

Chapitre 2

Les actions élémentaires simples

Leçon 1 :

L'affectation

I-Définition :

Une affectation permet d'affecter une valeur à une variable : **Variable** \leftarrow **Valeur**

Valeur peut être :

- Une constante ($X \leftarrow 3$)
 - Une variable ($X \leftarrow Y$)
 - Expression ($\text{Moy} \leftarrow (\text{DC} + 2 * \text{DS}) / 3$)
- ✓ L'expression à droite de signe d'affectation " \leftarrow ", sera complètement évaluée puis assignée (affectée) à la variable située à gauche de même signe.
- ✓ La valeur ou le résultat de l'expression à droite du signe d'affectation doit être de même type ou de type compatible avec celui de la variable à gauche.

II-Vocabulaire et syntaxe :

Analyse & Algorithme	Pascal
Forme générale: Variable \leftarrow valeur	Forme générale: Variable := valeur ;
Exemple: Moy \leftarrow (Note1 + note2) / 2	Exemple: Moy := (Note1 + note2) / 2 ;

Leçon 2:

Les opérations d'entrée/sortie

I-Les entrées :

 (lecture de données)

1-Définition :

Une entrée consiste à introduire une donnée à partir d'une source d'entrée (clavier, souris, stylo optique ...). C'est une opération qui permet d'affecter à une variable en mémoire, une valeur de même type ou compatible avec celle de la variable.

2-Vocabulaire et syntaxe :

Analyse	Algorithme	Pascal
<p>Forme générale: A=Donnée("commentaire sur A")</p> <p>var1, var2,var3=Donnée</p> <p>Exemple: Moy=Donnée("Entrer la moyenne:")</p>	<p>Forme générale: Écrire ("commentaire sur A") Lire (A) Lire(var1, var2,var3)</p> <p>Exemple: Écrire ("Entrer la moyenne:") Lire (moy) Ou Écrire ("Entrer la moyenne:"), Lire (moy)</p>	<p>Forme générale: WRITE (' commentaire sur A'); READLN (A); {ou READ (A) } ; Readln (var1, var2,var3) ; ou Read (var1, var2,var3);</p> <p>Exemple: WRITE (' Entrer la moyenne:'); READLN(moy);</p>

Remarques:

- WRITELN: affichage de donnée et retour à la ligne.
- READLN: lecture de données et retour à la ligne.
- La lecture de plusieurs variables de type chaîne à la fois est non fonctionnelle.

II-Les sorties :

 (écriture de données)

1-Définition :

La sortie est l'action élémentaire qui consiste à écrire une donnée sur un périphérique de sortie tel que l'écran, l'imprimante, ...

2-Vocabulaire et syntaxe :

Analyse & Algorithme	Pascal
<p>Affichage d'un texte: Écrire ("message") <i>Exemple:</i> Ecrire ("Bonjour")</p>	<p>Affichage d'un texte: WRITE ('Bonjour'); Bonjour</p>
<p>Affichage de contenu d'une variable: Écrire (variable) <i>Exemple :</i> A ← 165 Écrire (A)</p>	<p>Affichage de contenu d'une variable: A := 165 ; WRITELN (A) ; 165</p>
<p>Écrire (Expression) <i>Exemple :</i> Écrire (3+5)</p>	<p>Expression: WRITELN (3+5) ; 8</p>
<p>Affichage mixte: Écrire ("message", variable) <i>Exemple:</i> Écrire ("A=", A)</p>	<p>Affichage mixte: WRITE ('A=', A); A=165</p>

3- Formatage de l'affichage des résultats :

a- Affichage des entiers:

syntaxe : write(valeur-entière : n);

Affiche la valeur entière dans une colonne de n caractères à partir de la droite. Si la valeur entière comporte plus que n chiffres alors l'affichage commence par la gauche.

Exemple: n:=421;

	1	2	3	4	5	6	7	8	9
WRITELN (n:6);				4	2	1			
WRITELN (n:2);	4	2	1						

b- Affichage des réels:

syntaxe : WRITELN (variable:champ:chiffres significatifs);

Write(valeur_réelle); affiche le nombre en notation scientifique(x.xxxxxxxxExx précédé d'un espacement).

Write(valeur_réelle: np); affiche le nombre en notation scientifique sur np positions précédé d'un espacement.

Write(valeur_réelle: np:nd); affiche le nombre np positions avec nd décimales.

