

Les Interruptions des PC

Une interruption est un signal qui interrompt temporairement l'exécution d'un programme afin d'exécuter des instructions spécifiques. Cela permet d'activer des routines système ou des programmes définis par l'utilisateur.

Une table contenant les adresses d'appel est créée en mémoire ; cette table est à rapprocher à la FAT des disques durs ou disquettes. Ainsi chaque interruption renvoie à une adresse. Cette adresse correspond au premier octet d'une routine s'étendant sur plusieurs octets, voire kilo-octets.

L'intérêt de la table est essentiel : elle permet, en cas de changement d'adresse d'appel pour un microprocesseur futur, de conserver une compatibilité ascendante des programmes avec les nouvelles puces. C'est pour cela que les logiciels conçus au départ sur des 8086 ou des 386 fonctionnent encore sur des ordinateurs construits plus récemment. Par ailleurs, les interruptions ne sont pas l'exclusivité des PC, tous les ordinateurs en utilisent quel que soit leur standard (MacIntosh, Amstrad CPC...).

Les interruptions peuvent être activées depuis tout langage, que ce soit l'Assembleur, le Basic, le C, le C++, le TurboPascal... Nous

n'aborderons cette notion qu'à partir de ces deux derniers langages. Pour ceux qui souhaiteraient aborder ce sujet sur Amstrad CPC, je les invite, en complément de ce qui suit, à se reporter au dossier-feuilleton intitulé «Les adresses des CPC» dont le dernier épisode figure page 2 du précédent numéro, et à consulter l'article consacré à la réalisation de RSX (page 54 de ce même numéro 7) et à la programmation 8080 (numéro 7 page 13).

Mise en oeuvre et exemples

Il est bien entendu que les exemples qui suivent ne sont pas applicables aux CPC, même si les principes évoqués demeurent les mêmes.

Pour utiliser une interruption, il faut charger des registres de valeurs, puis appeler l'interruption proprement dite. Le processeur exécute la routine invoquée et retourne les résultats de l'exécution dans des registres. Ces résultats sont de diverses natures ; ils indiquent que la routine s'est déroulée normalement ou non (les registres étaient peut-être chargés de valeurs erronées), ou font état des résultats...

Il faut noter que l'appel d'une interruption peut provoquer un plantage de l'unité centrale (appel mal paramétré, recourt à

une interruption réservée au système, ou encore, altération du contenu de registres nécessaire ultérieurement...). A propos de cette dernière raison, il quelques fois nécessaire de sauvegarder le contenu des registres avant de faire appel à certaines interruptions.

En Turbo C :

```
/*Change la couleur du pourtour de
l'écran*/

#include <stdio.h>
#include <dos.h>
#include <conio.h>

static union REGS regs;

void main (void)
{
    regs.x.ax=0x1001;
    regs.h.bh=0x04;
    int86(0x10, &regs, &regs);
    getch();
}
```

En TurboPascal :

```
(Change la couleur du pourtour de l'écran)

uses dos, crt;

var
    reg : registers;

begin
    reg.ax := $1001;
    reg.bh := $04;
    intr($10, reg);
    readln;
end.
```

L'exécution de ces listings change la couleur du pourtour de l'écran par l'intermédiaire de l'interruption \$1001 (cf. la

description de celle-ci dans le paragraphe "Exemples d'interruption").

Gestion de la souris grâce aux interruptions (unité Turbopascal).

```
unit souris;
{unité de gestion de la souris}
{écrit par Benoît Rivière}
{extrait de TPDOSEXT}

{(c)1995, 1994, 1993 I.P.L.
Informatique}

interface

uses crt, dos, printer, graph;

var
    existesouris : boolean;

type
{déclaration de l'architecture
de l'objet souris}
    souris = object
        procedure Init;
        procedure Montre;
        procedure Cache;
        procedure LimiteX(x1, x2 :
word);
        procedure LimiteY(y1, y2 :
word);
        procedure Posit(sx, sy :
word);
        function X : integer;
        function Y : integer;
        function Press : boolean;
```

```

end;

implementation
{variables privées}
var
  soux, souy      : word;

{procédures souris}
procedure souris.init;
{initialise la souris}
var
  reg : registers;
begin
  reg.ax:=$0;
  intr($33,reg);
  existesouris:=(reg.ax<>0);
  if not existesouris then exit;
  reg.ax:=7;
  reg.cx:=0;
  reg.dx:=getmaxx;
  intr($33,reg);
  reg.ax:=8;
  reg.cx:=0;
  reg.dx:=getmaxy;
  intr($33,reg);
end; {souris.init}

procedure souris.montre;
{montre le curseur de la souris}
var
  reg : registers;
begin
  if not existesouris then exit;
  reg.ax:=$1;
  intr($33,reg);
end; {souris.montre}

procedure souris.cache;
{cache le curseur de la souris}
var
  reg : registers;
begin
  if not existesouris then exit;
  reg.ax:=$2;
  intr($33,reg);
end; {souris.cache}

procedure souris.limitex(x1,x2 :
word);
{limite le déplacement de la
souris sur l'axe horizontal}
var
  view : viewporttype;
  reg : registers;
begin
  if not existesouris then exit;
  reg.ax:=$7;
  reg.cx:=x1;
  reg.dx:=x2;
  intr($33,reg);
end; {souris.limitex}

procedure souris.limitey(y1,y2 :
word);
{limite le déplacement de la
souris sur l'axe vertical}
var
  view : viewporttype;
  reg : registers;
begin
  if not existesouris then exit;
  reg.ax:=$8;

