

Fonctions et procédures standards DE PASCAL

B. CHERGOU

NOTA : Ce fascicule ne contient pas toutes les fonctions et procédures standards disponibles sous Free Pascal, qui sont au nombre de 265, mais qu'une sélection.

En principe, seules les fonctions et procédures courantes pourront être utilisées car les procédures graphiques et celles portant sur les fichiers nécessitent des connaissances qui ne sont pas données dans le cours d'algorithmique. Si vous souhaitez vous initier au mode graphique et/ou à l'utilisation des fichiers demandez les documentations qui vous aideront à le faire.

N'hésitez pas à consulter l'aide de PASCAL !

I.	FONCTIONS STANDARDS COURANTES.....	1
II.	PROCEDURES STANDARDS COURANTES	6
III.	LES PROCEDURES ET LES FONCTIONS GRAPHIQUES STANDARDS	10
IV.	FONCTIONS ET PROCEDURES STANDARDS DES FICHIERS TYPES	15
V.	FONCTIONS ET PROCEDURES STANDARDS DES FICHIERS TEXTES.....	16

I. FONCTIONS STANDARDS COURANTES

NOTA : Les procédures ou fonctions appartiennent à des UNITES STANDARDS, lors de leur utilisation, ne pas omettre de préciser le nom de cette unité dans la clause USES, **sauf pour l'unité SYSTEM qui est prise par défaut.**

Exemple :

```
PROGRAM exo10;
USES CRT;
.
.
BEGIN
CLRSCR; {CLRSCR appartenant à l'unité CRT, il faut alors le préciser par USES CRT}
.
END.
```

Nom de la fonction	:	ABS (x)
Unité	:	System
Types des paramètres	:	réel ou entier
Type du résultat	:	le même que x
Rôle	:	donne la valeur absolue de x
Exemple	:	ABS(-13.5) donne 13.5

Nom de la fonction	:	ARCTAN (x)
Unité	:	System
Types des paramètres	:	réel
Type du résultat	:	réel (en radians)
Rôle	:	donne l'arc tangente de x
Exemple	:	

Nom de la fonction	:	CHR (x)
Unité	:	System

Types des paramètres : entier
 Type du résultat : caractère
 Rôle : donne le caractère (s'il est dans la liste des caractères ASCII) dont la valeur est x
 Exemple : x := CHR (68) mettra la caractère D dans x
 Write (CHR (7)) déclenchera un BIP sonore

Nom de la fonction : **CONCAT (ch1 [,ch2,.....,chn])**
 Unité : System
 Types des paramètres : ch1,ch2,...,chn de type chaîne de caractères
 Type du résultat : chaîne de caractères
 Rôle : concatène plusieurs chaînes de caractères
 Exemple : après la séquence
 x := '23'
 y := '1994'
 c := CONCAT (x,' Janvier ',y)
 on aura dans c la chaîne '23 Janvier 1994'

Nom de la fonction : **COPY (ch , p , l)**
 Unité : System
 Types des paramètres : ch : chaîne
 p et l : entier
 Type du résultat : chaîne
 Rôle : donne à partir de la chaîne ch une sous chaîne de longueur l à partir de la position p
 Exemple : après la séquence
 c := 'octnovdecjan'
 long := 3
 mois := COPY (c,7,long)
 on aura dans l'objet mois la sous-chaîne 'dec'

Nom de la fonction : **COS (x)**
 Unité : System
 Types des paramètres : réel ou entier
 Type du résultat : réel (en radians)
 Rôle : donne le cosinus de x
 Exemple :

Nom de la fonction : **EXP (x)**
 Unité : System
 Types des paramètres : réel
 Type du résultat : réel
 Rôle : donne l'exponentielle de x
 Exemple : EXP(10.50) donne 36315.5

Nom de la fonction : **FRAC (x)**
 Unité : System
 Types des paramètres : réel ou entier
 Type du résultat : réel
 Rôle : donne la partie fractionnaire de x
 Exemple : FRAC (138.278) donne 0.278

