

Sommaire

Exercices Supplémentaires : Niveau 01.....	2
Exercice 1 : Affichage.....	2
Exercice 2 : Calcul (affectations : initialisation, calcul et affichage).....	3
Exercice 3 : Affichage et Lecture.....	4
Exercice 4 : Exemples de problèmes.....	5
1- Valeur absolue et carré.....	5
2- Surface et Périmètre d'un cercle.....	6
3- Permutation de valeurs de deux variables.....	7
4- Permutation de valeurs de trois variables.....	8

L'objectif de cette première série d'exercices supplémentaires est de comprendre les notions de bases de la programmation en utilisant le langage de programmation PASCAL

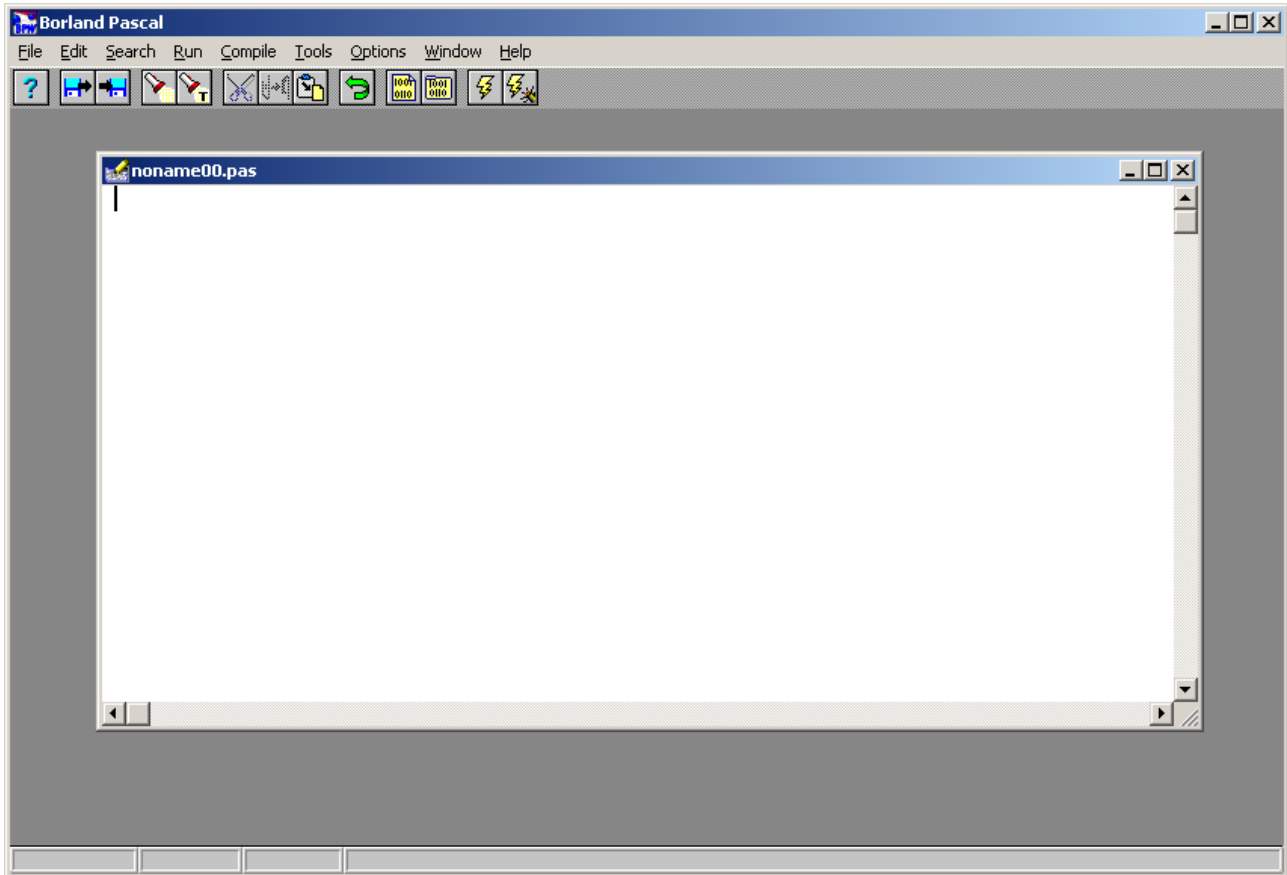
Vous allez aborder les points suivants :

