PRINT FNdegre(45)
```

## Routines arithmétiques

Attention, ces routines ne sont utilisables que sur des **Amstrad CPC**. Elles permettent d'avoir accès aux adresses internes afin d'effectuer des calculs par l'intermédiaire de l'assembleur. Pour leur mode d'utilisation, je vous propose de vous référer au dossier "Les adresses des CPC", page 5.

| 464  | 664  | 6128 | EFFET                                                                                                    |
|------|------|------|----------------------------------------------------------------------------------------------------------|
| BD3D | BD5E | BD61 | copie la variable pointée par DE vers la zone pointée par HL.                                            |
| BD40 | BD61 | BD64 | convertit en flottant l'entier contenu dans A et le place dans la zone pointée par DE                    |
| BD43 | BD64 | BD67 | convertit le nombre sur 4 octets pointé par HL en nombre flottant sur 5 octets.                          |
| BD46 | BD67 | BD6A | convertit un flottant situé dans la zone pointée par HL, en entier contenu dans HL.                      |
| BD49 | BD6A | BD6D | idem, mais le résultat se retrouve dans les deux premiers octets pointés par HL.                         |
| BD4C | BD6D | BD70 | fonction FLX.                                                                                            |
| BD4F | BD70 | BD73 | fonction INT.                                                                                            |
| BD55 | BD76 | BD79 | multiplication par 10^A                                                                                  |
| BD58 | BD79 | BD7C | addition de deux nombres réels pointés par HL et DE ; le résultat est placé dans la zone pointée par HL. |
| BD5B | BD7F | BD82 | soustraction de deux nombres réels, idem ci-dessus.                                                      |
| BD61 | BD82 | BD85 | multiplication de deux nombres réels, idem ci-dessus.                                                    |
| BD64 | BD64 | BD88 | division de deux nombres réels, idem ci-dessus.                                                          |
| BD6A | BD8F | BD9E | comparaison de deux nombres réels pointés par HL et DE ; le résultat est mis dans A                      |
| BD6D | BD9E | BD91 | inversion de signe du nombre pointé par HL.                                                              |
| BD70 | BD91 | BD94 | fonction SGN, résultat dans A                                                                            |
| BD73 | BD94 | BD97 | commande DEG ou RAD, selon que A est nul ou positif.                                                     |
| BD76 | BD97 | BD9A | la valeur de PI est placée dans la zone pointée par HL.                                                  |

| 464  | 664  | 6128 | EFFET                                                                                                                      |
|------|------|------|----------------------------------------------------------------------------------------------------------------------------|
| BD79 | BD9A | BD9D | fonction SQRT, appliquée sur la zone pointée par HL, le résultat est placé dans cette même zone.                           |
| BD7C | BD9D | BDA0 | élévation à la puissance : le nombre est pointé par HL, l'exposant pointé par DE, le résultat dans la zone pointée par HL. |
| BD7F | BDA0 | BDA3 | fonction LOG (néperien) : HL pointé sur la zone qui contient le nombre, résultat au même endroit.                          |
| BD82 | BDA3 | BDA6 | fonction LOG10 (décimal), idem ci-dessus.                                                                                  |
| BD85 | BDA6 | BDA9 | fonction EXP, idem ci-dessus.                                                                                              |
| BD88 | BDA9 | BDAC | fonction SIN.                                                                                                              |
| BD8B | BDAC | BDAF | fonction COS.                                                                                                              |
| BD8E | BDAF | BDB2 | fonction TAN.                                                                                                              |
| BD91 | BDB2 | BDB5 | fonction ATN.                                                                                                              |
| BD97 | BDB5 | BDDB | initialisation des trois octets de la zone mémoire utilisée par RND.                                                       |
| BD9A | BDBB | BDBE | calcule RND.                                                                                                               |
| BD9D | BD7C | BD7F | fonction RND.                                                                                                              |
| BDA0 | BD88 | BD8B | recupère RND.                                                                                                              |

## Résolution d'équations du second degré

Le programme suivant a été écrit sur une **Casio fx-8800GC**, mais il doit pouvoir fonctionner sous toute la gamme Casio graphique. Il permet de résoudre des équations du style :  $y=ax^2+bx+c$ . Il est très simple d'emploi, il suffit de taper les chiffres correspondant aux lettres a, b, c. Puis la machine, vous indique le déterminant (DET=), le nombre de racines (aucune, le cas échéant), et leur valeur.

Pour entrer le programme, pressez les touches **MODE+2**, sélectionnez un programme (1, par exemple), puis **EXE**. Le '#' (dièse) correspond en fait au petit triangle, qui ordonne l'affichage, on l'obtient par **SHIFT+PRGM+F5**. "-">" remplace la flèche qui se situe sur le "K" et la virgule.

Pour exécuter le programme, entrez **MODE+1+SHIFT+PRGM+F3+1**. Quand une lettre suivie d'un point d'interrogation apparaît, entrez le chiffre correspondant à la lettre. Quand l'indicateur d'affichage s'affiche (- Disp -), appuyez sur **EXE**.

