

```

10 ' *****
20 ' ***
30 ' *** Programme de creation graphique pour ***
40 ' *** AMSTRAD PCV 8256 ou PCV 8512 ***
50 ' ***
60 ' *** Ce programme utilise l'extension G.S.X ***
70 ' *** fournie sur une disquette CPM ***
80 ' ***
90 ' ***
100 ' ***
110 ' ***
120 ' *****
130 ' ***
140 ' *** Programme ecrit par Jean-Marie SCAYA ***
150 ' ***
160 ' *****
170 '
180 '
190 '
200 GOSUB 20000 ' *** Initialisation
210 cls$=CHR$(27)+"E"+CHR$(27)+"H"
220 PRINT cls$
230 INPUT "Sur quel peripherique voulez vous travailler (1,21,22) ";periph$
240 periph=VAL(periph$)
250 IF periph<>1 AND periph<>21 AND periph<>22 THEN 220
260 GOSUB 20200 ' *** Ouverture de la station graphique
270 GOSUB 1000 ' *** Dessine le titre
280 GOSUB 2000 ' *** Traces possibles
290 GOSUB 4000 ' *** Trace un histogramme
300 GOSUB 5000 ' *** Trace un graphique
310 GOSUB 6000 ' *** Trace des "camemberts"
320 GOSUB 20400 ' *** Fermeture
330 PRINT cls$
340 END
350 '
500 ' **** Vide tampon et efface l'ecran ou fait un saut de page ****
510 '
520 GOSUB 20500 ' *** Indispensable sur imprimante
530 IF periph<10 THEN WHILE INKEY$="" : WEND : PRINT cls$; ELSE LPRINT CHR$(12);
540 RETURN
550 '
1000 ' **** Dessine le titre ****
1010 '
1020 EntParam(0)=0 : GOSUB 21100 ' *** Couleur Ligne Invisible
1030 EntCoord(0)=0 : EntCoord(1)=32767 : EntCoord(2)=32767 : EntCoord(3)=32767
1040 GOSUB 22000 ' *** Trace une ligne
1050 EntParam(0)=1 : GOSUB 21100 ' *** Couleur Ligne visible
1060 EntCoord(1)=100 : GOSUB 20800 ' *** Taille du texte
1070 EntCoord(0)=5000 : EntCoord(1)=25000
1080 message$="G.S.X" : GOSUB 20600 ' *** Affiche un texte
1090 EntCoord(1)=20 : GOSUB 20800 ' *** Taille du texte
1100 EntCoord(0)=5000 : EntCoord(1)=20000
1110 message$="Programme de demonstration" : GOSUB 20600 ' *** Affiche un texte
1120 EntCoord(0)=5500 : EntCoord(1)=15000
1130 message$="pour PCV 8256 et 8512" : GOSUB 20600 ' *** Affiche un texte
1140 EntCoord(1)=40 : GOSUB 20800 ' *** Taille du texte
1150 EntCoord(0)=0 : EntCoord(1)=10000
1160 message$="Jean-Marie SCAYA" : GOSUB 20600 ' *** Affiche un texte
1170 GOSUB 500
1180 RETURN
1190 '
2000 ' **** Differents traces possibles ****
2010 '
2020 EntCoord(0)=1000 : EntCoord(1)=30000
2030 message$="Dessiner un damier sur imprimante" : GOSUB 20600 ' *** Affiche un
n texte

