

LES FONCTIONS

Objectif:

- ① Définir ce qu'est une fonction.
- ② Connaître les différentes manières d'usage des fonctions.
- ③ Introduire la notion de la récursivité.

Pré requis:

Les Chapitres précédents de ce cours de langage C

PLAN

I/ INTRODUCTION

II/ DEFINITION D'UNE FONCTION

III/ APPEL D'UNE FONCTION

VI/ PASSAGE D'ARGUMENTS A UNE FONCTION

V/ LA RECURSIVITE

LES FONCTIONS

I/ INTRODUCTION

Une fonction est une partie de programme destinée à remplir une certaine tâche bien déterminée.

Tout programme C est constitué d'une fonction au moins, appelée main. Son exécution commence toujours par main, on appelle tous les autres fonctions directement ou en cascade.

II/ DEFINITION D'UNE FONCTION

Toute fonction est définie par son nom, le type de donnée renvoyé, ses arguments et son corps.

Chaque fonction doit avoir un prototype dont la syntaxe est la suivante :

```
type_donnée nom (type arg1, type arg2, ..., type argn) ;
```

Le prototype d'une fonction permet la déclaration de celle ci.

Exemple :

```
#include<stdio.h>
main()
{
int a, b, c ;
int max ( int x, int y) ;      /* prototype de la fonction */
printf(" a='') ;
scanf(" %d'', &a) ;
printf( " b=") ;
scanf("%d ''",&b) ;
c=max(a,b) ;
printf("max de %d et %d est %d ", a, b, c) ;
}
/*-----*/
```

```
int max (int x, int y)      /* corps de la fonction */
{
if (x>y) return (x)
else return (y) ;
}
```

III/ APPEL D'UNE FONCTION

On appelle une fonction en mentionnant son nom, suivi d'une liste d'arguments entre parenthèses et séparés par des virgules. lorsque l'appel de la fonction doit se faire sans arguments, le nom de cette dernière doit être suivi de parenthèses vides.

Exercice :

Ecrire un programme en C qui recherche le maximum de trois entiers, en utilisant les fonctions.

VI/ PASSAGE D'ARGUMENTS A UNE FONCTION

Exemple :

```
#include <stdio.h>
main()
{
int a=2 ;
void modifie( int a) ;
printf( "\n avant appel de modifie, a=%d", a) ;
modifie (a) ;
printf("après appel de modifie, a=%d", a) ;
}
/*-----*/
void modifie(int a)
{
a=10 ;
printf("Dans modifie, a= %d",a) ;
return;
}
```

Résultat d'exécution :

avant de modifie, a=2

dans modifie, a=10

après modifie, a=2

Lorsqu'une valeur doit être transmise à une fonction au moyen d'un argument réel, la valeur de ce dernier est copiée dans une fonction. La valeur de l'argument formel peut aussi être modifiée au sein de la fonction, sans que la valeur initiale de l'argument réel soit modifiée dans la partie apparente du programme.

Ce processus de passage de valeur des arguments à une fonction porte le nom de « passage par valeur ».

V/ LA RECURSIVITE

Exemple :

```
#include <stdio.h>
main()
{
int n ;
long int factorielle( int n ) ;
printf (" n=") ;
scanf("%d",&n) ;
printf("n != %d \n", factorielle(n)) ;
}
/*-----*/

long int factorielle(int n)
{
if ( n<=1) return (1) ;
else return ( n*factorielle(n-1)) ;
}
```

La récursivité est la caractéristique des fonctions capables de s'appeler elles mêmes de façon répétitives jusqu'à ce que soit vérifiée une condition donnée.

Exercice :

Ecrire un programme en C récursif qui permet d'afficher une ligne de texte inversée.

Solution :

```
#include <stdio.h>

main()
{
void renverse(void) ;
printf("Saisir une ligne de texte :\n") ;
renverse() ;
}

/*-----*/

void renverse (void)
{
char c ;
if ((c==getchar()) !='\n')
renverse() ;
putchar(c) ;
return ;
}
```

ISSET DE SFAX
II N3

TP N°2 PROGRAMMATION C

A.U 1997/1998
Cours de Mr: TAYARI Lassaad

Objectif:

Se familiariser avec la manipulation des structures de contrôles (les boucles).

Durée: 2 séances de trois heures chacune

EX N°1

Établir un algorithme puis un programme C permettant de calculer et afficher le **PGCD** de deux entiers strictement positifs.

METHODE 1 : $\text{PGCD}(a,b) = \text{PGCD}(b, a \bmod b)$ jusqu'à $(a \bmod b = 0)$.

Exemple $\text{PGCD}(54,24) = \text{PGCD}(24,6) = \text{PGCD}(6,0)$ donc $\text{PGCD}(54,24) = 6$.

METHODE 2 : $a \leftarrow a-b$ si $a > b$ et $b \leftarrow b-a$ si $b > a$ jusqu'à $a=b$.

Exemple

a	54	30	6	6	6	6
b	24	24	24	18	12	6

donc $\text{PGCD}(54,24) = 6$.

EX N°2

Établir un programme C qui calcule et affiche **Y** pour des valeurs de **X** variants entre **I** et **J** avec un pas de **D**. $Y=5X^2+8X-6$.

EX N°3

Établir un algorithme puis un programme C qui lit un entier positif et impair et affiche les quatre sommes suivantes.

$$A = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$$

$$B = 1 - \frac{1}{2} + \frac{1}{3} - \dots + \frac{1}{N}$$

$$C = 1 + \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{N}$$

$$D = -\frac{1}{2} - \frac{1}{4} - \dots - \frac{1}{N-1}$$

EX N°4

Établir un programme C permettant de lire N nombres et affiche le plus grand, le plus petit et la moyenne de ces nombres.

EX N°5

Écrire un programme C qui lit un entier strictement positif et affiche l'un des deux messages suivants.

'PREMIER'
'NON PREMIER'

EX N°6

On veut calculer et afficher la fonction F définie par: $F(X) = \sin^2(X) + \cos(2X) + \pi$. dans un intervalle [i,j] avec un pas de D. Pour ceci, établir un algorithme puis un programme C permettant de lire les valeurs I, J et D, de calculer et afficher la valeur de F après chaque pas.

EX N°7

Établir un programme qui lit un entier compris entre 0 et 15 et affiche son équivalent dans la base 16.

Bon Travail