Nom de la fonction : **INT (x)**
 Unité : System

Types des paramètres	:	réel ou entier
Type du résultat	:	réel
Rôle	:	donne la partie entière de x
Exemple	:	INT (138.456) donne 138.
Nom de la fonction	:	KEYPRESSED
Unité	:	Crt
Types des paramètres	:	/
Type du résultat	:	booléen
Rôle	:	retourne TRUE si une touche est appuyée sur le clavier ou FALSE dans le cas contraire
Exemple	:	la séquence repeat write (CHR (7)); until KEYPRESSED; Va déclencher la sonnerie jusqu'à ce qu'une touche soit appuyée
Nom de la fonction	:	LENGTH (ch)
Unité	:	System
Types des paramètres	:	ch:chaîne
Type du résultat	:	entier
Rôle	:	donne la longueur effective de la chaîne ch. (Ne pas confondre la longueur effective de la chaîne avec la dimension de la chaîne)
Exemple	:	après la séquence c1 := 'octnovdecjan' x := length (c1) on aura 12 dans x
Nom de la fonction	:	LN (x)
Unité	:	System
Types des paramètres	:	réel
Type du résultat	:	réel
Rôle	:	donne le logarithme népérien de x
Exemple	:	$X^n = e^{n \log x}$ pour x > 0 s'écrit XN := EXP (n * LN(x))
Nom de la fonction	:	ODD (x)
Unité	:	System
Types des paramètres	:	entier
Type du résultat	:	booléen
Rôle	:	donne la valeur booléenne TRUE si x est impair et FALSE si x est pair
Exemple	:	x := 16; r := ODD(x); r va contenir FALSE
Nom de la fonction	:	ORD (x)
Unité	:	System
Types des paramètres	:	scalaire non réel
Type du résultat	:	entier
Rôle	:	donne le numéro d'ordre de la valeur scalaire associée. (la première valeur a pour rang zéro
Exemple	:	TYPE sem = (sept,oct,nov,dec,jan); pour le type défini précédemment ORD (oct) donne 1 ORD ('A') donne 65 (c'est le rang du A dans la liste des caractères ASCII)
Nom de la fonction	:	PI
Unité	:	System

Types des paramètres	:	/
Type du résultat	:	réel
Rôle	:	donne la valeur de PI = 3.141592653589792385
Exemple	:	
Nom de la fonction	:	POS (ch1 , ch2)
Unité	:	System
Types des paramètres	:	ch1 et ch2 de type chaîne
Type du résultat	:	octet (byte)
Rôle	:	donne la position de la sous-chaîne ch1 dans la chaîne ch2 nota : si ch n'est pas trouvée elle retourne la valeur zéro.
Exemple	:	après la séquence c1 := 'octnovdecjan' x := POS('ja',c1) y := POS ('aj',c1) on aura 10 dans x et 0 dans y
Nom de la fonction	:	PRED (x)
Unité	:	System
Types des paramètres	:	tout type scalaire
Type du résultat	:	le même que x
Rôle	:	donne la valeur scalaire qui précède x selon l'ordre des valeurs scalaires définies par son type
Exemple	:	TYPE sem = (sept,oct,nov,dec,jan); l'ordre x := pred (nov) va transférer oct dans x
Nom de la fonction	:	RANDOM [(x)]
Unité	:	System
Types des paramètres	:	Mot
Type du résultat	:	Réel ou mot
Rôle	:	Si x n'est pas précisé la fonction donne un nombre aléatoire compris entre 0 et 1. Par contre si x est précisé elle donne un nombre aléatoire compris entre 0 et x. . Si x = 0 la fonction donne 0
Exemple	:	res1 := random; res2 := random(45);
Nom de la fonction	:	READKEY
Unité	:	Crt
Types des paramètres	:	/
Type du résultat	:	caractère
Rôle	:	lit un caractère à partir du clavier
Exemple	:	lettre := READKEY; if lettre = 'x' then rep := 'oui' else rep := 'non';
Nom de la fonction	:	ROUND (x)
Types des paramètres	:	réel
Type du résultat	:	entier
Rôle	:	donne la valeur de x arrondie à l'entier le plus proche
Exemple	:	x := ROUND (17.68) mettra 18 dans x y := ROUND (17.38) mettra 17 dans y
Nom de la fonction	:	SIN (x)
Unité	:	System

Types des paramètres : réel ou entier
 Type du résultat : réel
 Rôle : donne le sinus de x
 Exemple :

Nom de la fonction : **SQR (x)**
 Unité : System
 Types des paramètres : réel ou entier
 Type du résultat : le même que x
 Rôle : donne le carré de x
 Exemple : SQR(2) donne 4

Nom de la fonction : **SQRT (x)**
 Unité : System
 Types des paramètres : réel ou entier
 Type du résultat : réel
 Rôle : donne la racine carrée de x
 Exemple : SQRT (16) donne 4

Nom de la fonction : **SUCC (x)**
 Unité : System
 Types des paramètres : tout type scalaire
 Type du résultat : le même que x
 Rôle : Donne le successeur de x
 Exemple : y := SUCC (nov) va transférer dec dans y