```

```

2040 EntParam(0)=3 : GOSUB 21900 ' *** Mode Ecriture
2050 EntParam(0)=1 : GOSUB 21600 ' *** Mode de Remplissage
2070 FOR i=0 TO 4
2080   EntCoord(0)=i*2000+1000 : EntCoord(1)=18000
2090   EntCoord(2)=i*2000+2000 : EntCoord(3)=28000
2100   GOSUB 22400 ' *** Rectangle Plein
2110   EntCoord(0)=1000 : EntCoord(1)=i*2000+18000
2120   EntCoord(2)=11000 : EntCoord(3)=i*2000+19000
2130   GOSUB 22400 ' *** Rectangle Plein
2140 NEXT i
2150 EntParam(0)=1 : GOSUB 21900 ' *** Mode Ecriture
2160 EntCoord(0)=1000 : EntCoord(1)=18000 : EntCoord(2)=11000 : EntCoord(3)=28000
2170 GOSUB 22300 ' *** Rectangle Vide
2180 x=25000 : y=23000 : rayon=7500 : GOSUB 22100 ' *** Trace un cercle
2190 EntCoord(0)=22000 : EntCoord(1)=25000
2200 message$="Il est pas beau" : GOSUB 20600 ' *** Affiche un texte
2210 EntCoord(1)=20 : GOSUB 20800 ' *** Taille du texte
2220 EntCoord(0)=21000 : EntCoord(1)=22000
2230 message$="mon cercle ?" : GOSUB 20600 ' *** Affiche un texte
2240 EntCoord(0)=1000 : EntCoord(1)=15000
2250 message$="Différents remplissages sur imprimante" : GOSUB 20600 ' *** Affi
che un texte
2255 EntCoord(1)=10 : GOSUB 20800 ' *** Taille du texte
2260 EntParam(0)=2 : GOSUB 21600 ' *** Mode de Remplissage
2270 FOR i=0 TO 5
2280   EntParam(0)=i+1 : GOSUB 21700 ' *** Hachures
2290   EntCoord(0)=i*4000 : EntCoord(1)=11000
2300   EntCoord(2)=i*4000+2500 : EntCoord(3)=13500
2310   GOSUB 22400 ' *** Rectangle Plein
2320 NEXT
2330 EntCoord(0)=25000 : EntCoord(1)=12000
2340 message$="Mode Rempl 2" : GOSUB 20600 ' *** Affiche un texte
2350 EntParam(0)=3 : GOSUB 21600 ' *** Mode de Remplissage
2360 FOR i=0 TO 5
2370   EntParam(0)=i+1 : GOSUB 21700 ' *** Hachures
2380   EntCoord(0)=i*4000 : EntCoord(1)=7500
2390   EntCoord(2)=i*4000+2500 : EntCoord(3)=10000
2400   GOSUB 22400 ' *** Rectangle Plein
2410 NEXT
2420 EntCoord(0)=25000 : EntCoord(1)=8500
2430 message$="Mode Rempl 3" : GOSUB 20600 ' *** Affiche un texte
2440 EntCoord(1)=20 : GOSUB 20800 ' *** Taille du texte
2450 EntCoord(0)=1000 : EntCoord(1)=5000
2460 message$="Différents pointeurs et lignes" : GOSUB 20600 ' *** Affiche un t
exte
2470 FOR i=1 TO 5
2480   EntParam(0)=1 : GOSUB 21200 ' *** Style des Points
2490   EntCoord(0)=i*1000 : EntCoord(1)=2000
2500   Controle(0)=7 : controle(1)=1 : GOSUB 20100 ' *** Commande GSX
2510 NEXT
2520 FOR i=1 TO 6
2530   EntParam(0)=1 : GOSUB 21000 ' *** Style des lignes
2540   EntCoord(0)=i*3500+7000 : EntCoord(1)=4000
2550   EntCoord(2)=i*3500+10000 : EntCoord(3)=4000
2560   GOSUB 22000 ' *** Trace une ligne
2570   EntCoord(0)=i*3500+7000 : EntCoord(1)=500
2580   EntCoord(2)=i*3500+7000 : EntCoord(3)=4000
2590   GOSUB 22000 ' *** Trace une ligne
2600   EntCoord(0)=i*3500+7000 : EntCoord(1)=500
2610   EntCoord(2)=i*3500+10000 : EntCoord(3)=4000
2620   GOSUB 22000 ' *** Trace une ligne
2630 NEXT i
2640 GOSUB 500
2650 RETURN
2660 '
3000 ' Trace un repère

