

---

## TP1 : Introduction au langage C

---

### Introduction

Le langage C a été mis au point au début des années 1970 et normalisé (ANSI) en 1988. Depuis cette date, il a conquis le monde des entreprises et des universités malgré ses quelques défauts (syntaxe parfois illisible, pas assez de contrôle sémantique, etc.). Le langage C est un langage de haut niveau qui génère un code très rapide grâce à un compilateur très performant.

Un programme écrit en langage C peut être exécuté sur n'importe quel système d'exploitation sans aucune modification. C'est pour cela qu'on dit qu'il est portable.

Le langage C n'est pas difficile à apprendre malgré son apparence car il dispose de peu d'instructions et ses structures de données sont limitées. Une fois l'apprentissage du langage C est achevé, cela nous facilite la familiarisation avec d'autres langages tels que C++ et Java qui sont dérivés du C.

### Programmer en langage C

Pour pouvoir programmer en langage C, vous auriez besoin de deux outils essentiels, à savoir : un éditeur de texte, et un compilateur.

- **L'éditeur de texte** vous offre un espace éditable pour taper une séquence de code de votre programme. Le fichier contenant le morceau de code que vous aviez introduit sur l'éditeur de texte, appelé « **fichier source** », est enregistré sous l'extension « **.c** » (qui désigne un fichier source écrit en langage C).
- **Le compilateur** traduit votre fichier source écrit dans un langage évolué proche au langage humain (C) en un programme de bas niveau que la machine peut comprendre et exécuter (le langage binaire).

### Structure d'un programme en C

Un programme en **C** est composé d'une suite d'instructions écrites en utilisant des mots clés propres au langage. La structure générale d'un programme en **C** est composée des deux parties suivantes : la partie en-tête, et la partie corps.

- **La partie en-tête** contient les bibliothèques qu'un programme a besoin pour s'exécuter. On importe une bibliothèque grâce au mot clé **#include**. Au minimum la bibliothèque **<stdio.h>** doit être importée.
- **La partie corps** contient des déclarations et des instructions.

#### Exemple :

```
#include <stdio.h> /* Partie entête*/
Main() /*Programme principale*/
{
    Int A, B, S ;
    Scanf("%i" , &A);
    Scanf("%i" , &B);
    S = A+B ;
    Printf("%i" , S);
}
```

Pour une bonne compréhension de l'exemple, nous donnons les descriptions suivantes :

- **#include <stdio.h>** : Importe la bibliothèque **stdio.h**. Plusieurs bibliothèques existent, on les importe au besoin de la même manière.
- **Blocs d'un programme** : Un « { » et un « } » constitue un bloc. Un programme en **C** peut contenir plusieurs blocs. Ceux-ci renferment toutes sortes d'instructions. Ces blocs peuvent être inclus les uns dans les autres.
- **Instructions** : Une instruction en **C** se termine souvent par un point-virgule. Les instructions sont exécutées l'une à la suite de l'autre, en débutant à partir de la première instruction du programme principal.
- **Commentaires** : Il est toujours préférable qu'un programme contienne des explications aux endroits complexes afin de faciliter leurs compréhensions. En langage C, on insère ces explications (commentaires) dans le programme en les délimitant par « /\* » et « \*/ ».

## Travail à faire

Dans le cadre de ce TP, nous vous proposons d'utiliser l'environnement de développement intégré (IDE) « **Code Blocks** ». Car, ce dernier regroupe plusieurs spécificités spécialement conçues pour les étudiants qui commencent à apprendre la programmation. Cependant, vous pouvez utiliser n'importe quel IDE de votre choix.

Code Blocks est disponible pour téléchargement gratuit via le site <https://www.codeblocks.org>

### Exercice 1 :

1. Tapez le programme suivant sur votre éditeur de texte :

```
#include <stdio.h> /* Partie entête*/
Main() /*Programme principale*/
{
    Printf("Faculté des Sciences Exactes");
    Printf("L1 Informatique");
}
```

2. Compilez le programme (en appuyant sur **Compiler** ou sur **F9**) et exécutez-le (en appuyant sur Run ou sur **F5**).
3. Remplacez « **L1 Informatique** » par « **Département d'Informatique** ». Que se passe-t-il ?
4. Rajoutez votre nom et prénom en commentaires devant la partie en-tête du programme.
5. Modifier le même programme comme suit :

```
#include /* Partie entête*/
Main() /*Programma principale*/ *
{
    int A, B, S ;
    Scanf ("%d" , &A);
    Scanf ("%d" , &B);
    S = A+B ;
    Printf ("%d" , S);
}
```

6. Compiler et exécuter ce programme. Que fait maintenant ce programme ?
7. Ajouter les instructions nécessaires pour calculer la différence, le produit et la division de A et B.

### Exercice 2 :

Compléter et traduire l'algorithme suivant qui calcule la surface A et le périmètre P d'un cercle.

```
Algorithme Cercle ;
Variables
Diam, A, P : .....
Début
  Ecrire (' Taper le diamètre : ')
  Lire (R)
  A .....
  P .....
  Ecrire ('L'aire = ', A)
  Ecrire ('Le périmètre = ', P)
Fin.
```

### Exercice 3 :

Ecrire un programme qui calcule la moyenne d'un étudiant en Algorithmique, sachant que les notes de TP, TD et EMD sont lues au clavier. Le résultat s'affiche comme suit :

La moyenne en Algorithmique est : .....

La moyenne d'un étudiant se calcule comme suit :

$$\text{Moy} = (\text{TD} + \text{TP} + 2 * \text{EMD}) / 4$$

Affichez la moyenne d'un étudiant ayant les notes suivantes : TP = 12, TD = 9 et EMD = 7 en utilisant le type **int** pour les variables TP, TD et EMD. Le résultat obtenu est-il correct ?

### Exercice 4 :

Ecrire un programme C qui calcule le carré et la racine carrée d'un nombre lu à partir du clavier.

A l'exécution, on doit avoir l'affichage suivant : (La valeur 9 est donnée comme un exemple)

```
-----
***** TP1_2 *****
-----
Taper un nombre : 9
Le carré de 9 est 81
La racine carrée de 9 est 3
```

### Exercice 5 :

Ecrire un programme C qui lit un entier N de 3 chiffres ensuite calcule et affiche le miroir de l'entier N.

Ex : Miroir de 273 est : 372.

### Exercice 6 :

Ecrire le programme qui fait la permutation de deux nombres entiers.