

A3 FISA S3E – Langage C

TP 1 – Déclarations

1. INTRODUCTION

Ce TP est un rappel des notions de base de la programmation en langage C.

Ce TP nécessite l'utilisation du logiciel CodeBlocks.

2. MON PREMIER PROGRAMME

En langage C, un programme doit être sauvegardé dans un fichier source avec une extension « .c ». Ce fichier source sera compilé à l'aide d'un compilateur pour ensuite éditer les liens, si plusieurs fichiers, et enfin créer le fichier exécutable.

Voici un exemple d'un programme simple.

```
Hello.c
/* Mon premier programme en C */
#include <stdio.h>
main()
{
    Printf(« Hello world !! \n »);
}
```

Le code est sauvegardé dans un fichier qui s'appelle « Hello.c ».

On commence le programme par préciser les bibliothèques à introduire :

- Dans ce programme, on introduit la bibliothèque « stdio.h » pour standard input output header. On trouve principalement dans cette bibliothèque toutes les fonctions des entrées sorties.
 - Deux autres bibliothèques sont très souvent utilisées :
 - Stdlib.h pour tous ce qui est allocation mémoire (alloc, malloc, free).
 - Math.h pour tous ce qui est fonction mathématique : cos, sin, sqrt, pow,...
- L'en-tête « main » précise que ce qui sera décrit à sa suite est le programme principal
- Un programme est délimité par des accolades. Les instructions situées entre ses accolades forment un bloc
- Une instruction se termine toujours par le caractère « ; »
- La fonction « printf » est une fonction de sortie. C'est une fonction prédéfinie. Les guillemets servent à délimiter la chaîne de caractère qui sera affichée. Le caractère « \n » de fin de ligne, provoque le passage à la ligne suivante.
- Les commentaires se trouvent entre les caractères « /* » et « */ ». Ces éléments ne sont pas pris en compte lors de la compilation et exécution du programme.

3. LA DECLARATION DES VARIABLES

En Langage C, les déclarations des types de variables sont obligatoires et doivent être regroupées au début du programme.

Les types les plus utilisés

int pour déclarer des entiers

float pour déclarer des nombres réels

char pour déclarer des caractères

Voici un exemple qui permet de déclarer un entier « *i* », de lui donner la valeur 36 et de l'afficher.

```
#include <stdio.h>
main()
{
    int i = 36;
    printf("Comment calculer la racine carre de %d ?\n", i);
}
```

printf prend ici deux arguments :

- Le format qui est entre les deux guillemets
- Et la valeur de *i*

Le caractère % précise que le caractère suivant est un « code-format » et qu'il faut considérer la valeur reçue en argument suivant.

4. DECLARER SAISIR ET AFFICHER DES VARIABLES.

Exercice 1 : printf et scanf

- 1- Ecrivez un programme C permettant d'afficher à l'écran le texte "Bonjour !".
Compilez-le et exécutez-le.
- 2- La fonction « scanf » permet de saisir des variables à partir du clavier.
L'instruction suivante permet de saisir un entier et le sauvegarder dans la variable déjà déclaré « i » :
scanf(« %d », &i).

Les types les plus utilisés

%d pour saisir un entier

%f pour saisir un nombre réel

%c pour saisir un caractère

Modifiez le programme afin qu'il demande et affiche également votre âge.

Exercice 2 : Déclaration des variables

Ecrire un programme qui déclare la variable constante Pi et la variable R contenant la valeur 20.

Déclarez trois variables D, P et S et affecter respectivement à ces variables les valeurs du diamètre, du périmètre et de la surface d'un cercle dont le rayon est R.

Affichez à l'écran le contenu de ces différentes variables.

5. LES CONDITIONS

La condition « if » permet d'exécuter une liste d'instruction si une condition est vraie.

If (condition)

{

Liste d'instructions à exécuter si la condition est vraie

}

Else

{

Liste d'instructions à exécuter si la condition est fausse.

« Else » n'est pas obligatoire

}

Exercice 3 :

Ecrire le programme permettant de déclarer et saisir 2 entiers. Le programme doit ensuite trouver lequel des deux est plus grand.

6. LES ITERATIONS

Les boucles permettent de répéter plusieurs fois les mêmes instructions.

La boucle « while » permet de répéter les mêmes instructions tant que la condition est vraie. La boucle « while » peut s'écrire de deux façon :

1- While (condition)

```
{
    Liste d'instructions à exécuter tant que la condition est vraie
}
```

Instruction à exécuter lorsque la condition n'est plus valide.

2- Do

```
{
    Liste d'instructions à exécuter tant que la condition est vraie
}
While(condition)
Instruction à exécuter lorsque la condition n'est plus valide.
```

Quelle est la différence entre les deux écritures de la boucle while ?

La boucle « for » permet de répéter une liste d'instruction un nombre défini de fois. Pour répéter la même instruction 10 fois, la boucle « for » peut s'écrire de la façon suivante :

For(i=0 ; i<10 ; i++)

```
{
    Liste d'instructions à exécuter.
}
```

La variable « i » doit être déclarée au début du programme.

NB : i++ est une façon simplifiée pour écrire i = i+1. Elle permet d'incrémenter de 1 la valeur de i.

D'une manière plus générale, la boucle « for » s'écrit de la manière suivante :

For(initialisation ; test ; incrémentation)

```
{
    Liste d'instructions à exécuter.
}
```

Exercice 4 : Itérations, la boucle while()

Ecrire le programme permettant de calculer x^n pour des variables x et n saisies au clavier.

Exercice 5 : Itérations, la boucle while()

Ecrire le programme en C permettant la multiplication par addition successives.