```

```

5010 '
5020 EntParam(0)=1 : GOSUB 21000 ' *** Style des lignes
5030 EntCoord(0)=0 : EntCoord(1)=2000 : EntCoord(2)=32700 : EntCoord(3)=2000
5040 GOSUB 22000 ' *** Trace une ligne
5050 EntCoord(0)=1000 : EntCoord(1)=0 : EntCoord(2)=1000 : EntCoord(3)=32700
5060 GOSUB 22000 ' *** Trace une ligne
5070 EntCoord(0)=1100 : EntCoord(1)=400
5080 message$=" Janv Fevr Mars Avri Mai Juin Juill Aout Sept Oct Nov
Dec" : GOSUB 20600 ' *** Affiche un texte
5090 EntParam(0)=2 : GOSUB 21600 ' *** Mode de Remplissage
5100 EntParam(0)=3 : GOSUB 21000 ' *** Style des lignes
5110 FOR k=1 TO 4
5120 message$=MID$(STR$(k*5),2)
5130 EntCoord(0)=100 : EntCoord(1)=k*5*1500+2000 : GOSUB 20600 ' *** Affiche
un texte
5140 NEXT
5150 FOR k=1 TO 20
5160 EntCoord(0)=1000 : EntCoord(1)=k*1500+2000
5170 EntCoord(2)=32000 : EntCoord(3)=k*1500+2000
5180 GOSUB 22000 ' *** Trace une ligne
5190 NEXT
5200 RETURN
5210 '
4000 ' Trace un histogramme
4010 '
4020 GOSUB 3000 ' *** trace repere
4030 EntParam(0)=2 : GOSUB 21700 ' *** Hachures
4040 FOR i=1 TO 12
4050 EntCoord(0)=i*2400-1400 : EntCoord(1)=2000 : EntCoord(2)=i*2400-700
4060 EntCoord(3)=INT(RND(1)*20*1500)+2000
4070 GOSUB 22400 ' *** Rectangle Plein
4080 NEXT
4090 EntParam(0)=4 : GOSUB 21700
4100 FOR i=1 TO 12
4110 EntCoord(0)=i*2400-700 : EntCoord(1)=2000 : EntCoord(2)=i*2400
4120 EntCoord(3)=INT(RND(1)*20*1500)+2000
4130 GOSUB 22400 ' *** Rectangle Plein
4140 NEXT
4150 EntParam(0)=5 : GOSUB 21700 ' *** Hachures
4160 FOR i=1 TO 12
4170 EntCoord(0)=i*2400 : EntCoord(1)=2000 : EntCoord(2)=i*2400+700
4180 EntCoord(3)=INT(RND(1)*20*1500)+2000
4190 GOSUB 22400 ' *** Rectangle Plein
4200 NEXT
4210 GOSUB 500
4220 RETURN
4230 '
5000 ' Trace un graphique
5010 '
5020 GOSUB 3000 ' *** trace repere
5025 FOR k=1 TO 3
5030 EntParam(0)=k : GOSUB 21000 ' *** Style des lignes
5040 EntCoord(0)=1500 : EntCoord(1)=INT(RND(1)*20*1500)+2000
5050 FOR i=2 TO 12
5060 EntCoord(2)=i*2400-900 : EntCoord(3)=INT(RND(1)*20*1500)+2000
5070 GOSUB 22000 ' *** Trace une ligne
5080 EntCoord(0)=EntCoord(2) : EntCoord(1)=EntCoord(3)
5090 NEXT i
5100 NEXT k
5110 GOSUB 500
5120 RETURN
5130 '
6000 ' Dessine des camemberts (ou des tartes)
6010 '
6020 RESTORE 6400
6030 EntParam(0)=2 : GOSUB 21600 ' *** Mode de Remplissage