Nom de la fonction : **SWAP (x)**
 Unité : System
 Types des paramètres : entier
 Type du résultat : le même que x
 Rôle : Permute le 1^{er} et le dernier octet de x
 Exemple : y := swap(1993) donne 3991

Nom de la fonction : **TRUNC (x)**
 Unité : System
 Types des paramètres : réel
 Type du résultat : entier
 Rôle : si $x \geq 0$ elle donne le plus grand entier plus petit ou égal à x
 si $x < 0$ elle donne le plus petit entier plus grand ou égal à x
 Exemple : TRUNC (3.73) donne 3
 TRUNC (-5.11) donne -5

Nom de la fonction : **UPCASE (x)**
 Unité : System
 Types des paramètres : caractère
 Type du résultat : caractère
 Rôle : donne la majuscule de x. Si x n'a pas de majuscule, il reste inchangé
 Exemple :

Nom de la fonction : **WHEREX**
 Unité : Crt
 Types des paramètres : /
 Type du résultat : octet
 Rôle : donne le numéro de la colonne dans laquelle se trouve le curseur, dans la fenêtre courante
 Exemple :

Nom de la fonction	:	WHEREY
Unité	:	Crt
Types des paramètres	:	/
Type du résultat	:	octet
Rôle	:	donne le numéro de la ligne dans laquelle se trouve le curseur, dans la
fenêtre courante	:	
Exemple	:	

II. PROCEDURES STANDARDS COURANTES

Nom de la procédure	:	CLREOL
Unité	:	Crt
Type des paramètres	:	/
Rôle	:	efface tous les caractères à partir de la position du curseur jusqu'à la fin de la ligne, sans déplacer le curseur.
Exemple	:	

Nom de la procédure	:	CLRSCR
Unité	:	Crt
Type des paramètres	:	
Rôle	:	efface la fenêtre active et positionne le curseur dans le coin supérieur gauche
Exemple	:	

Nom de la procédure	:	DEC (x [, n])
Unité	:	System
Type des paramètres	:	x est un scalaire n de type longint
Rôle	:	décrémente x avec un pas égal à n, si n est omis le pas est égal à 1
Exemple	:	après la séquence PROGRAM exple; VAR ind : integer; pas : longint; BEGIN ind := 100; pas := 6; DEC (ind,pas); END. on aura dans ind la valeur 94

Nom de la procédure	:	DELAY (X)
Unité	:	Crt
Type des paramètres	:	x : word
Rôle	:	provoque un temps d'arrêt de x millisecondes. (Attention : la durée exacte peut varier en fonction du matériel utilisé.)
Exemple	:	

Nom de la procédure	:	DELETE (ch , p , l)
Unité	:	System
Type des paramètres	:	ch : chaîne p,l :entier
Rôle	:	supprime l caractères de la chaîne ch à partir de la position p. Si p est plus grand que la longueur de ch, aucun caractère n'est supprimé. Si l désigne plus de caractères qu'il n'en reste, le reste de la chaîne est supprimé.
Exemple	:	après la séquence

```

c := 'octnovdecjan';
pos := 7;
DELETE (c,pos,3);
on aura dans c la chaîne 'octnovjan'

```

Nom de la procédure : **DELLINE**
 Unité : Crt
 Type des paramètres : /
 Rôle : supprime la ligne sur laquelle se trouve le curseur
 Exemple :

Nom de la procédure : **GETDATE (Année , Mois , Jour , Joursemaine)**
 Unité : Dos
 Type des paramètres :
 année : 1980..2099
 mois : 1..12
 jour : 1..31
 joursemaine : 0..6 (0 correspondant à dimanche)
 Rôle : donne la date courante contenue dans le système d'exploitation
 Exemple : GETDATE(a,m,j,js) ;
 write(a,m,j,js);
 donne 20072180 (il s'agissait du dimanche 18 février 2007)

Nom de la procédure : **GETTIME (Heure , Minute , Seconde , centième)**
 Unité : Dos
 Type des paramètres :
 heure : 0..23
 minute : 0..59
 seconde : 0..59
 centième : 0..99
 Rôle : donne l'heure courante contenue dans le système d'exploitation
 Exemple : GETtime(h,m,s,c) ;
 write(h,m,s,c);
 donne 16525330 (il s'agissait de 16h:52min:53src:30cent)

Nom de la procédure : **GOTOXY (x , y)**
 Unité : Crt
 Type des paramètres : x , y : octet (byte)
 Rôle : positionne le curseur au point spécifié par x et y (c étant la colonne et y la ligne. Le coin supérieur gauche a pour coordonnées 1 , 1)
 Exemple :

