Série de TD n°3

Exercice 1

```
#include <stdio.h>
int main()
{
   int i, k;
   int val;
   int T[10];
   for (i=0; i<=9; i++)
     {
       printf(" \n T[%d] = ",i);
       scanf(" %d ",&T[i]);
     }
   for (i=0; i<=9; i++)
     for (k=0; k<9-i; k++)
       if (T[k] > T[k+1])
         {
           val = T[k];
           T[k] = T[k+1];
           T[k+1] = val;
         }
   return 0;
}
```

1. Exécuter le programme sur un tableau rempli avec les valeurs :
   22 14 56 5 35 69 8 15 12 1
2. À quoi sert ce programme ?
3. Relancer le programme mais cette fois-ci avec les valeurs suivantes :
   1 5 8 12 14 15 22 35 56 69
   Que constatez-vous ?
4. Essayer de déterminer la complexité de cet algorithme, c'est-à-dire le nombre d'opérations nécessaire dans le pire des cas.
5. Relancer le programme avec les valeurs suivantes :
   1 5 8 12 14 15 22 69 56 35
   Déterminer le nombre de permutations réalisées. Proposer une amélioration de ce programme.

Exercice 2

Écrire un programme C permettant de trouver la première et la dernière occurrence d'un entier dans un tableau d'entiers.

Exercice 3

Écrire les fontions permettant de

1. déterminer la transposée d'une matrice,
2. calculer la trace d'une matrice (c'est-à-dire la somme de ses élément diagonaux),
3. rechercher le plus grand coefficient d'une matrice.

Exercice 4

1. Écrire une fonction permettant d'initialiser les valeurs contenues dans un tableau d'entiers en générant des nombres aléatoirement entre -100 et 100.
2. Écrire une fonction permettant de trier un tableau d'entiers suivant le critère : tous les entiers pairs se trouvent au début du tableau trié, les entiers impairs se trouvent à la fin du tableau trié. (Remarque : les nombres pairs (resp. impairs) ne doivent pas nécessairement être triés entre eux.)