```

```

6040 EntParam(0)=1 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6050 EntParam(0)=2 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6060 EntParam(0)=3 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6070 EntParam(0)=4 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6080 EntParam(0)=3 : GOSUB 21600 ' *** Mode de Remplissage
6090 EntParam(0)=3 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6100 EntParam(0)=5 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6110 EntParam(0)=6 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6120 EntParam(0)=0 : GOSUB 21600 : GOSUB 6500 ' *** Lecture DATA
6130 EntParam(0)=1 : GOSUB 21600 : GOSUB 6500 ' *** Lecture DATA
6140 EntParam(0)=2 : GOSUB 21600 ' *** Mode de Remplissage
6150 EntParam(0)=2 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6160 EntParam(0)=4 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6170 EntParam(0)=3 : GOSUB 21600 ' *** Mode de Remplissage
6180 EntParam(0)=3 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6190 EntParam(0)=6 : GOSUB 21700 : GOSUB 6500 ' *** Lecture DATA
6200 EntCoord(1)=30 : GOSUB 20800 ' *** Taille du texte
6210 EntCoord(0)=5000 : EntCoord(1)=4000
6220 message$="Bons graphismes avec" : GOSUB 20600 ' *** Affiche un texte
6230 EntCoord(1)=40 : GOSUB 20800 ' *** Taille du texte
6240 EntCoord(0)=10000 : EntCoord(1)=2000
6250 message$="G.S.X" : GOSUB 20600 ' *** Affiche un texte
6260 GOSUB 500
6270 RETURN
6400 DATA 5000,8000,27000,0,130,5000,8000,27000,130,90
6410 DATA 5000,8000,27000,220,80,5000,8000,27000,300,60
6420 DATA 5000,22000,27000,20,160,5000,22000,27000,180,140
6430 DATA 5000,23000,27000,320,60,16300,16350,6000,0,40
6440 DATA 16300,16350,6000,40,30,16300,16350,6000,70,45
6450 DATA 16300,16350,6000,115,30,16300,16350,6000,145,25
6460 DATA 16300,16350,6000,170,10
6470 '
6500 ' Lecture des donnees en DATA pour camemberts
6510 '
6520 READ rayon,x,y,origine,angle
6530 GOSUB 22500 ' *** Arc de cercle
6540 RETURN
6550 '
20000 ' Dimensionnement des tableaux
20010 '
20020 DEFINT c,e,g,l,i,j,k
20030 DIM controle(5),EntParam(79),EntCoord(148),LecParam(79),LecCoord(148)
20040 gsx=&H30
20050 RESTORE 20060
20060 DATA &hC5,&hD1,&h0E,&h73,&hC3,&h05,&h00
20070 FOR i=0 TO 6 : READ c : POKE gsx+i,c : NEXT i
20080 RETURN
20090 '
20100 ' Commande GSX
20110 '
20120 CALL gsx(gsx,gsx,Controle(0),EntParam(0),EntCoord(0),LecParam(0),LecCoord(0))
20130 RETURN
20140 '
20200 ' Ouverture (code 1)
20210 '
20220 EntParam(0)=periph
20230 RESTORE 20240
20240 DATA 1,1,1,1,1,1,0,2,1
20250 FOR i=1 TO 9 : READ EntParam(i) : NEXT i
20260 controle(0)=1 : controle(3)=10 : GOSUB 20100 ' *** Commande GSX
20270 Xcoeff=1 : Ycoeff=1
20280 x=LecParam(0)*LecParam(3) : Y=LecParam(1)*LecParam(4)
20290 IF y>x THEN Ycoeff=y/x
20300 IF x>y THEN Xcoeff=x/y ' *** de facon a tracer des cercles ronds
20310 RETURN

```

```

20320 '
20400 ' Fermeture (code 2)
20410 '
20420 Controle(0)=2 : Controle(3)=0 : GOSUB 20100 ' *** Commande GSX
20430 RETURN
20440 '
20500 ' Vide tampon (code 4) indispensable sur imprimante
20510 '
20520 controle(0)=4 : controle(3)=0 : GOSUB 20100 ' *** Commande GSX
20530 RETURN
20540 '
20600 ' Affiche un texte ( code 8 ) les coordonnees de position sont
20610 ' dans EntCoord(0) et EntCoord(1)
20620 ' le message dans message$
20630 FOR i=0 TO LEN (message$)-1
20640 EntParam(1)=ASC(MID$(message$,i+1,1))
20650 NEXT i
20660 controle(0)=8 : controle(1)=1 : controle(5)=1
20670 controle(3)=LEN(message$)
20680 GOSUB 20100 ' *** Commande GSX
20690 RETURN
20700 '
20800 ' Taille du texte ( code 12 ) grosseur (10 a 120 ) dans EntCoord(1)
20810 '
20820 EntCoord (0)=0 : controle(0)=12 : Controle(1)=0 : Controle(3)=0
20830 GOSUB 20100 ' *** Commande GSX
20840 RETURN
20850 '
21000 ' Style des lignes (code 15) style (1 a 6) dans EntParam(0)
21010 '
21020 controle(0)=15 : controle(1)=0 : Controle(3)=0
21030 GOSUB 20100 ' *** Commande GSX
21040 RETURN
21050 '
21100 ' Couleur de ligne (code 17) couleur (0 ou 1) dans EntParam(0)
21110 '
21120 controle(0)=17 : controle(1)=0 : controle(3)=1
21130 GOSUB 20100 ' *** Commande GSX
21140 RETURN
21150 '
21200 ' Style des points ( code 18) style (1 a 5) dans EntParam(0)
21210 '
21220 controle(0)=18 : controle(1)=0 : controle(3)=0
21230 GOSUB 20100 ' *** Commande GSX
21240 RETURN
21250 '
21600 ' Mode de Remplissage ( code 23 ) mode (0 a 3) dans EntParam(0)
21610 '
21620 Controle(0)=23 : Controle(1)=0 : controle(3)=0
21630 GOSUB 20100 ' *** Commande GSX
21640 RETURN
21650 '
21700 ' Hachures (code 24) hachures (1 a 6) dans EntParam(0)
21710 '
21720 Controle(0)=24 : Controle(1)=0 : Controle(3)=0
21730 GOSUB 20100 ' *** Commande GSX
21740 RETURN
21750 '
21900 ' Mode Ecriture ( code 32 ) mode (1 a 4) dans EntParam(0)
21910 '
21920 Controle(0)=32 : Controle(1)=0 : Controle(3)=0
21930 GOSUB 20100 ' *** Commande GSX
21940 RETURN
21950 '
22000 ' Trace une ligne (coordonnees dans EntCoord(0 a 4))
22010 '
22020 controle(0)=6 : controle(1)=2: controle(3)=0 : controle(5)=1
22030 GOSUB 20100 ' *** Commande GSX
22040 RETURN
22050 '
22100 ' trace un cercle (initialiser rayon,x et y )