```

```

    reg.cx:=y1;
    reg.dx:=y2;
    intr($33,reg);
end; {souris.limitey}

procedure souris.posit(sx,sy :
word);
{positionne le curseur de la
souris au point de coordonnées X
et Y}
var
    view : viewporttype;
    reg : registers;
begin
    if not existesouris then exit;
    getviewsettings(view);
    reg.ax:=$4;
    reg.cx:=sx;
    reg.dx:=sy;
    intr($33,reg);
end; {souris.posit}

function souris.x : integer;
{donne la coordonnée X indiquant
la position de la souris}
var
    view : viewporttype;
    reg : registers;
begin
    if not existesouris then exit;
    getviewsettings(view);
    reg.ax:=$3;
    intr($33,reg);
    x:=reg.cx-view.x1;
end; {souris.x}

function souris.y : integer;
{donne la coordonnée Y indiquant
la position de la souris}
var
    view : viewporttype;
    reg : registers;
begin
    if not existesouris then exit;
    getviewsettings(view);
    reg.ax:=$3;
    intr($33,reg);
    y:=reg.dx-view.y1;
end; {souris.y}

function souris.press : boolean;
{PRESS est TRUE si le bouton
droite ou le bouton gauche de la
souris est pressé}
var
    view : viewporttype;
    reg : registers;
begin
    if not existesouris then
begin
        press:=FALSE;
        exit;
    end;
    getviewsettings(view);
    reg.ax:=5;
    reg.bx:=0;
    intr($33,reg);
    press:=reg.ax<>0;
    soux:=souris.x;
    souy:=souris.y;
end; {souris.press}

begin
end.

```

```

program ex_souris;
uses crt,souris;
var
  mouse : souris;
begin
  mouse.init;
  mouse.montre;
  readln;
end.

```

L'unité Souris offre de nouvelles procédures et fonctions au TurboPascal. Notez que ces nouvelles instructions sont extraites de TPDOSEXT dont le listing complet a été publié dans le précédent numéro. Si vous avez déjà compilé TPDOSEXT, il est inutile de saisir l'unité Souris, remplacez `uses crt, souris;` par `uses crt, tpdosex;` dans le programme EX_SOURIS.

RAPPEL : Pour utiliser une unité, il faut, après l'avoir saisie, la compiler (Compile sur disque), puis exécuter un programme, par exemple EX_SOURIS (Run).

Exemples d'interruptions

Afin que vous puissiez vous exercer, je vous livre quelques interruptions.

Les interruptions suivantes seront décrites comme suit :

- numéro d'interruption ;
- description succincte de son utilité ;
- registres à charger ;
- résultats sur les registres le cas échéant ;
- notes diverses.

Interruptions Vidéo

INT 10 - Change les couleurs du fond et du pourtour.

AH=0Bh ;

BH=00h ;

BL=code de la couleur à affecter au fond et à la bordure de l'écran (en mode texte, seul le bord est modifié).

INT 10 - Affiche un point (PUTPIXEL en TurboPascal).

AH=0Ch ;

BH=numéro de la page vidéo ;

AL=couleur ;

CX=colonne ;

DX=ligne.

Notes :

- ne fonctionne qu'en mode graphique.
- BH est ignoré si le mode vidéo utilisé ne supporte qu'une seule page.

INT 10 - Lit la couleur d'un point (GETPIXEL en TurboPascal).

AH=0Dh ;

BH=numéro de la page graphique ;

CX=colonne ;

DX=ligne.

Résultat :

AL=couleur du point désigné.

Notes :

- ne fonctionne qu'en mode graphique.
- BH est ignoré si le mode graphique courant ne supporte qu'une seule page vidéo.

Interruptions Disque

INT 21 - Délivre l'espace disque libre.

AH=36h ;

DL=numéro du lecteur (00h : lecteur par défaut, 01h : lecteur A:...).

Résultat :

AX=FFFFh si le lecteur est invalide
simon

AX=nombre de secteurs par clusters (un cluster est un ensemble de secteurs, et constitue une «unité de mesure» en quelque sorte).

BX=nombre de clusters libres ;

CX= nombre d'octets par secteur ;

Bibliographie

Les interruptions énumérées dans cet article sont extraites de la disquette intitulée **Dos Interrupt** au contenu impressionnant équivalent à plus de 1000 pages papier. Toutes les interruptions sont énumérées, expliquées, analysées, décortiquées quasi-exhaustivement. Je vous conseille son acquisition surtout si vous envisagez la programmation en assembleur, ou si vos programmes font appel à des routines assembleur. Dos Interrupt est un freeware

DX=nombre total de clusters sur le support (disquette ou disque dur).

Notes :

- l'espace libre en octets se calcule avec la formule suivante : $AX * BX * CX$.

- l'espace total en octets se calcule avec la formule suivante : $AX * CX * DX$.

- les clusters perdus (défectueux) sont considérés comme étant occupés.

INT 21 : délivre le nom du répertoire courant.

AH=47h ;

DL=numéro du lecteur (00h=lecteur courant, 01h=A:, etc.) ;

DS:SI -> buffer pour le nom du chemin.

Résultat :

CF à zéro si Ok.

AX=0100h

(domaine public), fréquemment remis à jour, disponible, auprès de nombreux distributeurs (consultez les magazines d'informatique), pour un prix modique (moins de 50 FF). Attention, les données contenues sur cette disquette sont en anglais.

Quoique je ne l'aie jamais consultée, je vous invite à lire **La bible du PC**, éditée par Micro Application, qui traite semble-t'il très bien ce sujet à en croire les articles élogieux parus dans diverses revues. Benoit.

Dossier : les mathématiques et l'informatique

Ce dossier fait suite au deux premières parties publiées dans les numéros précédents. Benoît.

