



INFORMATIQUE

Sequence 2 : Boucles et fonctions

TP 2.1

v2025-09-15

IUT d'Annecy, 9 rue de l'Arc en Ciel, 74940 Annecy

BOUCLES ET PREMIÈRES FONCTIONS

Manipulation 1 : Squelette du TP

Tout d'abord, téléchargez le squelette du TP :

| wget <https://github.com/IUT-GEII-Annecy/squelettes/releases/download/branch-2025/tp3.zip>

1 Nos premières boucles While, For et Do...While

1.1 La boucle While - Miauler n fois

Manipulation 2 : Un chat qui miaule

Le programme `miauler.c` contient le code suivant, qui fait miauler un chat 3 fois :

```
#include <stdio.h>
int main(void) {
    int i = 0;
    while (i < 3) {
5         printf("Miaou !\n");
        i++;
    }
    return 0;
}
```

- ☐ Tester le programme `miauler.c`
- ☐ Modifier le programme pour qu'il miaule 5 fois.
- ☐ Modifier le programme pour qu'il miaule `n` fois, où `n` est un entier entré par l'utilisateur.

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/animaux/while

1.2 La boucle For - Aboier n fois

Manipulation 3 : Un chien qui aboie

Le programme `aboyer.c` contient le code suivant, qui fait aboyer un chien 3 fois :

```
#include <stdio.h>
int main(void) {
    for (int i = 0; i < 3; i++) {
5         printf("Ouaf !\n");
    }
    return 0;
}
```

```
| }
```

- ☐ Tester le programme aboyer.c
- ☐ Modifier le programme pour qu'il aboie 5 fois.
- ☐ Modifier le programme pour qu'il aboie n fois, où n est un entier entré par l'utilisateur.

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/animaux/for

1.3 La boucle Do...While - Chanter n fois

Manipulation 4 : Un oiseau qui chante

Le programme `chanter.c` contient le code suivant, qui fait chanter un oiseau 3 fois :

```
#include <stdio.h>
int main(void) {
    int i = 0;
    do {
5      printf("Cui-cui !\n");
        i++;
    } while (i < 3);
    return 0;
}
```

- ☐ Tester le programme chanter.c
- ☐ Modifier le programme pour qu'il chante 5 fois.
- ☐ Modifier le programme pour qu'il chante n fois, où n est un entier entré par l'utilisateur.

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/animaux/doWhile

Question : Rappeler le principe de chaque boucle et, pour chacune, dire quand elle est la plus adaptée.
Question : Tester chacun des trois programmes avec $n = 0$. Que constatez-vous? Question : Quelle boucle choisiriez-vous pour faire répéter une action un nombre inconnu de fois, mais au moins une fois? Pourquoi?

Manipulation 5 : Devine le nombre - Version While, avec n vies

Un nombre mystère est choisi au hasard entre 1 et 100. Le joueur a n vies pour deviner ce nombre.

Tout d'abord, le programme affiche un menu avec 4 niveaux de difficulté, qu'il choisira en entrant un chiffre entre 1 et 4 :

1. Facile (10 vies)
2. Moyen (7 vies)
3. Difficile (5 vies)
4. Impossible (0 vies)

Ensuite, le programme choisit un nombre au hasard entre 1 et 100, puis demande au joueur de deviner ce nombre.

- Après chaque proposition, le programme indique si le nombre mystère est plus grand ou plus petit que la proposition.
- Le jeu continue jusqu'à ce que le joueur trouve le nombre mystère ou qu'il n'ait plus de vies.
- Si le joueur trouve le nombre mystère, le programme affiche un message de félicitations.
- Si le joueur n'a plus de vies, le programme affiche un message de défaite et révèle le nombre mystère.
- Le programme doit utiliser une boucle `while` pour gérer les tentatives du joueur.

Le programme `devine_le_nombre_while.c` contient le code suivant :

```
| #include <stdio.h>
```

```
#include <stdlib.h>
#include <time.h>
int main(void) {
5   int vies;
   int nombre_mystere;
   int proposition;
   int niveau;
   // A COMPLETER : Demander le niveau de difficulté et initialiser le nombre de vies
10  // L'utilisation d'un switch case paraît appropriée ici.
   // Initialiser le générateur de nombres aléatoires
   srand(time(NULL));
   nombre_mystere = rand() % 100 + 1; // Nombre entre 1 et 100
   // A COMPLETER : Boucle principale du jeu
15  return 0;
}
```

- ☐ Compléter le programme `devine_le_nombre_while.c` pour qu'il fonctionne comme décrit.
- ☐ Tester le programme avec chaque niveau de difficulté.
- ☐ Que se passe-t-il si le joueur choisit le niveau "Impossible" ? Pourquoi ?
- ☐ Que se passe-t-il si le joueur choisit un niveau invalide (par exemple 5) ? Pourquoi ?