Exemple: x:=43.51;

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
WRITELN (x);		4	.	3	5	1	0	0	0	0	0	0	E	+	0	1	
WRITELN (x:6);		4	.	4	E	+	0	1									
WRITELN (x:10);		4	.	3	5	1	E	+	0	1							
WRITELN (x:7:3);		4	3	.	5	1	0										
WRITELN (x:4:4);		4	3	.	5	1	0	0									
WRITELN (x:2:1);		4	3	.	5												

c- Affichage des chaînes de caractères:

syntaxe: write(chaîne : n);

Affiche la chaîne sur n positions: insertion d'espacement à gauche de la chaîne s'il y a moins de n caractères sinon si n insuffisant alors ajustement automatique.

Exemple: ch:='Bonjour';

	1	2	3	4	5	6	7	8	9
WRITELN (ch:6);	B	o	n	j	o	u	r		
WRITELN (ch:8);		B	o	n	j	o	u	r	

d- Affichage des caractères:

syntaxe: write(car : n);

affiche le caractère à la position n et insertion d'espacement à gauche de caractère.

Exemple: car:= 'a';

	1	2	3	4	5	6	7	8	9
WRITELN (car:6);						a			
WRITELN (car:8);								a	

Exercice 8 : page 69

Préciser le résultat obtenu à la sortie d'un programme par les séquences d'instructions suivantes :

VAR var1, var2, var3, var4, var5 : REAL;

où :

var1 vaut 1.2

var2 vaut 0.00089

var3 vaut 45

var4 vaut 10095.095

var5 vaut 2.7182818281

Séquences d'instructions :

- 1) WRITE(var1 : 8 : 4);
- 2) WRITE(var2 : 8 : 4);
- 3) WRITE(var3 : 4 : 1);
- 4) WRITE(var4 : 3 : 1);
- 5) WRITE(var5 : 3 : 1);
- 6) WRITE(var1 : 6 : 4);
- 7) WRITE(var2 : 10 : 6);
- 8) WRITE(var3 : 10 : 6);
- 9) WRITE(var4 : 7 : 5);
- 10) WRITE(var5 : 10 : 4);

Corrigé :

1- :

var1=1.2 var2 = 0.00089 var3 = 45 var4 = 10095.095 var5 = 2.7182818281

- 1) WRITE(var1 : 8 : 4);
- 2) WRITE(var2 : 8 : 4);
- 3) WRITE(var3 : 4 : 1);
- 4) WRITE(var4 : 3 : 1);
- 5) WRITE(var5 : 3 : 1);
- 6) WRITE(var1 : 6 : 4);
- 7) WRITE(var2 : 10 : 6);
- 8) WRITE(var3 : 10 : 6);
- 9) WRITE(var4 : 7 : 5);
- 10) WRITE(var5 : 10 : 4);

1	2	3	4	5	6	7	8	9	10	11	12
		1	.	2	0	0	0				
		0	.	0	0	0	9				
4	5	.	0								
1	0	0	9	5	.	1					
2	.	7									
1	.	2	0	0	0						
		0	.	0	0	0	8	9	0		
		4	5	.	0	0	0	0	0		
1	0	0	9	5	.	0	9	5	0	0	
				2	.	1	7	8	3		

```

c:\bac2013\lex8p69.pas
program ex8page69;
uses wincrt;
VAR var1, var2, var3, var4, var5 : REAL;
Begin
var1 := 1.2;
var2 := 0.00089;
var3 := 45;
var4 := 10095.095;
var5 := 2.7182818281;

WRITEln(var1 : 8 : 4);
WRITEln(var2 : 8 : 4);
WRITEln(var3 : 4 : 1);
WRITEln(var4 : 3 : 1);
WRITEln(var5 : 3 : 1);
writeln;
WRITEln(var1 : 6 : 4);
WRITEln(var2 : 10 : 6);
WRITEln(var3 : 10 : 6);
WRITEln(var4 : 7 : 5);
WRITEln(var5 : 10 : 4);

end.
    
```

(Inactive C:\BAC2013\EX8P69.EXE)

```

1.2000
0.0009
45.0
10095.1
2.7

1.2000
0.000890
45.000000
10095.09500
2.7183
    
```

Exercice 9 : page 69

Écrire un programme en Pascal permettant de lire 2 nombres entiers positifs inférieurs à 999 et affiche à l'écran la multiplication et la division comme suit :

```

    120
  x  11
  .....
 = 1320

    130 : 11
     9  :.....
       : 11
       :

