

TP3

L'objectif est de comparer les performances de 2 algorithmes de tri.

1. Définir une constante `N`, qui vaudra 10 pour ce début du TP, et un type `Tab` tableau de `N` entiers.
2. La fonction suivante, qui utilise `stdlib.h`, permet d'initialiser un tableau de `n` entiers à l'aide de valeurs aléatoires de $[0..n - 1]$:

```
void remplirTableau(Tab t, int n)
{
    int i;
    srand ( time(NULL) ); // initialise le random
    for (i=0; i<n; i++)
        t[i]= rand() % n; // t[i] recoit une valeur aleatoire modulo n
}
```

Tester dans le `main` le remplissage et l'affichage d'un tableau `t1` de 10 valeurs aléatoires. Rem. : il peut y avoir plusieurs fois la même valeur dans le tableau.

3. Ecrire une fonction qui copie un tableau de `n` valeurs dans un autre. Modifier le `main` pour disposer de 2 tableaux identiques, `t1` et `t2`.
4. Implémenter la fonction de tri par sélection du min vue en TD (TD 4 ex. 2) et en cours. Ajouter au `main` le tri sur `t1`, puis l'affichage du tableau trié pour vérification.
5. Implémenter la fonction de tri rapide vue en TD (TD 5 ex. 1) et en cours, en prenant comme pivot la valeur du milieu du tableau. Ajouter au `main` le tri sur `t2`, puis l'affichage du tableau `t2` trié pour vérification.
6. Modifier le `main` pour pouvoir calculer et afficher les durées d'exécution des tris des tableaux `t1` et `t2` avec $N = 1000$. Aide : le code suivant permet d'afficher la durée d'exécution (en millisecondes) de la fonction `f` :

```
#include <stdio.h>
#include <time.h>
void f()
{
    // des traitements non detaillés ici
}
void main()
{
    double temps_debut, temps_fin;

    temps_debut=(double)clock();  f();    temps_fin = (double)clock();/
    printf("\nDuree de traitement : %f ms\n", 1000*(temps_fin - temps_debut)/CLOCKS_PER_SEC);
}
```

7. A l'aide d'une boucle `pour` dans le `main`, calculer puis afficher la durée moyenne de chaque tri sur 10 tours de boucle.