Nom de la procédure : **INC (x [, n])**
 Unité : System
 Type des paramètres :
 x : scalaire
 n : longint
 Rôle : incrémente x avec un pas égal à n, si n est omis le pas est égal à 1
 Exemple :
 après la séquence
 VAR ind : integer;
 pas : longint;
 BEGIN
 ind := 100;
 pas := 6;
 INC (ind, pas);
 INC (c) ;
 END;
 on aura dans ind la valeur 107

Nom de la procédure	:	INSERT (ch1, ch2, p)
Unité	:	System
Type des paramètres	:	ch1 ,ch2 : chaîne p : entier
Rôle	:	insere la chaîne ch1 dans la chaîne ch2 à partir de la position p
Exemple	:	après la séquence c1 := 'octnovdecjanmar'; r := insert ('fev',c1,13); on aura dans r la chaîne octvovdecjanfevmar
Nom de la procédure	:	INLINE
Unité	:	Crt
Type des paramètres	:	/
Rôle	:	insère une ligne à partir de la position du curseur. (avec décalage vers le bas de l'écran des autres lignes)
Exemple	:	
Nom de la procédure	:	NOSOUND
Unité	:	Crt
Type des paramètres	:	/
Rôle	:	provoque l'arrêt de la sonnerie
Exemple	:	la séquence sound(200); delay (100); nosound; va permettre d'émettre un son d'une fréquence de 200 hertz durant 100 millisecondes
Nom de la procédure	:	RANDOMIZE
Type des paramètres	:	/
Rôle	:	initialise le générateur de nombres aléatoires avec une valeur aléatoire nota : le nombre généré est stocké dans une variable prédéclarée de type longint appelée RANDSEED.
Exemple	:	randomize; write(randseed); donne 943001646 (bien sur le résultat change à chaque fois que la procédure est appelée car son but est de fournir un nombre aléatoire.) nota : Randseed est prédéclaré, il ne faut donc pas le déclarer.
Nom de la procédure	:	SOUND (x)
Unité	:	Crt
Type des paramètres	:	x : mot
Rôle	:	déclenche un son d'une fréquence de x hertz . Elle ne s'arrêtera qu'à la rencontre de la procédure NOSOUND
Exemple	:	SOUND (200);
Nom de la procédure	:	TEXTBACKGROUND (x)
Unité	:	Crt
Type des paramètres	:	x : byte
Rôle	:	permet de choisir la couleur de fond de l'écran, parmi les 8 constantes de couleur suivantes :
		black = 0 (noir)
		blue = 1 (bleu)
		green = 2 (vert)
		cyan = 3 (cyan)

red	=	4	(rouge)
magenta	=	5	(magenta)
brown	=	6	(brun)
lightgray	=	7	(gris clair)

Exemple : `textbackground(2)`
la couleur de fond de l'écran sera verte

Nom de la procédure : **TEXTCOLOR (x)**

Unité : Crt

Type des paramètres : x : byte

Rôle : permet de choisir la couleur des caractères parmi les 16 couleurs suivantes:

black	=	0	(noir)
blue	=	1	(bleu)
green	=	2	(vert)
cyan	=	3	(cyan)
red	=	4	(rouge)
magenta	=	5	(magenta)
brown	=	6	(brun)
lightgray	=	7	(gris clair)
darkgray	=	8	(gris foncé)
lightblue	=	9	(bleu clair)
lightgreen	=	10	(vert clair)
lightcyan	=	11	(cyan clair)
lightred	=	12	(rouge clair)
lightmagenta	=	13	(magenta clair)
yellow	=	14	(jaune)
white	=	15	(blanc)

On peut obtenir un clignotement soit en rajoutant 128 au numéro de la couleur, soit en mettant + BLINK après le numéro de la couleur.

Exemple : `textcolor(2)` permet d'écrire les caractères en vert

`textcolor(130)` permet d'écrire les caractères en vert et de plus ils clignoteront . Le même effet est obtenu en écrivant `textcolor(2 + blink)` `textcolor(green + blink)`

Nom de la procédure : **VAL (s, v, code)**

Unité : Crt

Type des paramètres : s : chaîne (doit contenir une séquence de chiffres représentant une valeur)
v : entier ou réel
code : entier

Rôle : convertit une chaîne en un nombre formé par la chaîne.<cependant si c <> 0 cela signifie qu'il y a une erreur, et code donne la position de l'erreur.