```

"MODE 1 PRGM 1 F3 1"
"AN: AX 0"
"X" -> A
"X" -> B
"X" -> C
"X" ->
"MODE 1"
"DET=" : "X=" : -B/2A#
D=>=>Goto 1
E=>=>Goto 2
E<=>"ES SOLUS"#
Goto 3
Lb1 1:"1 SOLUS
DOUBLE": "X=" : -B/2A#
Goto 3
Lb1 2:"2 RACINES"
"X'=" : (-B-VD) / 2A#
"X''=" : (-B+VD) / 2A#

```

# Régression linéaire

La droite de régression linéaire est une droite qui passe par le plus près de tous les points d'un nuage de points. Ce genre de calcul permet, si les données sont suffisamment cohérentes, de dresser des prévisions. Le programme suivant calcule donc l'équation de la droite de régression, puis en trace sa représentation, après que vous ayez rentré une à une les coordonnées de chacun des points. Le graphique est réactualisé après l'entrée de chaque point.

Ce programme peut également être utilisé pour calculer l'équation d'une droite reliant deux points, pour ce faire il suffit de ne déclarer que deux points.

```

REM programme de linéa-régression
REM
REM LINSEC v1.0
REM
REM écrit par Robert Fiviere
REM le 15 août 1993
REM pour ordimage
REM
REM (c)1993 L.F.L. Intermaïque
REM
REM déclaration des sous-programmes
DECLARE SUB progmain()
DECLARE SUB init()
DECLARE SUB point()
DECLARE SUB rectangle(x1!, y1!, x2!,
/1!, c AS INTEGER)
DECLARE SUB abscisse()
DECLARE SUB ordonnée()
DECLARE SUB aff()
DECLARE SUB axes()
DECLARE SUB plot(x AS SINGLE, y AS
SINGLE)
DECLARE SUB regress()
REM déclaration des variables globales
DIM SHARED c AS I
DIM SHARED maxx, maxy
DIM SHARED minx, miny
DIM SHARED origx, origy
DIM SHARED debx, deby
DIM SHARED finx, finy
DIM SHARED lxx, lyy
DIM SHARED echx, echy
DIM SHARED nbp
DIM SHARED point
REM programme principal
PROGRAM
END
REM les sous-programmes
SUB axes
  IF maxx = minx OR maxy = miny THEN
EXIT SUB
  CLS 1: REM efface l'écran graphique
  coulax = 7
  echx = lxx / (ABS(maxx) + ABS(minx))
  echy = lyy / (ABS(maxy) + ABS(miny))
  origx = debx + lxx * ABS(minx) /
(ABS(minx) + maxx)
  origy = finy - lyy * ABS(miny) /
(ABS(miny) + maxy)
  REM axe et flèche x
  IF miny < 0 AND maxy > 0 THEN
  PSET (debx, origy), coulax: LINE -
(finx, origy), coulax:
  PSET (finx - 10, origy - 5),
coulax: LINE -(finx, origy), coulax:
  LINE -(finx - 10, origy + 5),
coulax
  END IF
  REM axe et flèche y
  IF minx < 0 AND maxx > 0 THEN
  PSET (origx, deby), coulax: LINE -
(origx, finy), coulax:
  PSET (origx - 5, deby + 10),
coulax: LINE -(origx, deby), coulax:
  LINE -(origx + 5, deby + 10),
coulax
  END IF
  REM graduation x
  FOR e = debx TO finx STEP echx
  PSET (e, origy), coulax
  LINE -(e, origy - 5), coulax
  NEXT

```