```

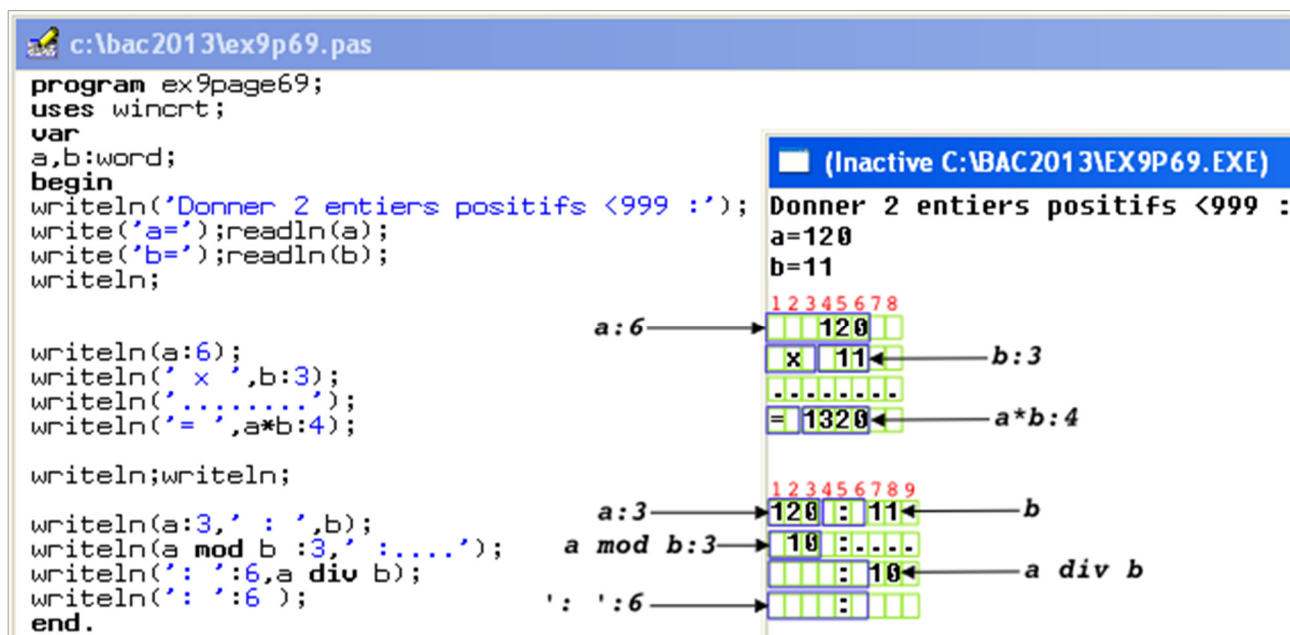
Corrigé :

```

1-
program ex9page69;
uses wincrt;
var
    a,b:word;
begin
    writeln('Donner 2 entiers positifs <999 :');
    write('a');readln(a);
    write('b');readln(b);
    writeln;
    writeln(a:6);
    writeln(' x ',b:3);
    writeln('.....');
    writeln('= ',a*b:4);
    writeln;writeln;

    writeln(a:3,' : ',b);
    writeln(a mod b :3,' :....');
    writeln(': ':6,a div b);
    writeln(': ':6 );
end.

```



Exercice 4 : page 50

Écrire un algorithme qui saisit un temps en seconde puis le convertit en jours, heure, minutes et secondes.

Corrigé :

sachant que jours=86400 heure=3600s minute=60s

Algorithme :

- 0) Début conversion
- 1) Ecrire("Donner un temps en seconde : "), lire(temps)
- 2) jours ← temps div 86400
- 3) heure ← (temps mod 86400) div 3600
- 4) minutes ← (temps mod 3600) div 60
- 5) secondes ← temps mod 60
- 6) Ecrire("jours=", jour, " heures = ", heures, " minutes= ", minutes, "Secondes= ", secondes)
- 7) Fin conversion

Remarque :

en Pascal il faut utiliser le type **longint** pour déclarer les variable (puisque'on dépasse la valeur 32767)

Exercice :

Ecrire la préanalyse, l'analyse, l'algorithme et le programme pascal d'un programme qui permet de saisir un entier de 3 chiffres puis affiche la somme de ces 3 chiffres.

Exemple : pour 123 S=1+2+3=6

Avec 2 méthodes : en utilisant (mod et div) puis avec (convch et valeur)

Préanalyse :

But : Afficher la somme de 3 chiffres d'un entier

données : n

Formules : $c \leftarrow n \text{ div } 100$ $d \leftarrow (n \text{ mod } 100) \text{ div } 10$ $u \leftarrow n \text{ mod } 10$
 $s \leftarrow u+d+c$

Analyse :

Nom : Somme

Résultat= Ecrire(" la somme de 3 chiffres est = ", s)

$s \leftarrow u+d+c$

$c \leftarrow n \text{ div } 100$

$d \leftarrow (n \text{ mod } 100) \text{ div } 10$

$u \leftarrow n \text{ mod } 10$

n=donnée("Donner un entier composé de 3 chiffres :")

