

TP Informatique 2

Série de TP N°2 – Procédures et fonctions

Exercice N°01 :

Soit le programme PASCAL suivant :

```
Program Exo_1;
Var a, b, c : real; {Variables globales du programme}

Procedure sous_prog1(x:real ; y:real ; var s:real) ;
Begin
  s := x+y ;
End;

Procedure sous_prog2(x:real ; y:real ; s:real);
Begin
  s := x+y ;
End;

BEGIN {Début du programme principal}
  a:=10; b:=5; c:=0;
  sous_prog1 (a, b, c);
  writeln('La somme est : ', c:1:0);
  a:=10; b:=5; c:=0;
  sous_prog2 (a, b, c);
  writeln('La somme est : ', c:1:0);
END. {Fin du Programme Principal}
```

Questions :

- 1) Exécuter le programme.
- 2) Quelle est la différence entre les deux procédures sous_prog1 et sous_prog2 ?
- 3) Quels sont les paramètres à passage par valeur et ceux à passage par variable ?
- 4) Quels sont les paramètres formels des deux procédures ?
- 5) Quels sont les paramètres effectifs ?
- 6) Dérouler le programme.
- 7) Exécuter le programme en donnant le type « integer » à la variable c. Que se passe-t-il ? pourquoi ?

Exercice N°02 :

Soit le programme PASCAL suivant :

```
Program Exo_2;
Var i, n: integer;
S: real; {Variables globales du programme}
FUNCTION Fact (m: integer): integer;
Var j, f : integer; {Variables locales de la fonction Fact}
Begin
  f:=1;
  for j:=1 to m do
    f:= f*j;
  Fact := f; {Une fonction se termine toujours
             par une affectation du résultat}
End;

BEGIN {Début du programme principal}
  write('Introduire n : ');
  read(n);
  S:=1;
  for i:=1 to n do
    S:=S+Fact(i);
  write('La somme = ',S:6:2);
END. {Fin du programme principal}
```

- 1) Exécuter le programme pour n = 3.
- 2) Dérouler le programme pour n = 3.
- 3) Réécrire le programme en transformant la fonction Fact en une procédure Fact.
- 4) Soit la procédure « Puiss » suivante :

```
Procedure Puiss (t: real; k: integer; var P: real);
Var i: integer;
Begin
  P:=1;
  for i:=1 to k do
    P:=P*t;
End;
```

- 4-a) Dérouler la procédure Puiss pour t=2 et k=4 et déduire ce qu'elle fait.
- 4-b) Transformer la procédure Puiss en une fonction.
- 5) On propose de calculer la valeur approximative de l'exponentiel de x avec la formule suivante :

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

En utilisant les deux fonctions Fact et Puiss, écrire un programme Pascal pour calculer e^x .

TP Informatique 2

Série de TP N°2 – Exercices supplémentaires sur les procédures et fonctions

Exercice N°01 :

- Écrire une fonction « Max » qui calcule le maximum entre deux nombres réels X et Y.
- En utilisant la fonction « Max », écrire un programme Pascal permettant de calculer et afficher le maximum entre trois nombres réels X, Y et Z.

Exercice N°02 :

Soit la séquence d'instructions suivante qui permet de calculer la somme « S » des diviseurs d'un nombre entier « X ».

```
S :=0 ;  
For i := 1 to X do  
  If (X mod i) = 0 then  
    S := S+i;
```

- 1) Écrire un sous-programme fonction qui s'appelle « somme_div » et qui calcule la somme « S » des diviseurs d'un nombre entier « X ».
- 2) Insérer cette fonction dans un programme Pascal qui lit un entier « X », fait appel à la fonction « somme_div » et affiche la somme de ses diviseurs « S ».
- 3) Réécrire le programme en transformant la fonction « somme_div » en procédure « somme_div ».
- 4) Modifier les instructions de la fonction pour qu'elle retourne la somme des diviseurs de « X » sauf 1 et lui-même.

Exercice N°03 :

- Écrire un programme qui lit un **tableau** T de N réels, fait appel à une **procédure** qui détermine le plus grand élément du tableau ainsi que sa position (son rang dans le tableau). Écrire cette procédure et l'insérer dans le programme. Afficher les résultats dans le programme principal.

Exercice N°04 :

- Écrire une procédure « Permuter » qui réalise la permutation entre deux réels X et Y.
- En utilisant la procédure « Permuter », écrire un programme Pascal permettant de permuter les deux diagonales d'une matrice carrée A(n×n) de type réel.

Exercice N°05 :

- Écrire en Pascal une fonction « Nombre_Premier » qui vérifie si un nombre entier N est premier ou non.
N.B : Un nombre est dit « premier » s'il est divisible uniquement par 1 et lui-même.
- Insérer la fonction « Nombre_Premier » dans un programme Pascal complet permettant de lire une matrice A (n×m) de type entier et de calculer et afficher le **nombre total des nombres premiers** présents dans la matrice A.
- Réécrire le programme précédent en transformant la fonction « Nombre_Premier » en une procédure du même nom.