```

REM graduation y
FOR e = debx TO finy STEP echy
  PSET (origx, e), coulax
  LINE - (origx + 5, e), coulax
NEXT
END SUB

SUB entien
COLOR 8
FOR poin = 1 TO nbp
  CLS 1: REM efface fenetre texte
  LOCATE 19, 2: PRINT «Entrez le
point numero»: poin:
  INPUT «x,y: », x, y
  denpoin, li = x
  denpoin, li = y
  pixel 0, 0
  m = 0
  IF x > maxx THEN
    maxx = x
    r = 1
  ELSE
    IF x < minx THEN
      minx = x
      r = 1
    END IF
  END IF
  IF y > maxy THEN
    maxy = y
    m = 1
  ELSE
    IF y < miny THEN
      miny = y
      m = 1
    END IF
  END IF
  IF m = 1 THEN
    GOTO 1
  ELSE
    pixel denpoin, li, denpoin, li
  END IF
NEXT
END SUB

SUB gra
axes
FOR a = 1 TO pass
  pixel iaxe, li, axe(a, li)
NEXT
END SUB

SUB nbdon
axes
CLS 1: REM efface la fenetre texte
COLOR 5
LOCATE 19, 3
  INPUT «Combien de points désirez-
vous entrer ? (Entrée pour quitter) :»,
nbp
  IF nbp = 0 THEN END ELSE IF nbp < 2
OR nbp > 50 THEN nbdon
END SUB

SUB pixel (x AS SINGLE, y AS SINGLE)
STATIC
  PSET (origx + x * echx, origy - y *
echy), 14
END SUB

SUB pres
SCREEN 12
VIEW IFINT
VIEW SCREEN (0, 0)-(639, 479), , 0
CLS
COLOR 14
rectangle 10, 10, 620, 35, 10
LOCATE 1, 23
PRINT «R E G R E S S I O N   L I N E
A I R E»
END SUB

SUB prapricip
pres
maxx = 0
maxy = 0
minx = 0
miny = 0
debx = 20: deby = 55
finx = 620: finy = 420
lgx = finx - debx: lgy = finy - deby
VIEW SCREEN (debx - 5, deby - 5)-
(finx + 5, finy + 5), 0, 2
VIEW PRINT 29 TO 30
nbdon
entdon
regress
END SUB

SUB rectangle (x1, y1, x2, y2, c AS
INTEGER) STATIC

```

```

PSET (x1, y1), c
LINE -(x2, y1), c
LINE -(x2, y2), c
LINE -(x1, y2), c
LINE -(x1, y1), c
END SUB

SUB regressea
REM somme des x, y, xx, xy, x^2
somx = 0: somy = 0: somxx = 0: somxy = 0: somx2 = 0
FOR i = 1 TO nbr
  somx = somx + don(i, 1): somy = somy + don(i, 2)
  somxx = somxx + don(i, 1) * don(i, 1)
  somxy = somxy + don(i, 1) * don(i, 2)
  somx2 = somx2 + don(i, 1) * 2
NEXT
REM calcul de a et b
b = (nbr * somxy - somx * somy) / (nbr
  * somxx - somx ^ 2)
a = (somy - b * somx) / nbr

CLS
COLOR 9
LOCATE 18, 3
PRINT «L'équation de la droite de
régression est : «;
COLOR 1
PRINT «y =»; b; «x «;
IF a >= 2 THEN PRINT «+»;
PRINT a
PSET (origx + minx * echx, origy + (b
  * minx + a) * echy), 4
LINE -(origx + maxx * echx, origy -
  (b * maxx + a) * echy), 4
END SUB

```

## Décomposition d'un nombre

La décomposition d'un nombre en facteurs premiers consiste à diviser un nombre en facteurs premiers. La décomposition peut notamment être utilisée pour simplifier des fractions.

```

PROGRAMME DECOMPOSER;

```

```

Ceci est un programme en langage pro-
grammé
écrit par Gabriel Riviere 10/09/93
en Pascal

```

```

CONST

```

```

type
  ent;

```

```

var
  n : longint;
  q, r : extended;
  af : boolean;

```

```

procedure fin;
begin
  textcolor(4);
  writeLn(' ');

```



```

  readln;
  halt;
end; {fin}

```

```

begin
  direct;
  textcolor(4);
  gotoxy(6,7);
  write('Quel nombre désirez-vous dé-
composer ? : ');
  textcolor(3);
  readln(n);
  gotoxy(5,10);
  textcolor(7);
  i:=2;
  repeat
    if n=i then fin;
    q:=n/i;
    if q<i then
      begin
        str(n:0:0,n2);
        writeLn(length(n2):0,' ');
        fin;
      end;
    if q=int(q) then
      begin

```

```

write(i, ' ');
n:=q;
end
else
i:=i+1;
until i=q;
end.

```



## Diviseurs

Ce programme recherche tous les diviseurs d'un nombre.

