

## Rappel

### Structure d'un programme

#### Syntaxe:

```
PROGRAM nom_du_programme ;  
USES wincrt ;           { bibliothèque nécessaire pour read et write }  
CONST identificateur = valeur ;   { déclaration ( éventuelle ) de(s) constante(s) }  
VAR identificateur1: type1 ;  
      identificateur2: type2 ;      { déclaration des variables }  
BEGIN                               { début des instructions }  
  instruction_1;  
  instruction_2;  
  instruction_n;  
END .                               { fin du programme }
```

#### Remarques:

Les instructions sont séparées par ' ; '

L'éditeur PASCAL ne distingue pas les majuscules des minuscules, ignore les tabulations, les répétitions d'espaces et les retours à la ligne.

Le ' END ' final est suivi d'un point. Tout ce qui suivra sera ignoré par le compilateur.

Ce qui est entre accolades { } est ignoré par le compilateur : Cela constitue un commentaire

#### Instructions élémentaires :

##### Entrée de données

Syntaxe            **READLN ( identificateur\_de\_variable );**

Permet l'entrée au clavier d'une variable

Seul un curseur clignotant indique que l'ordinateur attend une entrée. Il est judicieux d'afficher auparavant un message pour indiquer ce qu'on attend.

##### Affichage de données

Syntaxe

**WRITE ( identificateur\_de\_variable );**

Affiche à l'écran, à partir de la position courante du curseur, le contenu ( la valeur ) de la variable

**WRITE ( donnee\_1 , donnee\_2 , ... , donnee\_n );**

Affiche à l'écran les données ( contenu de variables, valeurs, ou textes ) sur une même ligne, sans espace.

**WRITELN ( . . . );**            Même chose, mais passe à la ligne après l'affichage

##### Affectation

Syntaxe

**identificateur\_1 := identificateur\_2 ;**

Affecte à la variable 1 le contenu de la variable 2

**identificateur := expression ;**

Affecte à la variable 1 le résultat de l'expression

Exemples: A := B ; { assigne à A la valeur de B }

A := i/j ; { assigne à A le quotient (de type real) de i par j }

## TP 2 INTRODUCTION A LA SYNTAXE PASCAL

### OPERATIONS ELEMENTAIRES

**BUT DU TP :** Manipulations et adaptations dans un environnement PASCAL (prendre connaissance du rôle de l'éditeur et du compilateur ) en utilisant des structures de données de base et des opérations élémentaires.

**Manipulation1:**

```
PROGRAM manip1;
USES crt ;
VAR
  a,b,c : integer;           { Partie de déclaration des variables }
                             { a, b et c sont 3 var. de type entier }
BEGIN
  Clrscr ;
  readln(a);                 { Saisie au clavier du contenu de a }
  b := 5;                    { Affectation avec le symbole « := » }
  c := a+b;
  writeln ('Valeur de c : ',c);
                             { Affichage d'un texte, puis du contenu }
                             { de c (séparé par une virgule) }
END.
```

1. Lancer l'éditeur de texte et taper le code source suivant :
2. Enregistrer dans un fichier *premier.pas* : tout fichier source en Pascal doit avoir l'extension « point pas ».
3. Reste maintenant à traduire ce code source en un binaire exécutable par l'ordinateur : c'est le rôle du compilateur. Après l'avoir compilé il faut exécuter le programme.  
Que fait le programme ?

A retenir :

- *readln* : saisie au clavier ; c'est l'équivalent de l'instruction *lire*
- *writeln* : affichage à l'écran ; c'est l'équivalent de l'instruction *ecrire*
- « := » : affecter une variable ; c'est l'équivalent de ←
- Les différents types de variables simples en Pascal sont :
  - Integer* : nombre entier
  - Real* : nombre réel, la partie entière et la partie décimale sont séparées par un point « . »
  - Boolean* : booléen, ne peut prendre que les deux valeurs *true* ou *false*
  - Char* : caractères
  - String* : chaîne de car
- Les principaux opérateurs mathématiques en Pascal sont :
  - +**: addition
  - : soustraction
  - \***: multiplication
  - /**: division (5 / 2 vaut 2.5)
  - div*: division entière (5 div 2 vaut 2)
  - mod* : reste de la division (5 mod 2 vaut 1)

**manipulation2:**

Ecrire un programme qui reçoit 2 entiers au clavier et qui affiche la somme, le produit, la moyenne, le résultat de la division entière du premier par le second, le reste de la division entière du premier par le second. On suppose que les nombres saisis sont correctes (ils sont différents de zéro). Le dialogue doit ressembler à :

```
Saisir 2 entiers
12
4
La somme est : 16
Le produit est : 48
La moyenne est : 8
Le résultat de la division entière est : 3
Le reste de la division entière est : 0
```

Manipulation3:

```
PROGRAM melange;
VAR x,y:real;
BEGIN
  write('x='); readln(x);
  write('y=');
  readln(y); x:=x+y;
  y:=x-y ; x:=x-y;
  writeln('après traitement :');
  writeln('x=',x, ',y=',y); END.
```

1-Exécuter plusieurs fois ce programme et expliquer les résultats obtenus.que fait le programme.

2-Comment aurait-on pu faire autrement pour avoir la même action ?

Manipulation 4 : Soit le programme :

```
Program affectation ;
Var a,b : integer ;
Begin
  b :=23 ; ----->
  a :=15 ;----->
  b :=a ; ----->
  a :=a+1 ; ----->
  b :=b-3 ; ----->
  b :=a mod 3 ; ----->
end.
```

Valeurs des variables	
a	b

Inscrire dans les colonnes prévues , les valeurs prises par les variables successivement après l'exécution de chaque instruction. Les valeurs indéterminées seront notées par « ? ».

Manipulation 5 : Compléter et exécuter le programme qui demande à l'utilisateur les coordonnées de 2 points distincts A(xA,yA) et B(xB,yB) du plan et qui affiche les coordonnées du point milieu C de AB et la distance

$$AB = \sqrt{(xA - xB)^2 + (yA - yB)^2}$$

```
program milieu ;
uses crt;
var xA, yA, xB, yB, xC, yC,AB : ..... ;
begin
  writeln ('Entrez les coordonnées du premier point A') ;
  ..... ;
  ..... ;
  writeln ('Entrez les coordonnées du deuxième point B') ;
  ..... ;
  ..... ;
  xC := ..... ;
  yC := ..... ;
  writeln ('Les coordonnées du milieu sont :', ..... , ..... ) ;
  AB := ..... ;
  Writeln('la distance entre A et B est',.....) ;
end.
```