

TP-Initiation à la Programmation

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

Exercice N°1 : Soit le programme suivant :

```
#include <stdio.h>
int x=3, y=7 ;
void Proc1 (int A, int *B)
{
    A = A + 1; *B = 22;
}
void Proc2 (int A, int B)
{
    A = A + 1; B = 22;
}
int main() //Début du programme principal
{
    x=3, y=7 ;
    Proc1(x, &y); //Appel à la procédure Proc1
    printf("x = %d y = %d\n", x, y);
    x=3, y=7 ;
    Proc2(x, y); //Appel à la procédure Proc2
    printf("x = %d y = %d\n", x, y);
    return 0;
} //Fin du programme principal
```

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) Quels sont les variables globales et locales dans le programme ?
- 6) Les appels **Proc1(3,7)** et **Proc2(3,7)** sont-ils corrects, expliquer pourquoi.
- 7) Dérouler le programme.
- 8) Exécuter le programme en donnant le type « réel » à la variable y. Que se passe-t-il ? Pourquoi ?

Exercice N°2 : Soit le programme suivant :

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

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.
- 6) Réécrire le programme en remplaçant la fonction

Exercice N°3 :

Écrire un programme en C qui permet de lire un tableau **T** de **N réels**. Le programme doit faire appel à une procédure qui détermine le plus grand élément du **tableau T** ainsi que sa position.

Exercice N°4 :

Écrire un programme en C qui lit les valeurs d'une matrice **M** (**N × N**) de type **entier**, comportant une fonction qui permet de retrouver le nombre de fois qu'un nombre (choisi par l'utilisateur) apparaît ainsi que ses positions.

TP-Initiation à la Programmation

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

Exercice Sup N°1 :

Écrire un programme en langage C qui permet de calculer la **moyenne** des notes de **10** étudiants. Le programme doit :

1. Demander à l'utilisateur de saisir les notes des 10 étudiants.
2. Stocker ces notes dans un tableau.
3. Utiliser une **fonction** pour calculer la moyenne des notes.
4. Utiliser une **procédure** pour insérer les notes dans le tableau.
5. Afficher la **moyenne** calculée à la fin.

Exercice Sup N°2 :

- 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 en langage C qui permet de calculer et afficher le plus grand élément de chaque colonne d'une matrice **M** carrée d'ordre **N**.

Exercice Sup N°3 :

Écrire un programme en C qui effectue le calcul de la somme **S** en utilisant deux fonctions distinctes :

- Une fonction qui calcule le factoriel d'un nombre entier.
- Une fonction qui calcule la puissance d'un nombre réel élevé à un exposant entier.

Le programme doit ensuite utiliser ces deux fonctions pour calculer la **somme S** suivante:

$$S = -\frac{2!}{x^3} + \frac{3!}{x^4} - \frac{5!}{x^6} + \frac{7!}{x^8} - \frac{11!}{x^{12}} + \dots - \frac{N!}{x^{(n+1)}}$$

Avec : **N** est un nombre entier positif non nul et **x** un nombre **flottant (reel)** non nul.

Exercice Sup N°4 :

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 Sup N°5 :

Écrire un programme en C qui lit les 10 valeurs entières d'un tableau à une dimension, comportant une procédure permet d'ordonner le tableau dans l'ordre croissant.

Exercice Sup N°6 :

Écrire un programme en C qui permet de calculer **N!**, en utilisant la fonction **fact**. Ensuite réécrire le programme en remplaçant la fonction par une procédure.