```



```

22110 '
22120 EntCoord(0)=INT(rayon/xcoeff)+x : EntCoord(1)=y
22130 controle(0)=7 : Controle(1)=1 : Controle(5)=1 : GOSUB 20100 ' *** Commande
    GSX
22140 FOR i=0 TO 360
22150     EntCoord(2)=INT(COS(1/57.295)*rayon/Xcoeff)+x
22160     EntCoord(3)=INT(SIN(1/57.295)*rayon/Ycoeff)+y
22170     GOSUB 22000 ' Trace une ligne
22180     EntCoord(0)=EntCoord(2) : EntCoord(1)=EntCoord(3)
22190 NEXT i
22200 RETURN
22210 '
22300 ' trace un rectangle vide (code 10) coord dans EntCoord(0 a 3)
22310 '
22320 controle(0)=10 : controle(1)=2 : controle(5)=1
22330 GOSUB 20100 ' *** Commande GSX
22340 RETURN
22350 '
22400 ' Trace un rectangle plein ( code 11 ) Coord dans EntCoord(0 a 3)
22410 '
22420 controle(0)=11 : controle(1)=2 : controle(5)=1
22430 GOSUB 20100 ' *** Commande GSX
22440 RETURN
22450 '
22500 ' trace un arc de cercle ( initialiser rayon,x,y,origine,et angle)
22510 ' les variables origine et angle sont des angles exprimes en degres
22520 '
22530 pas=angle/28
22540 EntCoord(0)=x : EntCoord(1)=y
22550 EntCoord(2)=INT(COS(origine/57.295)*rayon/Xcoeff)+x
22560 EntCoord(3)=INT(SIN(origine/57.295)*rayon/Ycoeff)+y
22570 FOR i=1 TO 28
22580     pointe=INT(origine+i*pas)
22590     EntCoord(i*2+2)=INT(COS(pointe/57.295)*rayon/Xcoeff)+x
22600     EntCoord(i*2+3)=INT(SIN(pointe/57.295)*rayon/Ycoeff)+y
22610 NEXT i
22620 controle(0)=9 : controle(1)=30 : Controle(5)=1
22630 GOSUB 20100 ' *** Commande GSX
22640 RETURN
22650 '

```

G_{SX} - PAS

type

```
chaîne = string [80];  
ctrl   = array [0..5]   of integer ;  
param  = array [0..79]  of integer ;  
coord  = array [0..148] of integer ;  
opt     = string [9] ;
```

