

# TP : tirage aléatoire

## Objectif(s)

- ★ Renforcer vos connaissances et compétences dans l'utilisation des boucles, des expressions logiques et des conditionnelles
- ★ Utiliser la fonction `random` permettant de tirer des nombres au hasard

## Exercice 1 – Statistiques

L'objectif ici est de faire des statistiques sur les tirages aléatoires. Ce TP reprend les exercices de TD.

### Question 1

Écrivez une fonction `lance_un_de` qui simule un lancer de dé à 6 faces, et retourne la valeur tirée. On rappelle qu'en C, la valeur retournée est comprise entre 0 et `RAND_MAX`, une constante de valeur élevée.

Ecrivez un programme `main` qui appelle la fonction `lance_un_de` et qui affiche le résultat. Exécutez plusieurs fois le programme.

Ajoutez la ligne de code suivante au début de votre programme : `srand( time( NULL ) )`

Exécutez plusieurs fois le programme. Que fait la fonction `srand` ?

### Question 2

Créez une fonction qui exécute 1000 lancers de dés et affiche le nombre de tirages de chaque valeur obtenu.

Ecrivez un programme `main` qui appelle votre fonction.

Exécutez plusieurs fois le programme.

### Question 3

Modifiez la fonction de la question précédente ainsi que la fonction `main` pour qu'elle demande le nombre de lancers à l'utilisateur.

### Question 4

Écrivez une fonction qui demande un nombre de lancers à l'utilisateur, et affiche le plus grand lancer obtenu.

### Question 5

Même question avec la somme des lancers.

### Question 6

Créez une fonction qui demande un nombre de lancers à l'utilisateur, et affiche la longueur de la plus grande séquence de tirages de même valeur obtenus consécutivement.

## Renforcement

## Exercice 2 – Jeu de devinette

### Question 1

Écrivez un programme dans un fichier `devinette.c` qui tire un nombre au hasard entre 1 et 1000, et l'affiche.

### Question 2

Modifiez votre programme pour qu'il demande à l'utilisateur de deviner le nombre, en lui donnant des indications du type "la cible est plus grande" ou "la cible est plus petite"

*Question 3*

Modifiez votre programme pour qu'il compte le nombre d'essais de l'utilisateur.

*Question 4*

On peut envisager que le programme ne soit pas totalement honnête, et qu'il choisisse la valeur qu'il fait deviner en fonction des questions du joueur. Par exemple, si le joueur fait une demande avec pour entrée le nombre 200 au départ, on constate que la solution sera plus facile à trouver si on répond "plus petit" que "plus grand". Le programme pourrait donc répondre plus grand. Au coup suivant, si l'utilisateur demande 700, on constate qu'il y a plus de valeurs entre 200 et 700 qu'entre 700 et 1000. Le programme choisirait donc de répondre "plus petit".

Implémentez donc cette version malhonnête du jeu, tout en vous assurant qu'il existe toujours une solution.