

**TP05 (Evaluation1)**

**Matière : ATELIER SYSTEME D'EXPLOITATION EMBARQUE**

**Classe : SEM21**

Ce TP est une évaluation sur l'utilisation de gcc, make et git. Ce Tp utilise le lien suivant :

- <http://langage-info.blogspot.com/2014/05/trois-algorithmes-du-tri-en-c.html>
- <https://pub.phyks.me/sdz/sdz/ime-h-et-ses-fonctions.html>
- <https://waytolearnx.com/2019/08/tri-rapide-en-c.html>
- <https://github.com/ahmedOumezzine/Algorithme-tri-language-C>

L'objectif du TP est d'écrire un programme C qui permet de comparer les temps d'exécution de quatre algorithmes de tri (Tri par sélection, tri à bulle, tri par permutation et tri rapide). La structure du projet est un dossier nommé comp\_tri qui contient : man.c, tri.h, tri\_selection.c, tri\_bulle.c, tri\_permutation.c, tri\_rapide.c.

## Travail demandé

- 1- ajouter un dossier TP05 et entrer dans TP05.
- 2- Dans TP05 ajouter un dossier comp\_tri et entrer dans TP05.
- 3- Installer emacs.
- 4- Installer build-essential.
- 5- Télécharger avec git les trois fichiers "05\_ASEE\_TP05\_main.cpp", "05\_ASEE\_TP05\_tri.h" et "05\_ASEE\_TP05\_util.c".
- 6- Renommer les trois fichiers pour les rendre "main.c", "tri.h" et "util.c".
- 7- Ajouter un fichier makefile pour compiler et exécuter les trois fichiers.
- 8- Tester les fonctions lire(...), afficher(...) et copier(...).
- 9- Ajouter un fichier tri\_selection.c et écrire son code
- 10- Dans "main.c" ajouter le code qui permet d'appeler tri\_selection et d'afficher le résultat
- 11- Ajouter les modifications nécessaires à makefile pour compiler et exécuter
- 12- Dans "main.c", en se basant sur le deuxième lien, ajouter le code nécessaire pour afficher le temps du tri\_selection
- 13- Recompiler et exécuter
- 14- Refaire le même travail avec les autres algorithmes de tri
- 15- Dans "main.c" ajouter le code qui permet de comparer les temps d'exécution et d'afficher le nom de l'algorithme le plus rapide.
- 16- Recompiler et exécuter.

## - Tri par sélection

```
int i,j,c;
for(i=0;i<N-1;i++)
    for(j=i+1;j<N;j++)
        if ( T[i] > T[j] ) {
            c = T[i];
            T[i] = T[j];
            T[j] = c;
        }
```

## - Tri à bulle

```
int i,j,c;

for(j=1;j<=N;j++) // pour faire l'operation N fois
    for(i=0;i<N-1;i++)
        if ( T[i] > T[i+1] ) {
            c = T[i];
            T[i] = T[i+1];
            T[i+1] = c;
        }
```

## - Tri par permutation

```
int i,j,k,c;

for(i=1;i<N;i++) {

    if ( T[i] < T[i-1] ) {
        j = 0;

        while ( T[j] < T[i] )
            j++;
        c = T[i];
        for( k = i-1 ; k >= j ; k-- )
            T[k+1] = T[k];
        T[j] = c ;
    }
}
```

## - Tri rapide

```
void permuter(int *a, int *b) {
    int tmp;
    tmp = *a;
    *a = *b;
    *b = tmp;
}

void triRapid(int tab[], int first, int last) {
    int pivot, i, j;
    if(first < last) {
        pivot = first;
        i = first;
        j = last;
        while (i < j) {
            while(tab[i] <= tab[pivot] && i < last)
                i++;
            while(tab[j] > tab[pivot])
                j--;
            if(i < j) {
                permuter(&tab[i], &tab[j]);
            }
        }
        permuter(&tab[pivot], &tab[j]);
        triRapid(tab, first, j - 1);
        triRapid(tab, j + 1, last);
    }
}
```