var

```
contrôle : ctrl ;           { Definition des tableaux }  
entParam : param ;         { Commandes et Contrôles }  
entCoord : coord ;         { Envoi les parametres }  
LecParam : param ;         { Envoi les coordonnees }  
LecCoord : coord ;         { Recupere parametres }  
Adresses : array [0..4]   of integer ; { Recupere coordonnees }  
                                         { Adresses des tableaux }  
  
coeffX , CoeffY : real ;    { coeff en fonction du peripherique }  
i,j,k : integer ;  
SinCos : array [0..362,0..1] of real ; { Table des sinus en degre }
```

```

sinus : file ;
tablesinus : integer ;

const
    CodeGSX = 115 ;

procedure initialise ;           { Initialise le tableau Adresses }
begin
    tablesinus := 0;
    Adresses [0] := addr (controle [0]) ;
    Adresses [1] := addr (EntParam [0]) ;
    Adresses [2] := addr (EntCoord [0]) ;
    Adresses [3] := addr (LecParam [0]) ;
    Adresses [4] := addr (LecCoord [0]) ;
end;

procedure LectureSinus ;
var
    v : real;
begin
    if (tablesinus)=0 then
    begin
        assign (sinus,'sinus.dat');
        {$i-}
        reset (sinus);
        {$i+}
        if IORESULT<> 0 then
        begin
            for k:=0 to 360 do
            begin
                v:=k/57.295;
                sincos[k,0]:=sin(v);
                sincos[k,1]:=cos(v);
            end;
            assign (sinus,'sinus.dat');
            rewrite (sinus);
            blockwrite (sinus,sincos,trunc(sizeof(sincos)/128));
        end
        else
            blockread (sinus,sincos,trunc(sizeof(sincos)/128));
        close (sinus);
        tablesinus:=1;
    end;
end;

procedure qsx ( a,b,c,d : integer );
begin
    controle [0] := a ;
    controle [1] := b ;
    controle [3] := c ;
    controle [5] := d ;
    bdos (CodeGSX,addr(adresses[0]));
end;

procedure Ouverture (periph : integer ;
                    options : opt );           { Code 1 }
var
    x,y,xx,yy : real ;
begin
    initialise ;
    EntParam [0] := periph ;
    for i:=1 to 9 do
        val (copy(options,i,1),EntParam [i],j);
    gsx (1,0,10,0) ;
    x:=LecParam [0]; xx:=LecParam [3]; x:=x*xx; { Calcule les coefficients }
    y:=LecParam [1]; yy:=LecParam [4]; y:=y*yy; { en fonction du periph }

```



```

coeffX:=1 ;      CoeffY:=1 ;      { pour tracer un cercle }
if y>x then coeffY:=y/x ;      { sur ecran ou imprimante }
if x>y then CoeffX:=x/y ;

```

```
end;
```

```
procedure Fermeture ;      { Code 2 }
```

```
begin
  gsx (2,0,0,0);
end;
```

```
procedure VideTampon ;      { Code 4 }
```

```
begin
  gsx (4,0,0,0);      { Inutile sur ecran mais pas sur imprimante }
end;
```

```
procedure Ligne (Sommets : integer);      { Code 6 }
```

```
begin
  gsx (6,sommets,0,1) ;
end;
```

```
procedure Point (Sommets : integer);      { Code 7 }
```

```
begin
  gsx (7,sommets,0,1) ;
end;
```

```
procedure Texte (message : chaine ; a,b : integer);      { Code 8 }
```

```
begin
  for i:=0 to length(message)-1 do
    EntParam [i]:=ord(copy(message,i+1,1));
  EntCoord [0]:=a;
  EntCoord [1]:=b;
  gsx (8,1,length(message),1) ;
end;
```

```
procedure Polygone (Sommets : integer);      { Code 9 }
```

```
begin
  gsx (9,Sommets,0,1);
end;
```

```
Procedure Rectangle (x1,y1,x2,y2 : integer );      { Code 10 }
```

```
begin
  EntCoord [0] := x1 ; EntCoord [1] := y1 ;
  EntCoord [2] := x2 ; EntCoord [3] := y2 ;
  gsx (10,2,0,1) ;
end;
```

```
Procedure RectPlein (x1,y1,x2,y2 : integer );      { Code 11 }
```

```
begin
  EntCoord [0] := x1 ; EntCoord [1] := y1 ;
  EntCoord [2] := x2 ; EntCoord [3] := y2 ;
  gsx (11,2,0,1) ;
end;
```

```
procedure taille (grosseur : integer);      { code 12 }
```

```
begin
  entcoord[0]:=0;
  entcoord[1]:=grosseur;
  gsx (12,0,0,0);
end;
```

```
procedure couleur (coul: integer);      { code 14 }
```

```
begin
  entparam [0]:=coul ;
  gsx (14,0,0,0);
end;
```

```

procedure StyleLigne (style : integer) ;           { code 15 }
begin
    EntParam [0]:=style ;
    gsx (15,0,0,0);
end ;

procedure CouleurLigne (couleur : integer);         { code 17 }
begin
    EntParam[0]:=couleur;
    gsx (17,0,1,0);
end;

procedure StylePoint (style : integer) ;           { code 18 }
begin
    EntParam [0]:=style ;
    gsx (18,0,0,0);
end;

procedure taillePoint (grosueur : integer);        { code 19 }
begin
    entcoord[0]:=0;
    entcoord[1]:=grosueur;
    gsx (19,0,0,0);
end;

procedure CouleurPoint (couleur : integer);         { code 20 }
begin
    EntParam[0]:=cpuleur;
    gsx (20,0,1,0);
end;

procedure CouleurTexte (couleur : integer);         { code 22 }
begin
    EntParam[0]:=couleur;
    gsx (22,0,1,0);
end;

procedure ModeRemplissage (mode : integer) ;        { code 23 }
begin
    EntParam [0]:= mode ;
    gsx (23,0,0,0);
end;

procedure Hachures (style : integer) ;              { Code 24 }
begin
    EntParam [0]:= style ;
    gsx (24,0,0,0) ;
end;

procedure CouleurSurface (couleur : integer);       { code 25 }
begin
    EntParam[0]:=couleur;
    gsx (25,0,1,0);
end;

procedure ModeEcriture (mode : integer);            { code 32 }
begin
    entparam[0]:=mode ;
    gsx (32,0,0,0);
end;

procedure Plot (x1,y1 : integer) ;                  { Debut Formes predefinies }
begin
    EntCoord [0] := x1 ; EntCoord [1] := y1 ;
    point (1) ;
end;