- La structure d'un programme / algorithme : entête, déclaration et corps (instructions)
- Notion d'affectation
- Affichage avec l'instruction *write / writeln*
- La lecture de donnée avec l'instruction *read / readln*
- Exercice 4 : comment réaliser un programme à partir d'un énoncé de problème
- Comment corriger les erreurs syntaxiques et / ou les erreurs logiques

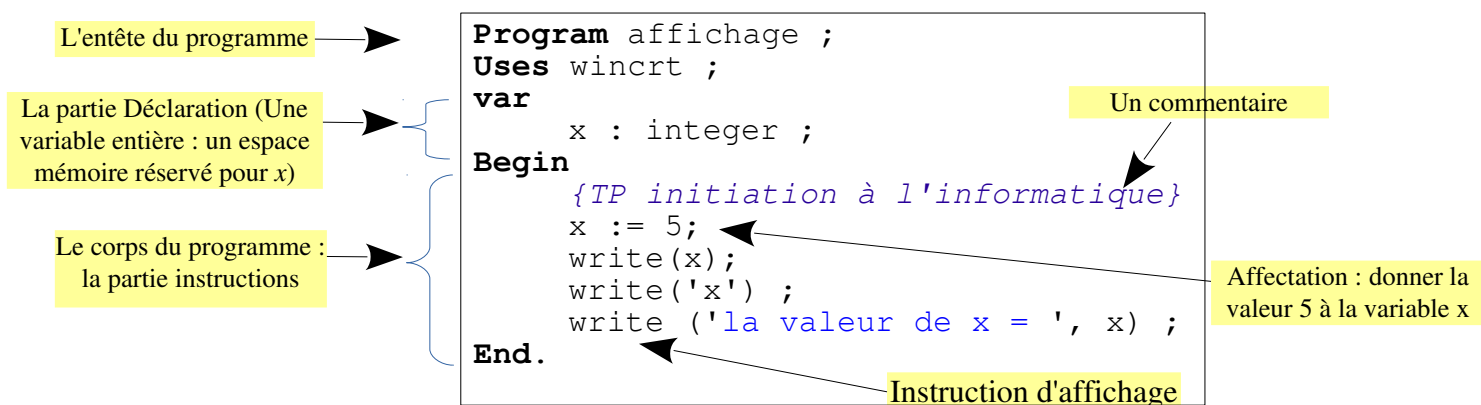
Exercices Supplémentaires : Niveau 01

Exercice 1 : Affichage

1. Démarrer l'éditeur de Borland Pascal (BPW) puis créer un nouveau fichier vide (File / New)



2. Taper le programme PASCAL suivant (la saisie du programme) :



(les info-bulles jaune et les flèches sont des illustration uniquement, il faut pas les dessiner)

- Enregistrer le programme (avec la commande File / Save)
- Compiler puis exécuter le programme (Compile / Compile et Run / Run respectivement *ou bien Alt + F9 et Ctrl + F9*)
- Remplacer l'instruction *write* par *writeln* : que se passe t-il ? (*writeln* = *write* + saut de ligne)

Remarques :

- *write (x)* : afficher la valeur enregistrée dans la variable x – tous court afficher la valeur de x
- *write ('x')* : afficher le caractère x (tous ce qui est entre apostrophes simples (non double) est affiché comme est écrit)
- *write ('la valeur de x = ', x)* : affiche deux objets, le premier c'est la chaîne 'la valeur de x = ' et le second et la valeur de x. Remarque que les deux objets sont séparés par *virgule*.
- D'une manière générale, pour afficher à l'écran la valeur d'un objets (une valeur numérique, un caractère, une chaîne de caractère, une variable et même une expression) on utilise l'instruction *write* ou bien *writeln*. Cette affichage sera réalisé **lors de l'exécution du programme**.
- Pour chaque variable déclarée, une zone mémoire (*Mémoire vive*) sera réservée pour cette variable.