Résolution de systèmes d'équations

Le programme suivant se propose de résoudre des systèmes de trois équations à trois inconnues sur calculatrices CASIO FX 8800GC. Un programme similaire à celui-ci a déjà été publié (page 12 ORDIMAGE n°5) en version QBasic et Basic CPC ; il ne résolvait que des systèmes de deux équations à deux inconnues, par la méthode de Cramer.

Définition du problème

Le système d'équations doit être de la forme :

$$\begin{cases} (1) & ax+by+cz=d \\ (2) & ex+fy+gz=h \\ (3) & ix+jy+kz=l \end{cases}$$

Afin d'illustrer mon propos, j'utiliserai l'exemple :

$$\begin{cases} (1) & 05x+3y+8z=100 \\ (2) & 15x+6y-4z=-20 \\ (3) & 25x-9y+7z=500 \end{cases}$$

Les paramètres (a, b, c,...) sont stockés dans les variables Z[1] à Z[30].

Méthode de résolution du problème

La méthode la plus facile à programmer est la formule de Cramer. Cette méthode consiste à disposer la première partie des égalités du système dans une matrice, puis d'en calculer le déterminant.

$$\text{dét} = \begin{vmatrix} a & b & c \\ e & f & g \\ i & j & k \end{vmatrix}$$

$$\text{dét} = \begin{vmatrix} 5 & 3 & 8 \\ 15 & 6 & -4 \\ 25 & -9 & 7 \end{vmatrix}$$

Ensuite, on calcule les déterminants X, Y et Z.

$$\det_X = \begin{vmatrix} d & b & c \\ h & f & g \\ l & j & k \end{vmatrix} \qquad \det_X = \begin{vmatrix} 100 & 3 & 8 \\ -20 & 6 & -4 \\ 500 & -9 & 7 \end{vmatrix}$$

$$\det_Y = \begin{vmatrix} a & d & c \\ e & h & g \\ i & l & k \end{vmatrix} \qquad \det_Y = \begin{vmatrix} 5 & 100 & 8 \\ 15 & -20 & -4 \\ 25 & 500 & 7 \end{vmatrix}$$

$$\det_Z = \begin{vmatrix} a & b & d \\ e & f & h \\ i & j & l \end{vmatrix} \qquad \det_Z = \begin{vmatrix} 5 & 3 & 100 \\ 15 & 6 & -20 \\ 25 & -9 & 500 \end{vmatrix}$$

Méthode de calcul des déterminants

Pour déterminer la valeur des déterminants, on utilisera la règle de Sarrus. Elle permet de calculer des déterminants de matrices d'ordre 3. Sa mise en oeuvre est simple : on dispose à droite de la matrice, ses deux premières colonnes, puis on additionne les produits des diagonales gauche-droite, et on soustrait les produits des diagonales droite-gauche. Pour la première matrice, cela donne :

$$\begin{vmatrix} a & b & c \\ e & f & g \\ i & j & k \end{vmatrix} \begin{vmatrix} a & b \\ e & f \\ i & j \end{vmatrix}$$

$$\text{Dét} = afk + bgi + cej - cfj - agj - bek$$

$$\begin{vmatrix} 5 & 3 & 8 \\ 15 & 6 & -4 \\ 25 & -9 & 7 \end{vmatrix} \begin{vmatrix} 5 & 3 \\ 15 & 6 \\ 25 & -9 \end{vmatrix}$$

$$\text{Dét} = 5*67 + 3*(-4)*25 + 8*15*(-9) - 8*6*25 - 5*(-4)*(-9) - 3*15*7 = -2865.$$

Cette matrice est stockée de Z[13] à Z[27].

Afin de rendre le programme plus court et plus rapide, on peut factoriser cette formule :

$$\text{Dét} = a(fk - gj) + b(gi - ek) + c(ej - fi)$$

Si Dét=0, il n'y a pas de solution.

Il faut procéder de la même façon avec les déterminants X, Y et Z.

La solution du système s'obtient ainsi (Cramer) :

$$x = \frac{\text{dét } X}{\text{dét}} \quad y = \frac{\text{dét } Y}{\text{dét}} \quad z = \frac{\text{dét } Z}{\text{dét}}$$

Le programme

PROG F : programme principal (environ 180 pas).

```
"RESOLUTION"
"SYSTEME"
"EQUATIONS"
"NOUVEAU SYSTEME ?"
"(1->NON)"
tapez <1>+<Ent>, si un système a déjà été
précédemment saisi et que vous désirez en
obtenir la solution, sinon entrez <0> pour
entrer un nouveau système et en déterminer le
résultat.
?->A
A=1⇨Goto 1
Defm 27
4*3+(3+2)*3+1=27

Entrée des données
1->A
Lbl 0
Int((A-1)/4)=(A-1)/4⇨"NV
EQUATION"
?->Z[A]
A+1->A
A<=12⇨Goto 0

Boucle de calcul des déterminants
Lbl 1
1->E
Lbl 2
Prog 0
D->F[E]
E+1->E
E<=4⇨Goto 2

Affichage des solutions ou des messages
indiquant l'absence de solution
F[1]<>0⇨Goto 3
F[2]=0⇨F[3]=0⇨F[4]=0⇨"INFINIT
E DE SOLUTION"
F[2]<>0⇨F[3]<>0⇨F[4]<>0⇨"PAS
DE SOLUTION"
Goto 4
Lbl 3
"X":F[2]/F[1]->X#
"Y":F[3]/F[1]->Y#
"Z":F[4]/F[1]->Z#
Lbl 4
```