```

```

Procedure UneLigne (x1,y1,x2,y2 : integer) ;
begin
  EntCoord [0] := x1 ; EntCoord [1] := y1 ;
  EntCoord [2] := x2 ; EntCoord [3] := y2 ;
  ligne (2) ;
end ;

procedure Trace (x1,y1 : integer) ;
begin
  entcoord [2] :=x1 ; EntCoord [3] :=y1 ;
  ligne (2) ;
  entcoord [0] :=x1 ; EntCoord [1] :=y1 ;
end;

procedure cercle (rayon,x,y : integer) ;
begin
  LectureSinus;
  plot (trunc(rayon/coeffx)+x,y);
  for i:=0 to 360 do
    trace (trunc((sincos[i,1]*rayon/coeffx)+x),
           trunc((sincos[i,0]*rayon/coeffy)+y));
  end;

procedure arc (rayon,x,y,origine,angle : integer) ;
var
  pas : real ;
  Pointe : integer ;
begin
  if origine >= 360 then origine:=origine-360 ;
  if angle >= 360 then angle:=angle-360 ;
  LectureSinus;
  pas:=angle/28 ;
  EntCoord [0]:=x; EntCoord [1]:=y;
  entCoord [2]:=trunc((sincos[origine,1]*rayon/coeffx)+x);
  EntCoord [3]:=trunc((sincos[origine,0]*rayon/coeffy)+y);
  for i:=1 to 28 do
    begin
      pointe := trunc (origine+i*pas)+1 ;
      if pointe >= 360 then pointe :=pointe-360;
      entcoord [i*2+2]:= trunc((sincos[pointe,1]*rayon/coeffx)+x);
      entcoord [i*2+3]:= trunc((sincos[pointe,0]*rayon/coeffy)+y);
    end;
  polygon (30);
end;●

