

## TP - Programmation

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

#### Exercice N°01 :

Soit le programme C suivant :

```
#include <stdio.h>

void sous_prog1(float x, float y, float s) {
    s = x + y;
}
void sous_prog2(float x, float y, float* s) {
    *s = x + y;
}
int main() //Début du programme principal
{
    float a, b, c; //Variables globales du programme

    a = 10; b = 5; c = 0;
    sous_prog1(a, b, c);
    printf("La somme est : %.2f \n", c);

    a = 10; b = 5; c = 0;
    sous_prog2(a, b, &c);
    printf("La somme est : %.2f \n", c);

    return 0;
} //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 « int » à la variable c. Que se passe-t-il ? pourquoi ?

#### Exercice N°2 :

Écrire une procédure qui prend deux entiers en entrée et qui échange leurs valeurs.

#### Exercice N°03 :

Soit le programme C suivant :

```
#include<stdio.h>
int maximum(int a, int b)
{
    if (a>b)
        return a ;
    else
        return b ;
}
int main() //Début du programme principal
{
    int x, y, results ; //Variables globales du programme
    printf("Donner la première valeur : ") ;
    scanf("%d", &x) ;
    printf("Donner la deuxième valeur : ") ;
    scanf("%d", &y) ;
    results= maximum (x, y) ;
    printf("Le maximum entre %d et %d est : %d \n ", x, y,
        results) ;
    return 0 ;
} //Fin du programme principal
```

#### Questions :

- 1) Exécuter le programme pour x = 5 et y = 3.
- 2) Dérouler le programme pour x = 5 et y = 3.
- 3) Quels sont les paramètres formels de la fonction ?
- 4) Quels sont les paramètres effectifs ?
- 5) Réécrire le programme pour déterminer le maximum entre trois nombres entiers x, y et z.

#### Exercice N°4 :

Écrire une fonction qui calcule la somme de deux nombres entiers passés en paramètres et retourne le résultat. Écrire un programme qui utilise cette fonction pour calculer et afficher la somme de deux nombres saisis par l'utilisateur.

## TP - Programmation

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

#### Exercice N°05 :

Soit le programme C suivant :

```
#include <stdio.h>
int Fact(int m) {
    int j, f;
    f= 1;
    for(j=1; j<= m; j++) {
        f =f* j;
    }
    return f;
}
int main() //Début du programme principal
{
    int i, n; float S; //Variables globales du programme
    printf("Introduire n : ");
    scanf("%d", &n);
    S=0;
    for (i =1; i<= n; i++) {
        S =S+ Fact(i);
    }
    printf("La somme = %.2f", S);
    return 0;
} //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 :

```
void Puiss(float t, int k, float* P) {
    int i;
    *P=1;
    for(i = 1; i <= k; i++) {
        *P =*P*t;
    }
}
```

- 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 en langage C pour calculer  $e^x$  .

## TP - Programmation

---

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

---

#### Exercices supplémentaires

##### Exercice Sup-01 :

Écrire un programme C qui calcule la somme  $e = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$ , Sachant que le programme fait appel à une **fonction** FACT(n) pour calculer n!. Écrire cette fonction et l'insérer dans le programme. Afficher le résultat dans le programme principal. (Le résultat doit produire la valeur de l'exponentielle).

##### Exercice Sup-02 :

Écrire un programme C 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 Sup-03 :

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

##### Exercice Sup-04 :

- Écrire en C 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 C 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.