PROG O : Calcul du déterminant d'une matrice carrée d'ordre 3 par la méthode de Sarrus (environ 313 pas).

```
"CALCUL DET"
1->A
13->B
Lbl 0
Z[A]->Z[B]
Int((A-3)/4)<>(A-3)/4⇨Z[A]
->Z[B+3]

Int((A-3)/4)=(A-3)/4⇨B+2->B
Int((A-3)/4)=(A-3)/4⇨A+1->A
A+1->A
B+1->B
A<=12⇨Goto 0
E=1⇨Goto 1
Z[4]->Z[11+E]
```

```

Z[8]→Z[16+E]          13→A
Z[12]→Z[21+E]         0→D
E=4⇨Goto 1             Lbl 2
Z[4]→Z[14+E]          A<=15⇨D+Z[A]*Z[A+6]*Z[A+12]→D
Z[8]→Z[19+E]          A=>15⇨D-Z[A]*Z[A+4]*Z[A+8]→D
Z[12]→Z[24+E]         A+1→A
Lbl 1                   A<=17⇨Goto 2

```

Notes :

- Pour entrer un nouveau programme, il faut presser **MODE+2**, puis sélectionner un programme (F et O, ne sont que des exemples). Avant de saisir les listings, vérifiez que vous disposez de suffisamment de mémoire.
- Les notes en italique insérées dans le listing ont pour but d'expliquer le fonctionnement du programme et ne doivent pas être recopiées.
- Les '#' (dièse), insérés dans le listing, correspondent au petit triangle, qui ordonne l'affichage du résultat du calcul en cours. Pour obtenir ce petit triangle, faites : **SHIFT+PRGM+F5**.
- Pour exécuter le programme, entrez **MODE+1+SHIFT+PRGM+F3+F**, ensuite suivez le déroulement du programme.

Nombres premiers

La fonction TurboPascal suivante, bâtie à partir du programme des pages 22 et 23 du numéro 6, répond 'TRUE' (vrai) si le nombre proposé (n) est premier, sinon elle renvoie 'FALSE' (faux) ainsi que le nombre qui l'a divisé.

```

function EstPremier(n : real ; var d : real) : boolean;
(Entrée : n => le nombre à tester ;
 Sortie : d => le nombre qui a divisé n si ce dernier n'est pas
 premier)
var
  i : real;
  q : real;
begin
  i:=2;
  repeat
    q:=n/i;
    if q<i then
      begin
        estpremier:=true;
        d:=0;
        exit;
      end
    else
      begin
        if q=int(q) then
          begin
            estpremier:=false;
            d:=i;
            exit;
          end
        else
          inc(i);
        end;
      until i=1;
    end; {EstPremier}

```

Fonctions diverses sous TP

Certaines fonctions font cruellement défaut au TurboPascal telles exposant, racine de x base y, logarithmes et exponentiels de base a. Ces quelques fonctions vont remédier à ces oublis.

```
function exposant(x, y : real) : real;
begin
  if x=0 then exposant:=0
  else
    exposant:=abs(x)/x*exp(y*ln(abs(x)));
end; {exposant}
```

```
function RacineBaseY(x, y : real) : real;
{calcule la racine de x en base y -> racinebasey(2,3) renvoie la
racine cubique de 2}
begin
  if x=0 then racinebasey:=0
  else
    racinebasey:=exp((1/y)*ln(x));
end; {racinebasey}
```

```
function LogA(a : integer ; x : real) : real;
begin
  loga:=ln(x)/ln(a);
end; {LogA}
```

```
function ExpA(a : integer ; x : real) : real;
begin
  expa:=exp(x*ln(a));
end; {ExpA}
```

Traitement des dates

Le listing suivant, sous forme d'une fonction TurboPascal, détermine le jour de la semaine (Lundi, Mardi...) d'une certaine date, traitement utile dans des programmes affichant la date.

```
function JourSemaine(jour, mois, annee : integer) : integer;
var
  siecle, an, js : integer;
begin
  if mois<3 then
    begin
      inc(mois,10);
      dec(annee);
    end
  else
    dec(mois,2);
```

```

siede:=annee div 100;
an:=annee mod 100;
js:=(((26*mois-2) div 10)+jour+an+(an div 4)+(siede div 4)-
(2*siede)) mod 7;
if js<0 then JourSemaine:=js+7
else JourSemaine:=js;
end; {JourSemaine}

```

Si vous faites : `writeln(joursemaine(10,12,1995));` vous obtiendrez le numéro du jour de la semaine. Pour obtenir directement le nom de ce jour, on pourrait déclarer un tableau contenant ces jours, et les afficher :

```

const
  jours : array[1..7] of string[12] =
    'lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi',
    'samedi', 'dimanche';
...
begin
  writeln(jours[joursemaine(10,12,1995)]);
end.

```

Une dernière chose : une année bissextile est divisible par 4, donc `Annee MOD 4` doit être égale à 0, si l'année considérée est bissextile.

Calcul de points dérivés et de tangentes

Soient $f(x)$ une fonction, $f'(x)$ sa dérivée, et $T(x)$ la tangente à la fonction $f(x)$ au point x_0 . $y_0=f(x_0)$, y_0 étant l'ordonnée de la fonction $f(x)$ à l'abscisse x_0 (x_0 est donc connue).

Exemple : $f(x)=x^2+x+5$ et $x_0=15$.

Calcul d'un point dérivé

Le point dérivé d'une fonction se calcule ainsi :

$$f'(x_0) = \frac{f(x_0 + c) - f(x_0 - c)}{2c}$$

c est une constante qui doit prendre une valeur la petite possible, afin que les résultats obtenus soient des plus précis. Dans le programme, on prendra $c=0,00001$.