```

program diviseurs;

```

(Recherche tous les diviseurs d'un nombre)  
 (écrit par Serif Diviere 16/02/93)  
 (pour Ordimage)

```

uses
    crt;

```

```

var
    n,i : integer;
    q : real;
    q1 : string;

```

```

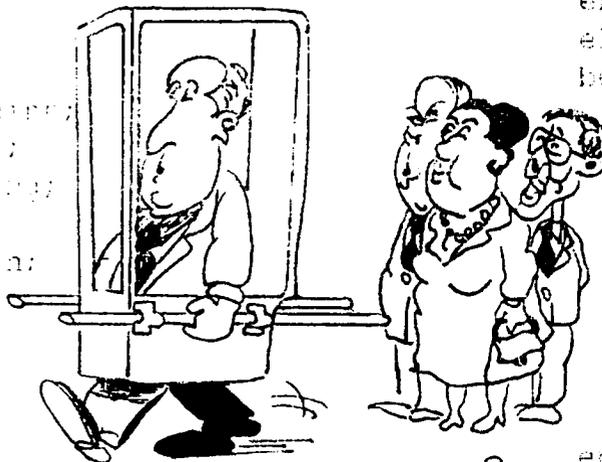
procedure fin;
begin
    readln;
    halt;
end; (*fin*)

```

```

begin
    clrscr;
    textcolor(7);
    write('De quel nombre désirez-vous

```



-Mais que lui arrive-t-il?  
 -Il fait des économies!

```

les diviseurs ? : ');
textcolor(3);
readln(n);
writeLn;
textcolor(7);
i:=1;
repeat
    q:=n/i;
    if i>q then
        begin
            fin;
        end
    else
        begin
            if q=integer(q) then
                begin
                    write(i, ' ');
                    if q<>i then
                        begin
                            str(q:0:0,q2);
                            write(q:length(q2):0, ' ');
                        end;
                end;
            end;
            i:=i+1;
        until i=n;
end.

```

## Nombre premier

Dans les programmes précédents, nous avons fait référence aux nombres premiers. Un nombre premier est un nombre qui n'est divisible que par un et par lui-même. Les premiers sont : 1, 3, 5, 7, 11, 13, 17, 19, 23...

Le programme qui suit vous indique, après saisie d'un nombre de votre choix, si ce dernier est un nombre premier ou si celui-ci est divisible, auquel cas il vous sera indiqué un nombre qui permet de le diviser.



## Quelques trucs sur Amstrad



CALL &BC02 provoque l'initialisation des couleurs (pour de plus amples explications sur cette adresse, regardez page 4 du numéro de juin).

CALL &BB18 ou CALL &BB06 suspendent le déroulement du programme jusqu'à l'appuie d'une touche.

CALL @ initialise l'ordinateur, BASIC fait de même.

## Au secours ! Help ! i Saccato !

Voilà les codes correspondant à la 1<sup>ère</sup> partie des niveaux supérieurs dans le jeu **SABOTEUR II** sur Amstrad CPC. Le code mission 2 à taper dans l'option *Appel mission* est JONIN le code de mission 3 est KIME le quatrième est KUJI KIRI. Quant aux autres codes correspondant aux missions 5, 6, 7, 8, 9 ils sont respectivement SAIMENJI'SU

GENIN MI LU KATA, DIM MAK, et enfin SATORI.

Pour accéder à **ARMY MOVES II**, toujours sur Amstrad, il suffit d'entrer le code 15372.

Pour sortir du premier niveau de **SHORT CIRCUIT**, tapez les touches O, C, E, A, N, néanmoins le second niveau est plus difficile.

—TO BE CONTINUED—

*Au sommaire du prochain numéro*

Vous le verrez bien assez tôt comme ça.  
Date de parution : .../.../1999. (à compléter par vous-même).