Exercice 2 : Calcul (affectations : initialisation, calcul et affichage)

Soit le programme PASCAL suivant (réaliser la saisie du programme) :

```

1 Program calcul;
2 Uses winCRT ;
3 var
4     t, x, y, z : integer ;
5     f : real ;
6 Begin
7     x:= 20 ; y:= 10 ; z:= 5 ;
8     t:= x + y ;
9     f:= t/z;
10    writeln ('t = ', t) ;
11    write ('f = ', f) ;
12 End.
```

- Compiler puis exécuter le programme
- Modifier l'instruction *write ('f = ', f) ;* par l'instruction *write ('f = ', f:6:2) ;* (la ligne n° 11). Qu'est-ce-que vous remarquer après l'exécution – comment la valeur de f est

affichée ? Ça signifie quoi **:6:2** ajouté après la variable f . (6 cases pour affiche f avec 2 après la virgule)

- Modifier le type de la variable f en lui donnant le type *integer* (entier). Compilez et vous corrigé les erreurs de la ligne n° 9 (remplacer la division / par le mot clé **div**) et la ligne n° 11 (enlever **:6:2** ajouté précédemment, uniquement pour les réelle et f est devenu entier)
- Effacer l'instruction n°8 ($t := x + y ;$) et remplacer l'instruction 10 (`writeln('t=', t) ;`) par l'instruction : `writeln ('t=', x+y) ;`. Que se passe t-il lors de la prochaine exécution et pourquoi ?

Remarques :

- Ce programme contient cinq variables : 4 variables entières (t, x, y et z) et une réelle (f)
- On peut écrire plusieurs instructions sur la même ligne (comme la ligne n° 7), à condition de terminer chacune d'elles par un **point-virgule** (chaque instruction se termine par un point-virgule)
- Une affectation peut être une initialisation, une incrémentation, une décrémentation ou un calcul générale. Dans le programme, les trois premières affectations (ligne n° 7) représentent des initialisations, par contre, les deux autres affectations (lignes n° 8 et n° 9) représentent des calculs.

Exercice 3 : Affichage et Lecture

Soit le programme suivant :

read(x) veut dire : donnez une valeur à x à travers le clavier pendant l'exécution. (il faut valider par la touche entrée). Bien évidemment, on connaît pas la valeur au préalable

```

1 Program affichage_lecture;
2 Uses wincrt ;
3 var
4     x, y, z : integer ;
5 Begin
6     write('introduire un nombre entier : ') ;
7     read(x) ;
8     write('introduire deux nombres entiers : ') ;
9     read(y, z) ;
10    write ('x = ', x, 'y = ', y, 'z = ', z) ;
11 End.
```

- Compiler puis exécuter le programme (bien sûr après l'avoir saisi)
- Remplacer le corps du programme par les instructions suivantes :


```

write('introduire trois nombres entiers : ') ;
read(x, y, z);
write ('x = ', x, 'y = ', y, 'z = ', z) ;
```
- Exécuter de nouveau le programme.

Remarques

- L'instruction **read** permet d'introduire des valeurs à travers le clavier pendant l'exécution et

de la mettre dans une variable (donc *read* concerne uniquement les variables). *Read(x)* veut dire : introduire une valeur entières (puisque *x* est une variable entière) à travers le claviers et valider par la touche entrée, la valeurs saisie sera enregistrée dans l'espaces mémoire réservé pour la variable *x*.

– On peut lire plusieurs variables avec une seule instruction *read*, comme l'instruction de la ligne n° 9 : `read (y, z) ;`

Exercice 4 : Exemples de problèmes

Écrire un algorithme (et programme PASCAL) pour chacune des opérations suivante :

1. Lit un nombre réel, calcule et affiche sa valeur absolue et son carré.
2. Calcule la surface et le périmètre d'une cercle et les affiche. (Déclarer PI comme une constante).
3. Lit deux réels X et Y, réalise leur permutation en mémoire et affiche les nouvelles valeurs de X et Y.
4. Lit trois réel X, Y et Z, permute ces valeurs de telle sorte que X passe dans Z, Y passe dans X et Z passe dans Y et affiche les nouvelles valeurs de X, Y et Z.

