

# Programmation impérative en langage C

## feuille de TP n°1 : Révisions langage C

---

*Objectifs du TP :*

- *Éditer compiler et exécuter un programme C en ligne de commande*
- *Écrire des fonctions impératives et récursives*
- *Révision : types structurés , génération de nombres aléatoires*

## 1 Premières fonctions

◇ Exercice 1 : Écrire une fonction qui calcule  $X^N$  où  $X$  est un réel et  $N \geq 0$  un entier naturel passés en paramètre.

1. On écrira 2 versions de cette fonction, l'une itérative et l'autre récursive.
2. Écrire une boucle qui teste l'égalité des résultats fournis par les 2 fonctions pour tous les  $X^i, 0 \leq i \leq N$  avec  $X$  et  $N$  saisis par l'utilisateur.

◇ Exercice 2 : Étant donné un vecteur de  $N(\leq 10)$  entiers entrés au clavier, on veut écrire un programme qui vérifie si ce vecteur est un palindrome. Un vecteur est palindrome lorsque ses cases symétriques sont identiques.

Les données sont *int N*, *int T[10]* vérifiant  $0 \leq N \leq 10$

*Donnez la dimension du tableau: 9*  
*Donnez le tableau: 5 3 8 26 7 26 8 3 5*  
*C'est un palindrome.*

## 2 Révision : types structurés, génération de nombres aléatoires

◇ Exercice 3 : Dans cet exercice, on utilise la constante symbolique :

```
#define N_MAX 1000
```

et le type structuré :

```
struct Tab {
    int tab[N_MAX]; //tableau statique
    int n; //nombre de valeurs
}
```

1. Écrire une fonction qui génère un tableau de type `struct Tab` dont le nombre de valeurs est passé en paramètre. Ce nombre de valeurs est supposé être inférieur à `N_MAX`. Les valeurs du tableau seront générées aléatoirement.

Le code qui suit rappelle comment générer des entiers aléatoires :

```

1 #include <stdlib.h> //pour utiliser le generateur pseudo-aléatoire
2 #include <time.h>   //pour l'initialiser grâce à l'horloge
3
4 int main (void)
5 {
6     //placée en début de main pour être exécutée une seule fois
7     //cette instruction permet d'éviter de générer
8     //les mêmes valeurs à chaque exécution
9     srand(time(NULL))
10    int x=rand() //generation d'un entier dans [0,RAND_MAX]
11    int y=5 + rand()%7 //generation d'un nombre dans [5,12]
12    return 0;
13 }

```

2. On veut supprimer tous les multiples de 7 du tableau précédemment généré. A la fin, le tableau initial doit être débarrassé de tous les multiples de 7. On ne conservera pas nécessairement l'ordre initial. Cette tâche peut être effectuée en parcourant une seule fois le tableau de départ.

Écrire une fonction qui effectue cette tâche :

*Avant suppression, 5 valeurs dans le tableau : 0 6 7 15 3*  
*Après suppression, il ne reste plus que 3 valeurs 6 15 3*

### 3 Pour terminer la séance

◇ Exercice 4 : On considère la suite infinie : 1 2 2 3 3 3 4 4 4 4 5 5 5 5 ...

Écrire une fonction qui pour un entier  $N \geq 1$  passé en paramètre renvoie la valeur du  $N$ ième terme de la suite.