Exemple : $f(x)=2x+1$; $f(15)=2*15+1=31$.

Calcul de la tangente d'une fonction en un point

$T(x_0)$ est définie par $y-f(x_0)=f'(x_0)(x-x_0)$.

$$y=f'(x_0)(x-x_0)+f(x_0)$$

$$y=f'(x_0)x-f'(x_0)x_0+f(x_0)$$

Exemple : $T(x)=(2*15)(x-15)+(15^2+15+5)$; $T(x)=31x-220$.

Le programme

Avant d'utiliser le programme, il faut mémoriser la fonction $f(x)$ à étudier. Tapez votre fonction, (x^2+3) par exemple, puis pressez la séquence de touches suivante : Shift+FMem puis STO (dans le menu) 1, la fonction est stockée dans f1.

Pour mettre en oeuvre le programme, suivez les mêmes instructions que celles décrites dans le paragraphe Résolution de systèmes d'équations.

PROG D :

"DERIVEE/TANGENTE"

"A="

"EN UN POINT"

B#

"ENTREZ X":?->A

"B="

A+.00001->X

A->X

"DERIVEE F'(X)="

B*A+f1#

(B-f1)/0.00002->B#

A propos de ce dernier calcul, référez-vous à la note concernant la tangente.

"TANGENTE"

"Y=AX+B"

Notes :

- On obtient f1 par Shift+FMem+Fn (menu) 1.

- Tangente : $y=f'(x_0)x-f'(x_0)x_0+f(x_0)$; $A=f'(x_0)$ et $B=f(x_0)x_0+f(x_0)$ donc $y=Ax+B$.

Divers

Diagonale d'un carré

La diagonale d'un carré de côté de longueur a vaut $a\sqrt{2}$.

Distance entre deux points

Soient deux points $A(x,y)$ et $B(x',y')$, la distance les séparant est égale à racine carré de $((x'-x)^2+(y'-y)^2)$.

Milieu d'un bi-point

Soient $A(x,y)$ et $B(x',y')$, le point médian M a pour coordonnées $(x+x')/2$ et $(y+y')/2$.

Droites perpendiculaires

Deux droites sont perpendiculaires si $\vec{AB} \cdot \vec{CD} = 0$, donc $xx'+yy'=0$, avec $\vec{AB}(x,y)$ et $\vec{CD}(x',y')$.

Droites parallèles

Deux droites sont parallèles si $\det(\vec{AB}, \vec{CD}) = 0$, $\det = xy' - yx'$.

Maths sur CPC

Math+ est un logiciel éducatif, programmé en Basic CPC, qui devrait aider des élèves de 6ème et de 5ème à la résolution de petits exercices, notamment grâce au calcul de PGCD, de PPCM, d'identités remarquables... Ce programme ne demande qu'à être complété, par exemple par des listings publiés précédemment.

```

10 'MATH+ : BENOIT RIVIERE
20 MODE 2
30 MEMORY 42000
40 IF PEEK(&8000) <> &2A THEN GOSUB 2320
50 ON ERROR GOTO 2300
60 '
70 FOR i=40 TO 0 STEP -1:OUT &BC00,1:OUT &BD00,1:FOR y=1 TO 10:NEXT:NEXT
80 CLEAR
90 EVERY 50,0 GOSUB 2290
100 ON BREAK GOSUB 2290
110 ON ERROR GOTO 2300
120 CLS
130 FOR i=0 TO 15:INK i,0:NEXT
140 WINDOW#0,2,79,2,24
150 PAPER#1,1:FEN#0,0:CLS#1
160 PEN 1:PAPER 0
170 CLS
180 RESTORE 200