Fin Somme

T.D.O (Tableau de Déclaration des Objets)

Objet	Type/Nature	Rôle
N	Entier	Entier à saisir
U	Entier	Unité
D	Entier	Dizaine
C	Entier	Centaine
S	Entier	Somme de 3 chiffres

Algorithme :

- 0) Début** somme
 1) Ecrire("Donner un entier composé de 3 chiffres :"), lire(n)
 2) $c \leftarrow n \text{ div } 100$
 3) $d \leftarrow (n \text{ mod } 100) \text{ div } 10$
 4) $u \leftarrow n \text{ mod } 10$
 5) $s \leftarrow u+d+c$
 6) Ecrire(" la somme de 3 chiffres est = ", s)
 7) Fin Somme

Traduction en Pascal :

```

Program somme ;
uses winCRT ;
var
n,u,d,c,s : integer ;
Begin
Writeln('Donner un entier composé de 3 chiffres :)') ;
readln(n) ;
c := n div 100 ;
d := (n mod 100) div 10 ;
u:= n mod 10 ;
s := u+d+c ;
writeln(' la somme de 3 chiffres est = ', s) ;
end.

```

2^{ème} méthode :**Préanalyse :**

But : Afficher la somme de 3 chiffres d'un entier

données : n

Formules : convch(n,ch)

valeur(ch[1],c,e) valeur(ch[2],d,e) valeur(ch[3],u,e)
 $s \leftarrow u+d+c$

Analyse :

Nom : Somme

Résultat= Ecrire(" la somme de 3 chiffres est = ", s)

$s \leftarrow u+d+c$

valeur(ch[1],c,e)

valeur(ch[2],d,e)

valeur(ch[3],u,e)

convch(n,ch)

n=donnée("Donner un entier composé de 3 chiffres :")

Fin Somme

T.D.O (Tableau de Déclaration des Objets)

Objet	Type/Nature
n, u, d, c, s	Entier
ch	Chaîne de 3 caractères
e	entier

Algorithme :**0) Début** somme

1) Ecrire("Donner un entier composé de 3 chiffres :"), lire(n)

2) convch(n,ch) ;

3) valeur(ch[1],c,e)

4) valeur(ch[2],d,e)

5) valeur(ch[3],u,e)

6) $s \leftarrow u+d+c$

7) Ecrire(" la somme de 3 chiffres est = ", s)

8) Fin Somme

Traduction en Pascal :**Program** somme ;**uses** wincrt ;**var**

n,u,d,c,s, e : integer ;

ch : string[3] ;

Begin

Writeln('Donner un entier composé de 3 chiffres :) ;

readln(n) ;

str(n,ch) ;

val(ch[1],c,e) ;

val(ch[2],d,e) ;

val(ch[3],u,e) ;

s := u+d+c ;

writeln(' la somme de 3 chiffres est = ', s) ;

end.**Exercice2 :**

Ecrire un programme pascal qui permet de calculer la surface d'un cercle.

Remarques $=\pi*r^2$ **Corrigé :****program** surface ;**uses** wincrt ;**const** pi=3.14 ;**var**

r,s:real ;

begin

writeln('Donner le rayon du cercle=') ;

readln(r) ;

s:=pi*r*r ;

writeln('La surface du cercle est= ',s :7:3) ;

end .

Activité 10 : page 67

Activité 10

Écrire en Pascal le programme intitulé DATE, qui initialise la date maintenue par le système d'exploitation, en utilisant la procédure SetDate (...) de la bibliothèque WinDos.

NB.

- La syntaxe de la procédure est : `procedure SetDate(Annee, Mois, jour);`
- les paramètres valides sont 1980..2099 pour Année, 1..12 pour Mois et 1..31 pour Jour. Si la date est invalide, la demande est ignorée.

Traduction en Pascal

```
PROGRAM DATE;  
USES WinCrt, WinDos;  
VAR  
    a, m, j: WORD;  
  
BEGIN  
    WRITE ('Entrez l'année : '); READLN ( a );  
    WRITE ('Entrez le mois : '); READLN ( m );  
    WRITE ('Entrez le jour : '); READLN ( j );  
    SetDate(a, m, j);  
END.
```

Remarque :

le type WORD est le type entier non signé ; le domaine de définition est de 0 à 65535.

Exercice 6 : page 68

Exercice 6

Pour créer un répertoire, on utilise la procédure prédéfinie Mkdir(Path: string) de l'unité DOS.

Écrire un programme en Pascal, qui permet de créer un répertoire ; le nom du répertoire et le chemin sont saisis au clavier

Exemple : `Mkdir('C:\TEST');` permet de créer le répertoire 'test' sous la racine C :.