Solution

1- Valeur absolue et carré

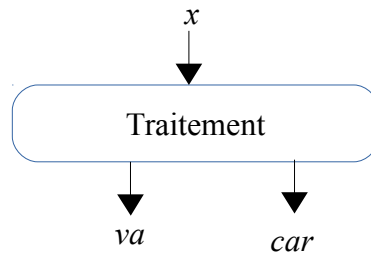
Avant de faire un programme PASCAL, il faut tout abord analyser le problème, le formaliser et écrire ensuite un *algorithme* (étapes à suivre pour avoir la solution du problème). La dernière étape et de traduire l'algorithme en programme PASCAL. Donc, le programme n'est que le résultat de traduction de l'algorithme (en choisissant un langage de programmation – pour vous ça sera PASCAL).

Pour ce problème, on veux calculer la valeur absolue et le carré d'un nombre réelle. Pour l'algorithmique on doit déclarer une variable réelle *x* qui contiendra cette valeur. Et Bien sûr, on aura besoins de deux autres variables réelles qui contiendrons respectivement la valeur absolue et le carré (on suppose que ces deux variables sont nommée *va* et *car*). Dans la logique de déroulement, on doit tout d'abord donner la valeur à *x* et calculer après *va* et *car* et en fin afficher les valeurs de *va* et *car*. (on dit alors que *x* et une variable d'entrée et, *va* et *car* sont des variables de sortie)

Ensuite (après avoir déterminer les variables d'entrée et les variables de sorties), il faut indiquer

comment l'algorithme déduit (calcule) les variables de sorties à base des variables d'entrée.

Dans ce problème (qui est facile), on fait tout simplement $car \leftarrow sqr(x)$ et $va \leftarrow abs(x)$ (C'est la formalisation)



L'algorithme :

```

Algorithme val_absolue_carre
  Variables
    x, va, car : réel
  Début
    Lire (x)
    va ← abs(x)
    Car ← x * x
    Écrire(va, car)
  Fin
  
```

Le programme PASCAL :

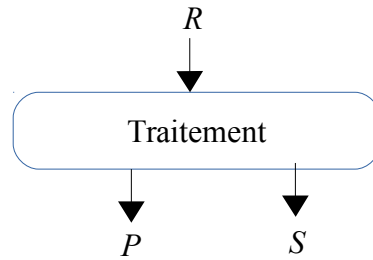
```

1 Program val_absolue_carre;
2 Uses wincrt ;
3 var
4   x, va, car : integer ;
5 Begin
6   write('introduire un nombre réel : ') ;
7   read(x) ;
8   va:= abs(x) ;
9   car:= sqr(x) ; {on bien : car:=x*x ;}
10  write ('valeur absolue = ', va:6:2, 'carré = ', car:6:2) ;
11 End.
  
```

2- Surface et Périmètre d'un cercle

Calcul de la surface et du périmètre d'un cercle : un cercle est défini par son centre et son rayon. Soit P et S (respectivement) le périmètre et la surface d'un cercle dont le rayon égale à R. On a : $P = 2 \pi R$ et $S = \pi R^2$. (des formules mathématique que vous devez connaître).

On déduit automatiquement que R est une variable d'entrée et, P et S sont des variables de sortie. Comme schématiser par la figure suivante :

**L'algorithmme :**

```

Algorithme exo_4_2
  constantes      PI = 3.14
  Variables
    x, va, car : réel

  Début
    Lire (R)
    P ← 2*PI*R
    S ← PI*R*R
    Écrire(P, S)

  Fin
  
```

Le programme PASCAL :

```

1 Program val_absolue_carre;
2 Uses wincrt ;
3 const
4   PI = 3.14;
5 var
6   R, P, S : integer ;
7 Begin
8   write('introduire le rayon du cercle : ') ;
9   read(R) ;
10  P:= 2*PI*R ;
11  S:= PI*sqr(R) ; {on bien : S:=PI*R*R ;}
12  write ('Surface = ', S:6:2, 'Périmètre= ', P:6:2) ;
13 End.
  