```

```

190 LOCATE 7,22:ODL:PRINT CHR$(164)"1988 & 1989  BENOIT RIVIERE":UST:LOCATE 20,5:PRINT"Mathématiques Assistées par
Ordinateur":ODT:LOCATE 30,4:PRINT"M A T H +":UST
200 DATA " 1.....RESOLUTION D'UN SYSTEME D'EQUATIONS", " 2.....CALCUL DES NOMBRES PREMIERS DE 1 à 1000", " 3.....RECONNAITRE
SI UN NOMBRE EST PREMIER", " 4.....DECOMPOSITION D'UN NOMBRE"
210 DATA " 5.....CALCUL DE PGCD & PPCM", " 6.....RECONNAITRE LES DIVISEURS D'UN NOMBRE"
220 DATA " 7.....SIMPLIFICATION DE FRACTION", " 8.....CALCUL DU PERIMETRE D'UN STADE", " 9.....IDENTITES REMARQUABLES"
230 FOR I=7 TO 15:READ TIT$:LOCATE 12,I:PRINT TIT$:NEXT
240 PLOT 20,3&5:DRAW 620,3&5:DRAW 620,100:DRAW 20,100:DRAW 20,3&5
250 INK 1,2&:INK 0,0:INK 2,14:INK 3,15:BORDER 0
260 FOR i=0 TO 40:OUT &BC00,1:OUT &BD00,1:FOR y=1 TO 10:NEXT:NEXT
270 i=1:RESTORE 2090:GOSUB 20&0
280 '
290 p#=INKEr$: IF p#="" GOTO 300
300 LOCATE 1,20:GOSUB 2240
310 IF p#="/" THEN STOP
320 REP=INSTR(" 123456789",p#)
330 IF REP=0 THEN 280
340 IF p#="/" THEN STOP
350 ON rep-1 GOTO 370,1150,790,1700,910,1540,1410,620,1880
360 GOTO 280
370 loc=7:tid$="RESOLUTION D'UN SYSTEME DE DEUX EQUATIONS PAR LA METHODE DE CRAMER":GOSUB 2040
380 ' RESOLUTION DE SYSTEMES D'EQUATIONS
390 '
400 PRINT
410 LOCATE 20,7:PRINT"LE SYSTEME EST SOUS LA FORME:"
420 LOCATE 30,8:PRINT"(A*X)+(B*Y)=C"
430 LOCATE 30,9:PRINT"(D*X)+(E*Y)=F"
440 WINDOW #0,5,30,12,17
450 LOCATE 1,1:INPUT" COEFFICIENT A :",a
460 LOCATE 1,2:INPUT" COEFFICIENT B :",b
470 LOCATE 1,3:INPUT" COEFFICIENT C :",c
480 LOCATE 1,4:INPUT" COEFFICIENT D :",d
490 LOCATE 1,5:INPUT" COEFFICIENT E :",e
500 LOCATE 1,6:INPUT" COEFFICIENT F :",f
510 WINDOW #0,2,79,2,24
520 dt=a*e-b*d
530 IF dt=0 THEN GOTO 590
540 x=(c*e-b*f)/dt
550 y=(a*f-c*d)/dt
560 PRINT
570 LOCATE 15,18:PRINT"SOLUTION:x=";x;"ET Y=";y
580 GOTO 610
590 PRINT
600 LOCATE 20,18:PRINT"DETERMINANT NUL, JE NE SAIS PAS RESOUDRE!"
610 GOSUB 1100:GOTO 370
620 loc=13:tid$="C A L C U L  D U  P E R I M E T R E  D ' U N  S T A D E":GOSUB 2040
630 PLOT 300,115:DRAW 300,250:DRAW 400,250:DRAW 400,115:DRAW 300,115
640 FOR a=1 TO 180 STEP 2.5:DEG:PLOT 350,250:PLOT 350+50*COS(a),250+50*5IN(a):NEXT
650 TAG:MOVE 290,200:PRINT"H";:MOVE 350,110:PRINT"D";:TAGOFF
660 WINDOW 5,30,10,13
670 LOCATE 3,1:INPUT"Largeur D=",d
680 p=(d*PI)/2
690 LOCATE 3,3:INPUT"Longueur H=",h
700 WINDOW 2,79,2,24
710 p=p+2*h+d
720 LOCATE 6,14:PRINT"Perimètre :":p
730 GOSUB 1100:GOTO 620
740 CLS
750 PRINT"DONNEZ LA VALEUR DU RAYON"

```

```

760 INPUT r
770 PRINT r#2#PI
780 GOSUB 1100:GOTO 740
790 '
800 loc=5:tid#="RECONNAITRE SI UN NOMBRE EST PREMIER":GOSUB 2040
810 WINDOW#0,4,77,9,10
820 LOCATE 24,1:INPUT "Quel nombre voulez-vous ";n
830 WINDOW#0,2,79,2,24
840 i=2
850 quotient=n/i
860 IF quotient<i THEN LOCATE 28,13:PRINT n;" est premier":GOTO 900
870 IF quotient=INT(QUOTIENT) THEN LOCATE 20,13:PRINT n;" n'est pas premier (divisible par";i;")":GOTO 900
880 i=i+1
890 GOTO 850
900 GOSUB 1100:GOTO 790
910 '
920 loc=17:tid#="CALCUL DE P.G.C.D. ET P.P.C.M.":GOSUB 2040
930 WINDOW #0,10,77,7,10
940 LOCATE 13,1:INPUT "Quel est le premier nombre ? : ",a
950 IF a<=0 THEN 940
960 LOCATE 13,3:INPUT "Quel est le deuxième nombre ? : ",b
970 IF b<=0 THEN 960
980   t=a
990   f=b
1000 IF a<b THEN 1060
1010 WINDOW 2,79,2,24
1020 LOCATE 22,12:PRINT "PGCD ("; t;","; f;)"= ";a
1030 LOCATE 22,13:PRINT "PPCM ("; t;","; f;)"= "; t* f/a
1040 IF a=1 THEN LOCATE 19,15:PRINT "c;"; f;"sont premiers entre eux."
1050 GOTO 1080
1060 IF a>b THEN a=a-b ELSE b=b-a
1070 GOTO 1000
1080 GOSUB 1100:GOTO 910
1090 END
1100 WINDOW#0,2,79,2,24:FOR i=1 TO 500:NEXT:LOCATE 12,22:PRINT"V O U L E Z - V O U S   R E C O M M E N C E R   ? (N/O)"
1110 p#=INKEY$:IF p#=""THEN 1110
1120 p#=UPPER$(p#):IF p#="O"THEN RETURN
1130 IF p#="N" THEN FOR i=40 TO 0 STEP -1:OUT &BC00,1:OUT &BD00,1:FOR y=1 TO 10:NEXT:NEXT:GOTO 80
1140 GOTO 1110
1150 '
1160 loc=7:tid#="CALCUL DES NOMBRES PREMIERS DE 1 à 1000":GOSUB 2040
1170 WINDOW#0,5,77,7,20
1180 LOCATE 17,6:PRINT"P A T I E N C E ,   J E   C A L C U L E "
1190 DIM a(1000),b(200)
1200 FOR i=2 TO 1000
1210   a(i)=0
1220 NEXT i
1230   c=0
1240   d=SQR(1000)
1250 FOR b=2 TO 1000
1260 IF a(b)<0 THEN 1330
1270   c=c+1
1280   b(c)=b(c)+b
1290 IF b>d THEN 1330
1300 FOR j=b TO 1000 STEP b
1310   a(j)=-1
1320 NEXT j
1330 NEXT b
1340 LOCATE 1,3

