

TP Informatique 2

Série de TP N°3 – Sous programmes : Fonctions et Procédures

Exercice N°01 :

Soit le programme pascal suivant :

```

Program Exo_1;
Var x, y : integer;           { variables globales }
Procedure Proc1(A: integer; Var B: integer); {SP1}
Begin
  A := A + 1; B := 22;
End;                               {Fin SP1}
Procedure Proc2(A: integer; B : integer); {SP2}
Begin
  A := A + 1; B := 22;
End;                               {Fin SP2}
Begin                               {PP}
  x := 3; y := 7;
  Proc1(x, y);                     { Appel au SP1 }
  Writeln('x= ', x, ' y = ', y);
  x := 3; y := 7;
  Proc2(x, y);                     { Appel au SP2 }
  Writeln('x= ', x, ' y = ', y);
End.                               {Fin PP}

```

Questions :

- 1) Exécuter le programme.
- 2) Quelle est la différence entre les deux procédures Proc1 et Proc2 ?
- 3) Quels sont les paramètres à passage par valeur et ceux à passage par variable ?
- 4) Quels sont les paramètres formels des deux procédures ? les paramètres effectifs ?
- 5) Les appels Proc1(3,7) et Proc2(3,7) sont-ils corrects, expliquer pourquoi.
- 6) Dérouler le programme.
- 7) Exécuter le programme en donnant le type « réel » à la variable y. Que se passe-t-il ? Pourquoi ?

Remarque :

On définit les acronymes SP et PP comme suit :

SP : Sous-Programme,

PP : Programme Principal

Exercice N°02 :

Soit le programme pascal suivant :

```

Program exo_2;
Var N: integer; Fact: integer; { Variables globales }
Function Factoriel (m: integer): integer; {SP}
Var j, f: integer;           { Variables locales au SP }
Begin
  f := 1;
  For j := 1 to m do
    f := f*j;
  Factoriel := f;              { Résultat retourné }
End;                         {Fin SP}
Begin                           {Début PP}
  Write('Introduire N : ');
  Read(N);
  Fact := Factoriel(N);       { Appel au SP }
  Write('Le factoriel de ', N, ' = ', Fact);
End.                           {Fin PP}

```

Questions :

- 1) Exécuter le programme pour N = 6.
- 2) Réécrire le programme pour calculer la somme S suivante :

$$S = \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{N!} \quad N \geq 1$$
- 3) Exécuter le programme pour N = 3.
- 4) Dérouler le programme pour N = 3.
- 5) Réécrire le programme modifié en remplaçant la fonction par une procédure de même nom.

Exercice N°03 :

Un nombre parfait est un nombre entier positif supérieur à 1 qui est égal à la somme de ses diviseurs excepté lui-même (on ne compte pas comme diviseur le nombre lui-même).

- 1) Écrire une fonction « **Parfait** » qui vérifie si un nombre entier A est parfait ou non.
- 2) En utilisant la fonction, écrire un programme Pascal pour chercher les nombres parfaits entre 1 et 100.

Exemple : 6 et 28 sont des nombres parfaits car : $6 = 1 + 2 + 3$, $28 = 1 + 2 + 4 + 7 + 14$

TP Informatique 2

Série de TP N°3 – Sous programmes : Fonctions et Procédures

Exercices supplémentaires

Exercice 01-Sup :

Écrire un programme Pascal qui permet de vérifier, à l'aide de sous-programmes « **oppose** » et « **inverse** », si deux nombres sont opposés ou non et s'ils sont inverses ou non, respectivement.

Deux nombres sont opposés si leur somme est égale à 0.

Deux nombres sont inverses si leur produit est égal à 1.

Exercice 02-Sup :

1) Écrire une fonction « **Puiss** » qui calcule la puissance énième de X (X^N) :

$$\text{Puiss}(X,N) = \begin{cases} 0 & \text{Si } X = 0 \\ 1 & \text{Si } N = 0 \\ X \times X \times \dots \times X \text{ (N fois)} & \text{Si } N > 1 \end{cases}$$

2) En utilisant la fonction « **Puiss** », écrire un programme Pascal qui permet de calculer la somme S :

$$S = X^1 + X^3 + X^5 + \dots + X^{2N+1}$$

Exercice 03-Sup :

1) Écrire une procédure « **occurrences** » qui calcule le nombre d'occurrences d'un chiffre ($0 \leq \text{chiffre} \leq 9$) dans un tableau T de N éléments entiers.

Exemples:

Le nombre d'occurrences du chiffre 7 dans T =

7	7	8
---	---	---

 est 2.

Le nombre d'occurrences du chiffre 1 dans T =

7	7	8
---	---	---

 est 0.

2) Écrire la procédure, l'insérer dans le programme et afficher les résultats dans le programme principal.

Exercice 04-Sup :

- 1) Écrire une procédure « **Max_vecteur** » qui calcule le plus grand élément du tableau T de K composantes réelles.
- 2) En utilisant la procédure « **Max_vecteur** », écrire un programme Pascal qui permet de calculer et afficher le plus grand élément de chaque colonne d'une matrice M carrée d'ordre N.

Exercice 05-Sup :

- 1) Écrire une procédure « **Décaler** » qui permet de décaler de manière circulaire à droite les éléments d'un vecteur T de M composantes entières.
- 2) En utilisant la procédure « **Décaler** », écrire un programme Pascal qui permet de lire les éléments d'une matrice A de taille N×M, décaler de manière circulaire à droite les éléments de chaque ligne et afficher la matrice résultat.