Exemple : si C contient la chaîne '12345', après l'appel de `VAL(c, r, Co)`
r contient la valeur 12345 et Co contient 0.
si C contient la chaîne '123A45', après l'appel de `VAL(c, r, co)`
Co contient 4, cela signifie qu'il y a une erreur et qu'elle se trouve à la position 4 . Effectivement cette dernière contient le caractère A qui n'est pas un chiffre

Nom de la procédure : **WINDOW (x1 , y1 , x2 , y2)**

Unité : Crt

Type des paramètres : x1,y1,x2,y2 : byte

Rôle : définit une fenêtre texte sur l'écran. x1 et y1 sont les coordonnées du coin supérieur gauche et x2, y2 celles du coin inférieur droit. Par défaut, la taille de la fenêtre texte est 1,1,80,25) si le mode lignes/écran = 25 ou (1,1,80,43) si le mode lignes/écran = 43. La taille minimum d'une fenêtre est 1 colonne par 1 ligne. Lors d'un appel de la procédure WINDOW le curseur se positionne toujours à la position 1,1 de la fenêtre.

UNITE PRINTER

Elle n'est utilisée que pour envoyer des données vers l'imprimante, elle déclare un fichier texte appelé LST, lequel est associé à l'unité de sortie LTP1.

Exemple : Lorsque l'on veut sortir nos résultats sur l'imprimante et non sur l'écran qui est l'unité de sortie par défaut, on écrit :

USES PRINTER;

WRITELN(LST, param1,param2,.....paramn);

III. LES PROCEDURES ET LES FONCTIONS GRAPHIQUES STANDARDS

ATTENTION : pour utiliser ces modules vous devez être en mode graphique.

Nom de la procédure	:	ARC (x , y, angledebut, anglefin, rayon)
Unité	:	Graph
Type des paramètres	:	x,y :entier angledebut, anglefin, rayon : mot
Rôle	:	Trace un arc de cercle de rayon RAYON et dont les coordonnées du centre sont x et y et angledébut et anglefin indiquent le début et la fin du tracé (les angles sont exprimés en degrés). Si angledebut = 0 et anglefin = 360, arc dessine un cercle. Les angles augmentent dans le sens inverse des aiguilles d'une montre. 0° à 3 heures, 90° à 12 h.,...
Nom de la procédure	:	BAR (x1, y1, x2, y2)
Unité	:	Graph
Type des paramètres	:	x1, y1, x2, y2 : entier
Rôle	:	dessine une barre ayant pour coordonnées du coin supérieur gauche (x1,y1) et pour coordonnées du coin supérieur droit (x2, y2).
Nom de la procédure	:	BAR3D (x1,y1, x2, y2, epaisseur, haut)
Unité	:	Graph
Type des paramètres	:	x1, y1, x2, y2 : entier epaisseur : mot haut : booléen
Rôle	:	Dessine une barre à 3 dimensions dont les coordonnées des coins supérieurs gauche et droit sont ,respectivement (x1, y1) et (x2,y2), avec comme epaisseur EPAISSEUR et fermée si haut est vrai ou non fermée dans le cas contraire.
Nom de la procédure	:	CIRCLE (x, y, rayon)
Unité	:	Graph
Type des paramètres	:	x, y : entier rayon : mot
Rôle	:	Dessine un cercle de rayon RAYON et dont les coordonnées du centre sont (x,y)

Nom de la procédure	:	CLEARDEVICE
Unité	:	Graph
Type des paramètres	:	/
Rôle	:	Efface l'écran
Nom de la procédure	:	CLEARVIEWPORT
Unité	:	Graph
Type des paramètres	:	/
Rôle	:	Efface la fenêtre graphique courante
Nom de la procédure	:	CLOSEGRAPH
Unité	:	Graph
Type des paramètres	:	/
Rôle	:	Permet de quitter le mode graphique et de revenir au mode texte
Nom de la procédure	:	DRAWPOLY (nbrepoints, points)
Unité	:	Graph
Type des paramètres	:	nbrepoints : mot points est un paramètre sans type contenant les coordonnées des sommets du polygone
Rôle	:	Dessine un polygone ou nbrepoints indique le nombre de sommets, donc le nombre de coordonnées contenues dans points. Chaque coordonnée est un couple (x,y) de type mot. Si vous voulez dessiner un polygone fermé, il faudra transmettre à drawpoly n + 1 coordonnées (ou points[n + 1] = points[1]) . Voir l'exemple donné dans la partie C.
Nom de la procédure	:	ELLYPSE (x, y, angledebut, anglefin, rayonx, rayony)
Unité	:	Graph
Type des paramètres	:	x, y : entier angledebut, anglefin, rayonx, rayony : mot
Rôle	:	Dessine un arc elliptique à partir de angledebut à anglefin. Le centre ayant pour coordonnées (0, 0) et de rayon horizontal rayonx et de rayon vertical rayony.
Nom de la procédure	:	FILLELLYPSE (x, y, rayonx, rayony)
Unité	:	Graph
Type des paramètres	:	x, y : entier rayonx, rayony : mot
Rôle	:	Dessine une ellipse remplie ayant pour centre le point de coordonnées (x, y) et rayonx de rayon horizontal et rayony de rayon vertical. Le remplissage se fait avec la couleur et le motif de remplissage courant, et le contour avec la couleur de tracé courant.
Nom de la procédure	:	FILLPOLY (nbrepts, pts)
Unité	:	Graph
Type des paramètres	:	nbrepts : mot pts : est un paramètre sans type contenant les coordonnées des sommets du polygone.
Rôle	:	Dessine un polygone rempli par le motif de remplissage et la couleur courants, et dont le contour a le style et la couleur courants . (Voir SETLINESTYLE et DRAWPOLY).