```



```

1350 FOR i=1 TO c
1360 PRINT b(i);
1370 NEXT i
1380 ERASE a,b
1390 WINDOW#0,2,79,2,24
1400 GOSUB 1100:GOTO 1150
1410 loc=15:tid$="S I M P L I F I C A T I O N   D E   F R A C T I O N":GOSUB 2040
1420 WINDOW 20,77,7,9
1430 LOCATE 5,1:PRINT"FRAC TION X/Y  ":LOCATE 19,1:INPUT"NUMERATEUR  :",c1:LOCATE 19,2:INPUT"DENOMINATEUR  :",d1
1440 IF c1=0 OR d1=0 THEN 1420
1450 WINDOW 2,79,2,24
1460 gg=c1:hh=d1
1470 IF gg=hh THEN 1500
1480 IF gg<hh THEN hh=hh-gg:GOTO 1470
1490 gg=gg-hh:GOTO 1470
1500 e1=c1/gg:f1=d1/gg
1510 PRINT:IF c1=e1 AND d1=f1 THEN LOCATE 25,12:PRINT"FRAC TION IRREDUCTIBLE":GOTO 1530
1520 LOCATE 22,12:PRINT"RESULTAT=";E1;"/";F1
1530 GOSUB 1100:GOTO 1410
1540 '
1550 loc=5:tid$="R E C O N N A I T R E   L E S   D I V I S E U R S   D ' U N   N O M B R E":GOSUB 2040
1560 WINDOW 10,77,8,10:LOCATE 5,1:INPUT"De quel nombre voulez-vous les diviseurs ?:",n
1570 WINDOW 2,79,2,24
1580 i=1
1590 LOCATE 10,19:PRINT"Appuyez sur <ESPACE> pour noter les nombres"
1600 WINDOW#0,20,60,11,18:CLS
1610 quotient=n/i
1620 IF i>quotient THEN WINDOW#0,2,79,2,24:GOSUB 1100:GOTO 1540
1630 IF INKEY$=" " THEN FOR y=1 TO 600:NEXT:GOTO 1630
1640 IF quotient<>INT(quotient) THEN GOTO 1680
1650 PRINT TAB(5);i;:PRINT" ";
1660 IF quotient=i THEN 1680
1670 PRINT quotient
1680 i=i+1
1690 GOTO 1610
1700 pas=0
1710 loc=17:tid$="D E C O M P O S I T I O N   D ' U N   N O M B R E":GOSUB 2040
1720 WINDOW#0,5,77,7,8
1730 LOCATE 5,1:INPUT"Quel nombre voulez-vous décomposer ? :",n
1740 WINDOW#0,5,77,9,11
1750 PRINT n;:PRINT" = ";
1760 i=2
1770 IF n=1 THEN WINDOW#0,2,79,2,24:GOTO 1840
1780 quotient=n/i
1790 IF quotient<i THEN PRINT n:WINDOW#0,2,79,2,24:GOTO 1840
1800 IF quotient<>INT(quotient) THEN i=i+1:GOTO 1780
1810 PRINT i;:PRINT" ";
1820 n=quotient
1830 GOTO 1770
1840 IF pas=1 THEN GOSUB 1100:GOTO 1700 ELSE LOCATE 7,12:PRINT" Voulez-vous comparer ces résultats ? (O/N)";
1850 repc$=INKEY$:IF repc$="" THEN 1850
1860 repc$=UPPER$(repc$):IF repc$="O" THEN PRINT:PRINT:PRINT:WINDOW#0,5,77,14,15:LOCATE 5,14:INPUT "Quel nombre voulez-vous
décomposer ? :",n:WINDOW#0,5,77,17,20:quotient=0:pas=1:GOTO 1750 ELSE IF repc$="N" THEN GOSUB 1100:GOTO 1700
1870 GOTO 1850
1880 loc=15:tid$="L E S   I D E N T I T E S   R E M A R Q U A B L E S":GOSUB 2040
1890 LOCATE 3,6:PRINT"(A+B)^2= A^2 + 2AB + B^2 "
1900 LOCATE 30,6:PRINT"(A-B)^2= A^2 - 2AB + B^2 "
1910 LOCATE 56,6:PRINT"(A-B)(A+B)= A^2 - B^2"
1920 WINDOW 5,77,9,9:LOCATE 10,1:INPUT"Valeur A :",a