```

DEMO - GSX - PAS

```
program DemoGSX;
```

```
($i gsx.pas)
```

```
var note : array [0..12,1..3] of real;  
    periph : integer ;  
    notation : opt;
```

```
procedure TracesPossibles;    { ***** Differents traces possibles ***** }  
begin  
    Texte ('Dessiner un damier sur imprimante',1000,30000);  
    rectangle (1000,18000,11000,28000);  
    ModeEcriture (3);  
    ModeRemplissage(1);  
    for i:=0 to 4 do  
    begin  
        rectplein (i*2000+1000,18000,i*2000+2000,28000);  
        rectplein (1000,i*2000+18000,11000,i*2000+19000);
```

```

end;
modeEcriture(1);
cercle (7500,25000,23000);
texte ('Il est pas beau',22000,25000);
taille (20);
texte ('mon cercle ?',21000,22000);
texte ('Différents remplissages sur imprimante',1000,15000);
taille (10);
ModeRemplissage(2);
for i:=0 to 5 do
begin
    hachures (i+1);
    rectplein (i*4000,11000,i*4000+2500,13500);
end;
texte ('Mode Rempl 2',25000,12000);
ModeRemplissage(3);
for i:=0 to 5 do
begin
    hachures (i+1);
    rectplein (i*4000,7500,i*4000+2500,10000);
end;
texte ('Mode Rempl 3',25000,8500);
taille (20);
texte ('Différents pointeurs et lignes',1000,5000);
for i:=1 to 5 do
begin
    stylePoint(i);
    plot (i*1000,2000);
end;
for i:=1 to 6 do
begin
    StyleLigne(i);
    uneLigne (i*3500+7000,4000,i*3500+10000,4000);
    uneLigne (i*3500+7000,500,i*3500+7000,4000);
    uneLigne (i*3500+7000,500,i*3500+10000,4000);
end;
VideTampon;
if periph<10 then read (k);
end;

```

```

procedure TraceRepere ;
begin
    for i:=1 to 12 do          { tire des valeurs au hasard }
    begin
        note[i,1]:=random*20;
        note[i,2]:=random*20;
        note[i,3]:=random*20;
    end;
    styleLigne (1);
    uneLigne (0,2000,32700,2000);
    uneLigne (1000,0,1000,32700);
    ModeRemplissage(2);
    texte (' Janv  Fevr  Mars  Avri  Mai  Juin  Juill  Aout  '+
           '  Sept  Oct  Nov  Dec',1100,400);
    StyleLigne (3);
    for k:=1 to 4 do
    begin
        str(k*5,notation);
        texte (notation,100,k*5*1500+2000);
    end;
    for k:=1 to 20 do
        uneLigne (1000,k*1500+2000,32000,k*1500+2000);
    end;

```

```

Procedure Histogramme; { ***** Histogramme ***** }
begin

```

```

TraceRepere ;
Hachures (2);
for i:=1 to 12 do
    rectplein(i*2400-1400,2000,i*2400-700,trunc(note[i,11*1500)+2000);
Hachures (4);
for i:=1 to 12 do
    rectplein(i*2400-700,2000,i*2400,trunc(note[i,21*1500)+2000);
Hachures (5);
for i:=1 to 12 do
    RectPlein(i*2400,2000,i*2400+700,trunc(note[i,31*1500)+2000);
videTampon ;
if periph<10 then read (k);
end;

procedure Camembert ; ( ***** Tarte ou Camembert ? ***** )
begin
    ModeRemplissage(2);
    Hachures (1) ; arc (5000,8000,27000,0,130);
    Hachures (2) ; arc (5000,8000,27000,130,90);
    Hachures (3) ; arc (5000,8000,27000,220,80);
    Hachures (4) ; arc (5000,8000,27000,300,60);
    ModeRemplissage(3);
    Hachures (3) ; arc (5000,22000,27000,20,160);
    Hachures (5) ; arc (5000,22000,27000,180,140);
    hachures (6) ; arc (5000,23000,27000,320,60);
    ModeRemplissage(0);
    arc (16300,16350,6000,0,40);
    ModeRemplissage(1);
    arc (16300,16350,6000,40,30);
    ModeRemplissage(2);
    hachures (2) ; arc (16300,16350,6000,70,45);
    hachures (4) ; arc (16300,16350,6000,115,30);
    ModeRemplissage(3);
    hachures (3) ; arc (16300,16350,6000,145,25);
    hachures (6) ; arc (16300,16350,6000,170,10);
    taille (30) ;
    texte ('Bons graphismes avec',5000,4000);
    taille (40) ;
    texte ('G.S.X',10000,2000);
    videTampon;
    if periph<10 then read (k) ;
end;

procedure sautePage ;
begin
    if periph>20 then
        begin
            write (1st,#12) ; ( Saut de page sur imprimante )
        end
    else clrscr ;
end;

begin
    clrscr;
    write ('Quel peripherique voulez vous utiliser (1,21,22) ? ');
    read (periph) ;
    Ouverture (periph,'111111001');
    couleur (1) ; { Utile sur CPC 6128 }
    TracesPossibles ;
    SautePage;
    Histogramme ;
    SautePage;
    Camembert ;
    SautePage;
    fermeture;
end.●

```