Nom de la procédure	:	FLOODFILL (x , y , bordure)
Unité	:	Graph
Type des paramètres	:	x , y : entier bordure : mot
Rôle	:	Remplit une surface avec le motif de remplissage courant. X et y représentent les coordonnées d'un point se trouvant dans la surface que l'on veut remplir. Le motif est celui fixé par SETFILLSTYLE ou SETFILLPATTERN. Bordure définit la couleur délimitant la surface.
Nom de la procédure	:	GETASPECTRATIO (x , y)
Unité	:	Graph
Type des paramètres	:	x , y : mot
Rôle	:	Permet d'avoir des formes parfaitement circulaires. Le ratio ou facteur d'allongement est obtenu en calculant x/y; vous devez utiliser ce ratio si les dessins ont une forme allongée.
Nom de la Fonction	:	GETMAXX
Unité	:	Graph
Type des paramètres	:	/
Rôle	:	Fournit la valeur maximale de la coordonnée x
Nom de la Fonction	:	GETMAXY
Unité	:	Graph
Type des paramètres	:	/
Rôle	:	Fournit la valeur maximale de la coordonnée y
Nom de la Fonction	:	GETX
Unité	:	Graph
Type des paramètres	:	/
Rôle	:	Fournit la valeur actuelle du curseur graphique de la coordonnée x
Nom de la Fonction	:	GETY
Unité	:	Graph
Type des paramètres	:	/
Rôle	:	Fournit la valeur actuelle du curseur graphique de la coordonnée y
Nom de la procédure	:	LINE (x1, y1, x2, y2)
Unité	:	Graph
Type des paramètres	:	x1, y1, x2, y2 : entier
Rôle	:	Dessine une ligne dont les coordonnées de début sont (x1,y1) et les coordonnées de fin (x2 , y2).
Nom de la procédure	:	LINE TO (x, y)
Unité	:	Graph
Type des paramètres	:	x, y : entier
Rôle	:	Dessine un trait de la position actuelle du curseur au point de coordonnées (x, y)
Nom de la procédure	:	MOVETO (x, y)
Unité	:	Graph
Type des paramètres	:	x, y : entier
Rôle	:	Positionne le curseur au point de coordonnées (x, y).

Nom de la procédure : **OUTTEXT (ch)**
 Unité : Graph
 Type des paramètres : ch : chaîne de caractères
 Rôle : Affiche la chaîne ch à partir de la position du curseur (NB : les données numériques doivent être sous forme de chaînes de caractères)

Nom de la procédure : **OUTTXTXY (x, y, ch)**
 Unité : Graph
 Type des paramètres : x, y : entier
 ch : chaîne de caractères
 Rôle : Affiche la chaîne ch à partir du point ayant pour coordonnées (x, y).

Nom de la procédure : **PIESLICE (x, y, angledebut, anglefin, rayon)**
 Unité : Graph
 Type des paramètres : x, y : entier
 angledebut, anglefin, rayon : mot
 Rôle : Dessine et remplit un diagramme par secteur, appelé aussi " camembert " de rayon RAYON avec comme début et fin du tracé : angledebut et anglefin et comme centre le point de coordonnées (x, y).

Nom de la procédure : **PUTPIXEL (x, y, couleur)**
 Unité : Graph
 Type des paramètres : x, y : entier
 couleur : mot
 Rôle : Allume un point ayant pour coordonnées (x, y) et comme couleur COULEUR

Nom de la procédure : **RECTANGLE (x1, y1, x2, y2)**
 Unité : Graph
 Type des paramètres : x1,y1, x2, y2 : entier
 Rôle : Dessine un rectangle dont le coin supérieur gauche a pour coordonnées (x1,y1) et le coin inférieur droit les coordonnées (x2, y2).