```



```

2320 'attrb
2330 MEMORY &7FFF
2340 RESTORE 2440
2350 a=&B000:FOR j=100 TO 400 STEP 10:s=0
2360 FOR j=1 TO 10
2370 READ x#:xx=VAL("&"+x#):POKE a,xx:s=s+xx:a=a+1
2380 NEXT
2390 READ xx:IF s<>xx THEN PRINT"erreur ligne":END
2400 NEXT
2410 CALL &B000
2420 RETURN
2430 'DH, DL, DT, ST
2440 DATA 2a,d4,bd,22,2b,80,01,13,80,21,829
2450 DATA 0f,80,c3,d1,bc,00,00,00,00,21,768
2460 DATA 80,c3,2d,80,c3,34,80,c3,3b,80,1253
2470 DATA c3,42,80,44,cc,44,c8,44,d4,53,9:92
2480 DATA d4,00,c3,ff,ff,21,49,80,22,d4,1397
2490 DATA bd,c9,21,97,80,22,d4,bd,c9,21,1371
2500 DATA ce,80,22,d4,bd,c9,2a,2b,80,22,1217
2510 DATA d4,bd,c9,f5,e5,1e,02,cd,a5,bb,1665
2520 DATA 3e,19,cd,5a,bb,3e,ff,cd,5a,bb,1368
2530 DATA 0e,08,06,04,56,af,cb,4b,20,0e,617
2540 DATA cb,22,cb,22,cb,22,cb,22,18,04,976
2550 DATA cb,27,cb,27,cb,22,30,02,f6,03,1020
2560 DATA 10,f4,cd,5a,bb,23,0d,20,db,3e,1103
2570 DATA ff,e1,d5,e5,cd,2a,80,e1,d1,1d,1760
2580 DATA c8,3e,09,cd,5a,bb,f1,24,e5,18,1283
2590 DATA b6,f5,2d,e5,1e,02,cd,a5,bb,3e,1352
2600 DATA 19,cd,5a,bb,3e,ff,cd,5a,bb,06,1312
2610 DATA 04,cb,4b,20,04,23,23,23,23,7e,584
2620 DATA cd,5a,bb,cd,5a,bb,23,10,f6,3e,1323
2630 DATA ff,e1,d5,e5,cd,2a,80,e1,2c,d1,1775
2640 DATA 1d,c8,f1,e5,18,ce,f5,2d,e5,1e,1478
2650 DATA 04,cd,a5,bb,3e,19,cd,5a,bb,3e,1192
2660 DATA ff,cd,5a,bb,cb,43,28,04,23,23,1121
2670 DATA 23,23,0e,04,06,04,56,7b,fe,03,564
2680 DATA 30,08,cb,22,cb,22,cb,22,cb,22,1004
2690 DATA af,cb,27,cb,27,cb,22,30,02,f6,1192
2700 DATA 03,10,f4,cd,5a,bb,cd,5a,bb,23,1262
2710 DATA 0d,20,d9,3e,ff,e1,d5,e5,cd,2a,1493
2720 DATA 80,e1,d1,cb,43,c2,21,81,2c,7b,1355
2730 DATA fe,03,c2,2e,81,3e,09,cd,5a,bb,1179
2740 DATA 2d,24,f1,1d,c8,f5,e5,18,9e,00,1207

```

Dossier Trucs et Astuces sur Amstrad CPC Complément

Le programme intitulé «KELCPC», publié dans le précédent numéro, permettait de connaître la version de l'ordinateur utilisé, ce qui est utile notamment pour les programmes en langage machine ou faisant appel à de telles routines. Exécuté sur un 6128+ (nouvelle gamme), celui-ci renvoie i=4, nous en déduisons donc qu'un CPC 464+ renverrait 3.

Afin de tenir compte des précédentes remarques, il serait donc souhaitable de modifier les lignes 110 et 120 comme suit et d'ajouter une ligne 130 :

```

110 if (i=0) or (i=3) then cpc=464 else if i=1 then cpc=664
else if (i=2) or (i=4) then cpc=6128
120 print"Vous avez un CPC" cpc;
130 if i>2 then print"+" else print

```

LE PASSAGER DU TEMPS

Voici la suite (mais pas la fin) de la solution de ce jeu d'aventure. Bon voyage, Laurent.

PAVE NUMERIQUE

Certaines touches du pavé numérique sont affectées d'un verbe ce qui évite d'avoir à les taper.

2 : discute avec
3 : navigue
4 : allume
5 : éteinds
6 : charge

7 : prends
8 : examine
9 : sauve
0 : recopie le dernière phrase tapée

LE DEBARRAS :

- Prends clef à douille (sans faire attention au commentaire du chat...).

LA CHAMBRE :

En sortant de la chambre et après avoir lu la lettre, retourner dehors et examiner la boîte à lettre. Réponse : «le facteur est passé... circuit marque BZX».

- Prends circuit.

LE PETIT SALON :

Il faut être muni des allumettes et de la feuille sentant le citron.

- Allume allumettes
- Chauffe feuille

Un message apparaît alors (lire vite car il ne reste pas longtemps).

- Allume interrupteur
- Bas

- Allume torche
Rendez-vous dans la pièce du groupe électrogène.



LE GROUPE ELECTROGENE :

- 1°) Regarde groupe
- 2°) Branche fil
- 3°) Regarde sous groupe
- 4°) Prends écrou
- 5°) Branche fil
- 6°) Huile écrou
- 7°) Branche fil

8°) Démarre groupe

Une fois le groupe électrogène prêt à l'emploi, remontez à l'étage.

LES W.C.

- 1°) Assis
- 2°) Examine Blouse
On lit 3=4.
- 3°) Sud

Dirigez-vous vers la bibliothèque.

LA BIBLIOTHEQUE

Assurez-vous que l'interrupteur du petit salon est éteint avant d'entreprendre ce qui suit :

- 1°) Ouest
- 2°) Allume ordinateur
- 3°) Charge démarre
- 4°) Eteinds ordinateur
- 5°) Est
- 6°) Tourne tableau
- 7°) Prends livre de Dumas (l'indice «3=4» signifiait les trois mousquetaires qui sont en fait 4, histoire écrite par Alexandre Dumas)
- 8°) Nord
- 9°) Haut

10°) Tapez C O D E

DANS LE LABORATOIRE

- 1°) Tire manette
- 2°) Prends carte CPU
- 3°) Bas
- 4°) Sud

LE PETIT SALON

- Allume interrupteur
Allez à l'établi.
- Soude BZX
Remontez au petit salon.
- Eteinds interrupteur (A ne pas oublier !).

LA BIBLIOTHEQUE

Recommencez les mêmes opérations que précédemment (Allume ordinateur...).

DANS LE LABORATOIRE

- 1°) Enfiche carte CPU
- 2°) Tire manette
- 3°) Pousse manette

Et c'est parti pour le grand voyage !!!
A suivre...

