



TP N° 1 (une séance)

▸ Les objectifs de ce premier tp sont de comprendre et mettre en œuvre la conception des programmes simples

Exercice 1 : Premier programme

Voici une version du premier programme que l'on étudie habituellement.

Il affiche "**Bonjour tout le monde !**" à l'écran :

```
#include<stdio.h>
/* Directive de préprocesseur pour inclure Les fonctions standards d'entrée/sortie
dans Le programme */
void main(void)
{
    printf("Bonjour tout le monde !\n");
}
```

Tout programme en C doit posséder une fonction nommée **main** (principale) pour indiquer où commencer l'exécution. Une fonction est essentiellement une suite d'instructions que l'ordinateur exécutera dans l'ordre où elles sont écrites.

Une fonction comprend quatre parties :

- un **type de retour** : ici **void** (ne retourne rien) qui spécifie le genre de résultat que la fonction retournera lors de son exécution. En C, le mot **void** est un mot réservé (un mot-clé) : il ne peut donc pas être utilisé pour nommer autre chose.
- un **nom** : ici **main**
- une **liste de paramètres** entre parenthèses (que l'on verra plus tard) : ici la liste de paramètres est vide (**void**)
- un **corps de fonction** entre accolades qui énumère les instructions que la fonction doit exécuter.

Remarque : la plupart des instructions C/C++ se terminent par un point-virgule (;).

En C, les chaînes de caractères sont délimitées par des guillemets ("). "**Bonjour tout le monde !\n**" est donc une chaîne de caractères. Le code **\n** est un "caractère spécial" indiquant le passage à une nouvelle ligne.

Expérimentez avec les séquences d'échappement que vous trouvez ci-dessous et les discutez :

\n \t \b \r \" \\ \0 \a

Exercice 2 :

Ecrire un programme complet en C qui calcule la moyenne de trois nombres entiers.

Exercice 3 :

Ecrire un programme qui calcule et affiche le périmètre et la superficie du cercle dont on saisit son rayon.

Exercice 4 :

Ecrire un programme qui vérifie si un nombre pair et impair en utilisant l'opérateur de modulo.

Exercice 5 :

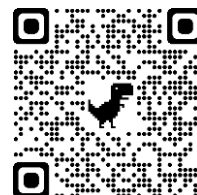
Réécrire le programme précédent en utilisant l'opérateur de modulo et l'alternative à **if - else** :

<expr1> ? <expr2> : <expr3>

Exemple : **printf("Vous avez %i an%c \n", age, (age==1) ? ' ' : 's');**

Exercice 6 :

Ecrire un programme qui résout dans IR l'équation $(a.x^2 + b.x + c = 0)$.





TP N° 2 (une séance)

Exercice 1 :

Ecrire un programme en C qui permet de convertir une information binaire en décimale. L'information binaire est un entier positif (contient uniquement des « 0 » et des « 1 ») saisi au clavier

Exemple : **1101** (*base 2*) → **13** (*base 10*)

Exercice 2 :

Ecrire un programme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne en utilisant la boucle tant que.

Exercice 3 :

Réécrire programme précédent (Exercice N° 2), en utilisant cette fois l'instruction répéter tant que

Exercice 4 :

Ecrire un programme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 6) :

Table de 6 : 6 x 1 = 6 6 x 2 = 12 6 x 3 = 18 ... 6 x 10 = 60

Exercice 5 :

Ecrire un programme permettant de déterminer si un entier naturel entré au clavier est premier.

Exercice 6 :

Ecrire un programme qui calcule la racine carrée de la valeur réel positif **a** grâce à la formule récurrente $U_{n+1} = 1/2 * (U_n + a/U_n)$. Les calculs doivent commencer avec **1** comme valeur initiale de **U₀** et s'arrêtent quand la valeur absolue de la différence entre les deux derniers valeurs calculés est inférieur strictement à **0.001**.