Nom de la procédure : **SECTOR (x, y, angledebut, anglefin, axex, axey)**
 Unité : Graph
 Type des paramètres : x, y : entier
 angledebut, anglefin, axex, axey : mot
 Rôle : Dessine un arc d'ellipse plein, avec x,y comme coordonnées du centre, axex et axey représentant les axes utilisés et les débuts et fin d'ellipse sont définis par angledebut et anglefin.

Nom de la procédure : **SETBKCOLOR (couleur)**
 Unité : Graph
 Type des paramètres : couleur : mot
 Rôle : Définit la couleur courante de fond de l'écran

black	=	0	(noir)
blue	=	1	(bleu)
green	=	2	(vert)
cyan	=	3	(cyan)
red	=	4	(rouge)
magenta	=	5	(magenta)

brown	=	6	(brun)
lightgray	=	7	(gris clair)

Nom de la procédure : **SETCOLOR (couleur)**
 Unité : Graph
 Type des paramètres : couleur : mot
 Rôle : Définit la couleur courante du dessin

black	=	0	(noir)
blue	=	1	(bleu)
green	=	2	(vert)
cyan	=	3	(cyan)
red	=	4	(rouge)
magenta	=	5	(magenta)
brown	=	6	(brun)
lightgray	=	7	(gris clair)
darkgray	=	8	(gris foncé)
lightblue	=	9	(bleu clair)
lightgreen	=	10	(vert clair)
lightcyan	=	11	(cyan clair)
lightred	=	12	(rouge clair)
lightmagenta	=	13	(magenta clair)
yellow	=	14	(jaune)
white	=	15	(blanc)

Nom de la procédure : **SETFILLSTYLE (motif, couleur)**
 Unité : Graph
 Type des paramètres : motif, couleur : mot
 Rôle : Permet de définir une couleur et un motif de remplissage

0	remplir avec la couleur de fond
1	remplir avec la couleur de remplissage pleine
2
3	/////
4	///// en gras
5	\\\\\\
6	\\\\\\ en gras
7	hachures fines
8	hachures croisées épaisses
9	trame serrée
10	points espacés
11	points serrés
12	définit par l'utilisateur

NOTA : Le motif choisi pourra être utilisé par les procédures FII_poly, BAR, BAR3D, PIESLICE, FLOODFILL

Nom de la procédure : **SETLINESTYLE (type, motif, épaisseur)**
 Unité : Graph
 Type des paramètres : type, motif, épaisseur : mot
 Rôle : Définit les type, motif et épaisseur des traits

SOLIDLN	0	plein
DOTTEDLN	1	pointillé
CENTERLN	2	tiret-point
DASHEDLN	3	tiret
USERBITLN	4	défini par l'utilisateur
NORMWIDTH	1	épaisseur normale

THICKWIDTH 3 gras

Nom de la procédure : **SETTEXTJUSTIFY (hor, vert)**
 Unité : Graph
 Type des paramètres : hor, vert : mot
 Rôle : Définit les types de justification horizontale et verticale

= 0 alignement gauche

= 1 centré

= 2 alignement droit

= 0 alignement sous le point

= 1 centré

= 2 alignement sur le point

Nom de la procédure : **SETTEXTSTYLE (fonte, direction, taillecar)**
 Unité : Graph
 Type des paramètres : fonte, direction, taillecar : mot
 Rôle : Fixe la fonte de caractères, la direction d'affichage (horizontale, verticale) et la taille des caractères du texte à afficher (taillecar par défaut = 1 pour les fontes par défaut (matrice 8 x 8) et 4 pour les fontes matricielles. La taille maxi = 10)

DEFAULTFONT 0 par défaut

TRIPLEXFONT 1 caractères vectoriels

SMALLFONT 2 petite fonte

SANSSERIFFONT 3 helvetica

GOTIKFONT 4 gothique

HORIZDIR 0 de gauche à droite

VERTDIR 1 de bas en haut

NORMSIZE 1 texte de taille normale

Nom de la procédure : **SETVIEWPORT (x1, y1, x2, y2, limite)**
 Unité : Graph
 Type des paramètres : x1, y1, x2, y2 : entier
 limite : booléen
 Rôle : Définit une fenêtre graphique courante dont (x1, y1) et (x2, y2) sont les coordonnées des coins supérieur gauche et inférieur droit. La variable limite détermine si les affichages sont limités à la fenêtre active

IV. FONCTIONS ET PROCEDURES STANDARDS DES FICHIERS TYPES

ATTENTION : pour utiliser ces modules vous devez connaître les opérations de base sur les fichiers

Nom de la fonction : **EOF (f)**
 Types des paramètres : f : fichier
 Type du résultat : Booléen
 Rôle : détecte la fin de fichier. VRAI si fin de fichier, FAUX dans le cas contraire

Nom de la fonction : **IORESULT**
 Types des paramètres : /
 Type du résultat : mot (word)

Rôle : Retourne une valeur indiquant si la dernière opération d'E/S s'est déroulée sans erreur. (0 indique une opération sans erreur).