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/devinelenombre/while

Manipulation 6 : Devine le nombre - Version For, avec n vies

- ☐ Faire le même exercice que précédemment, mais en utilisant une boucle `for` pour gérer les tentatives du joueur.

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/devinelenombre/for

Manipulation 7 : Devine le nombre - Version Do...While, avec n vies

- ☐ Faire le même exercice que précédemment, mais en utilisant une boucle `do...while` pour gérer les tentatives du joueur.
- ☐ Tester le programme avec $n = 0$. Que constatez-vous ?
- ☐ Ce type de boucle vous paraît-il pertinent pour ce jeu ? Pourquoi ?

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/devinelenombre/doWhile

Manipulation 8 : Vérifier le choix d'un menu

- ☐ Quelle boucle est la plus adaptée pour vérifier que l'utilisateur a bien entré un choix valide dans un menu ? Justifier.
- ☐ Vérifier votre choix avec l'enseignant.
- ☐ Modifier vos programmes précédents pour utiliser la boucle choisie.

Manipulation 9 : Table de multiplication

- Écrire un programme `table_multiplication.c` qui demande à l'utilisateur un entier `n` et affiche la table de multiplication de `n` de 1 à 10 en utilisant une boucle `for`.

Par exemple, si l'utilisateur entre 5, le programme affichera :

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/maths/table_multiplication

Manipulation 10 : Somme des n premiers entiers

- Écrire un programme `somme_n_entiers.c` qui demande à l'utilisateur un entier `n` et calcule la somme des `n` premiers entiers (de 1 à `n`) en utilisant une boucle `for`. Le programme doit ensuite afficher le résultat.

Par exemple, si l'utilisateur entre 5, le programme affichera :

```
| 1 + 2 + 3 + 4 + 5 = 15
```

Utiliser deux boucles différentes pour le calcul de la somme et pour l'affichage du résultat.

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/maths/somme_n_entiers

Manipulation 11 : Factorielle

- Écrire un programme `factorielle.c` qui demande à l'utilisateur un entier `n` et calcule `n!` (la factorielle de `n`) en utilisant une boucle `for`.

Par exemple, pour `n = 5`, le programme affichera :

```
| 5! = 1 x 2 x 3 x 4 x 5 = 120
```

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/maths/factorielle

Manipulation 12 : Carre

- Écrire un programme `carre.c` qui demande à l'utilisateur un entier `n` et affiche un carré de `n` par `n` utilisant des astérisques (*). Utiliser une boucle `for` imbriquée pour générer les lignes et les colonnes du carré.

Par exemple, si l'utilisateur entre 4, le programme affichera :

```
****
****
****
****
```

Manipulation 13 : Triangle – Niveau 1

- Écrire un programme `triangle1.c` qui demande à l'utilisateur un entier `n` et affiche un triangle d'étoiles aligné à gauche, de hauteur `n`.

Par exemple, pour `n = 4` :

```
*
**
***
****
```

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/formes/triangle/niveau1

Manipulation 14 : Triangle – Niveau 2

- Écrire un programme `triangle2.c` qui demande à l'utilisateur un entier `n` et affiche un triangle isocèle aligné au centre, de hauteur `n`.

Par exemple, pour `n = 4` :

```
  *
 ***
*****
*****
```

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/formes/triangle/niveau2

Manipulation 15 : Damier

- Écrire un programme `damier.c` qui demande deux entiers `lignes` et `colonnes` et affiche un damier en alternant `#` et `..`.

Par exemple, pour `lignes = 4` et `colonnes = 6` :

```
#.#.#.
.#.#.#
#.#.#.
.#.#.#
```

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/formes/damier

Manipulation 16 : Motif en diagonale

- Écrire un programme `diagonale.c` qui demande un entier `n` et affiche un carré de taille `n` avec une diagonale de `*`.

Exemple, pour `n = 5` :

```
*....
.*...
..*..
...*.
....*
```

check50 IUT-GEII-Annecy/squelettes/branch-2025/tp3/formes/diagonale