```

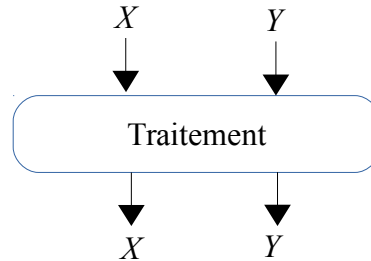
3- Permutation de valeurs de deux variables

Permutation de deux variables : Permuter (échanger) les valeurs de deux variables réelles X et Y . Si, par exemple, $X=5$ et $Y=-7$ on aura à la fin du programme $X=-7$ et $Y=5$. (Il faut réaliser cette permutation dans la mémoire centrale).

La solution qui vient à l'esprit immédiatement est de mettre la valeur de y dans x et celle de x dans y et ceci avec les deux affectations : $x \leftarrow y$ et $y \leftarrow x$. Cependant, cette solution a un problème : la valeur initiale de x sera écrasée par celle de y . Si par exemple $x = 5$ et $y = -7$, avec cette solution

on aura : $x=-7$ et $y=-7$. Puisque la valeur 5 est perdue (écrasée). Il faut corriger ce problème avec la sauvegarde de la valeur initiale de x dans une autre variable (par exemple t). Donc, la suite d'instructions est : $t \leftarrow x$; $x \leftarrow y$; $y \leftarrow t$;

Pour les variables d'entrée et de sortie, il sont représentées par le schéma suivant :



L'algorithme :

```

Algorithme exo_4_3
  Variables
    x, y, t : réel
  Début
    Lire (x, y)
    t ← x
    x ← y
    y ← t
    Écrire(x, y)
  Fin
  
```

Le programme PASCAL :

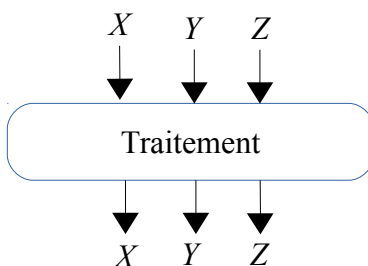
```

1 Program exo_4_3;
2 Uses wincrt ;
3 var
4   x, y, t : integer ;
5 Begin
6   write('donnez deux valeurs réelles : ') ;
7   read(x, y) ;
8   t:= x ; {on conserve la valeur de x dans t}
9   x:= y ; {pas de risque de perte de valeur}
10  y:= t ; {on récupère l'ancienne valeur de x}
11  write ('X= ', x:6:2, 'Y = ', y:6:2) ;
12 End.
  
```

4- Permutation de valeurs de trois variables

Permutation de trois variables : Comme la question précédente, on veut permuter les valeurs de trois variables réelles X , Y et Z de telle sorte que la valeur de X passe dans Z , valeur de Y passe dans X et celle de Z passe dans Y . La même chose que l'exercice précédent, on a besoin d'une variable

intermédiaire t qui permet de conserver la valeur d'une variable (ça dépend de la variable avec laquelle on commence l'affectation).



L'algorithme :

```

Algorithme exo_4_4
  Variables
    x, y, z, t : réel
  Début
    Lire (x, y, z)
    t ← z
    z ← x
    x ← y
    Y ← t
    Écrire(x, y, z)
  Fin
  
```

Le programme PASCAL :

```

1 Program val_absolue_carre;
2 Uses wincrt ;
3 var
4   x, y, t : integer ;
5 Begin
6   write('donnez trois valeurs réelles : ') ;
7   read(x, y) ;
8   t:= z ; {on conserve la valeur de z dans t}
9   z:= x ;
10  x:= y ;
11  Y:= t ;
12  write ('X= ', x:6:2, 'Y = ', y:6:2, 'Z = ', z:6:2) ;
13 End.
  
```

Question :

Réaliser les *organigrammes* pour tous les programmes (ou algorithmes) réalisés dans cette série.