

**TP4**  
**Programmation C L2.1**  
**Utilisation de tableaux**

---

Dans cette séance de travaux pratiques, nous allons travailler sur des tableaux de  $N$  entiers.

---

**1. Valeurs aléatoires**

La fonction `random` (de `<stdlib>`) renvoie un entier pseudo-aléatoire positif compris entre 0 et `RAND_MAX`. Afin d'obtenir des valeurs différentes à chaque exécution, on initialise (une fois) la suite de valeurs grâce à la fonction `srandom(int seed)`; . Une manière d'obtenir une graine (`seed`) différente à chaque exécution est d'utiliser la fonction `time()` de `<time.h>` qui renvoie le nombre de secondes écoulées depuis le 1 janvier 1970. Ceci s'exprime par l'instruction (dans le `main`) :`srandom(time(NULL))`;

- (a) Écrire un programme qui affiche une suite de 20 valeurs pseudo-aléatoires, sans utiliser l'instruction `srandom(time(NULL))`; . Lancer plusieurs fois ce programme puis ajouter cette instruction.
- (b) Écrire une fonction `void initAlea(int t[], int taille, int max)` qui remplit les `taille` premières cases du tableau `t` avec des entiers positifs aléatoires inférieurs à `max`.

**2. Recherche d'un entier dans un tableau**

Écrire une fonction `int position(int t[], int taille, int x)`; prenant en arguments un tableau `t`, sa taille et un entier `x`. et qui renvoie l'indice de la première occurrence de `x` dans `t` si `x` apparaît dans les `taille` premiers éléments de `t` et `-1` sinon.

contient la somme des tableaux `A` et `B`.

**3. Histogramme**

Écrire une fonction `Histogramme` qui reçoit deux tableaux d'entiers `Note` et `Histo`. Le tableau `Note` a pour taille `N` et contient des notes de 0 à 20. Le tableau `Histo` de taille 21 devra représenter, après l'appel de la fonction, l'histogramme des notes du premier tableau, c'est à dire que `Histo[i]` est le nombre de valeurs égales à `i` dans le tableau `Note`.

4. Écrire une fonction `main()` utilisant les fonctions précédentes (remplissage d'un tableau aléatoire et affichage de l'histogramme).

**5. Somme de deux tableaux**

On définit la *somme* de deux tableaux `A` et `B` de taille  $N$ , comme le tableau `C` de taille  $N$  tel que :  $\forall i \in [0..N - 1] \ C[i] = A[i] + B[i]$  Écrire une fonction `sommeTab` recevant en arguments trois tableaux `A`, `B` et `Resultat` ainsi que leur taille. A la sortie de `somme`, `Resultat` contient la somme des tableaux `A` et `B`