Nom de la fonction : **FILEPOS (f)**

Types des paramètres : f : fichier

Type du résultat : Longint

Rôle : Retourne la position actuelle dans un fichier .

Nom de la fonction : **FILESIZE (f)**

Types des paramètres : f : fichier

Type du résultat : Longint

Rôle : Retourne la taille en nombre de composants du fichier.

Nom de la procédure : **ERASE (f)**

Type des paramètres : f : fichier

Rôle : Supprime un fichier , s'il n'est pas ouvert mais dont l'assignation a été faite, du support.

ATTENTION : à utiliser avec précaution car le fichier est définitivement perdu.

Nom de la procédure : **SEEK (f , pos)**

Type des paramètres : f : fichier , pos : longint

Rôle : Place à la position pos le pointeur du fichier indiqué. ATTENTION : le premier composant à pour rang zéro, donc dans un fichier de 100 composants, ces derniers sont numérotés de 0 à 99.

Nom de la procédure : **TRUNCATE (f)**

Type des paramètres : f : fichier

Rôle : Tronque un fichier à partir de la position actuelle

Nom de la procédure : **RENAME (f , nouv_nom_de_fichier_externe)**

Type des paramètres : f : fichier, nouv_nom_de_fichier_externe : chaîne

Rôle : Renomme un fichier externe. Attention ! à faire après l'assignation mais avant l'ouverture.

Exemple : RENAME (repert , 'a:f2.dat');

V. FONCTIONS ET PROCEDURES STANDARDS DES FICHIERS TEXTES

Nom de la procédure : **APPEND (f)**

Type des paramètres : f : texte

Rôle : Ouvre un fichier texte existant déjà pour ajouter une ou plusieurs lignes à la fin du fichier.

Nom de la procédure : **FLUSH (f)**

Type des paramètres : f : texte

Rôle : Ecrit dans le fichier le contenu de la variable fenêtre même si elle n'est pas Totalement remplie. La rencontre de CLOSE , FLUSH est automatiquement exécuté.

Nom de la fonction : **EOLN (f)**

Types des paramètres : f : fichier

Type du résultat : Booléen

Rôle : Teste si nous sommes à la fin de la ligne.

NOTA : vous pouvez dupliquer cette page et y ajouter vos propres modules

Nom de la Fonction : Fibo (n)
Unité : mettre ici le nom de votre bibliothèque
Type des paramètres : entier
Type du résultat : entier
Rôle : Donne le Nième élément de la suite de Fibonacci
Exemple : donne 13 si n =7

Nom de la Fonction :
Unité :
Type des paramètres :
Type du résultat :
Rôle :
Exemple :

Nom de la Fonction :
Unité :
Type des paramètres :
Type du résultat :
Rôle :
Exemple :

Nom de la Fonction :
Unité :
Type des paramètres :
Type du résultat :
Rôle :
Exemple :

Nom de la Fonction :
Unité :
Type des paramètres :
Type du résultat :
Rôle :
Exemple :

Nom de la Fonction :
Unité :
Type des paramètres :
Type du résultat :
Rôle :
Exemple :

Nom de la Fonction :
Unité :
Type des paramètres :
Type du résultat :
Rôle :
Exemple :

Nom de la procédure :
Unité :
Type des paramètres :
Rôle :
Exemple :

Nom de la procédure :
Unité :
Type des paramètres :
Rôle :

Exemple :

Nom de la procédure :
Unité :
Type des paramètres :
Rôle :

Exemple :

Nom de la procédure :
Unité :
Type des paramètres :
Rôle :
Exemple
Nom de la procédure :
Unité :
Type des paramètres :
Rôle :

Exemple :

Nom de la procédure :
Unité :
Type des paramètres :
Rôle :

Exemple :

Nom de la procédure :
Unité :
Type des paramètres :
Rôle :
Exemple
Nom de la procédure :
Unité :
Type des paramètres :
Rôle :

Exemple :

: