

TD 4 – Programmation C
Tableaux – Structures de données

Exercice 1 :

Écrire un programme en C qui crée et initialise un tableau, puis supprimez les éléments en double dans ce tableau.

Pour supprimer un élément du tableau, déplacez les éléments justes après la position donnée vers une position à gauche.

Exercice 2 : Recherche dichotomique dans une table

On a une table de n nombres distincts, rangés par ordre croissant : $a[1] < a[2] < \dots < a[n]$. On veut savoir si le nombre x est dans la table. Nous connaissons grosso modo la méthode de recherche : on compare x à l'élément du milieu de la table ; s'il lui est égal, c'est fini, sinon on le recherche dans la moitié basse ou haute suivant les cas.

Exercice 3 :

1. Créer une **structure de données Client** dont les informations sont structurées de la manière suivante :

Numéro de matricule (entier)

Nom (chaîne de caractères)

Prénom (chaîne de caractères)

2. Créer un tableau nommé « **liste** » de type **structure de données Client**.
3. Créer une fonction **Ajout** qui ajoute un nouveau client (entré au clavier) à la liste.

On y prévoira aussi les trois fonctions suivantes :

- a. void AfficheMat(...) : affiche tous les clients de la liste dont le numéro de matricule se termine par 8.
 - b. void RechercheNom(...) : Recherche puis affiche tous les clients dont le nom est à entrer au clavier par l'utilisateur.
 - c. void tri(...) : Trie la liste des clients par nom selon un algorithme de votre choix.
 - d. void Affiche(...) : affiche tous les clients de la liste dont le nom est un palindrome.
4. Écrire un programme main () de test.

TD4 Prog C

Exercise 1:

```
#include <stdio.h>
int main () {
    int i, j, occ, k=0, a, n=100;
    int T[n];
    for (i=0; i < n-k; i++) {
        for (j=i+1; j < n-k; j++) {
            occ = 0;
            if (T[i] == T[j]) {
                k++;
                for (a=j; a < n; a++)
                    T[a] = T[a+1];
                j--j;
            }
        }
    }
    return 0;
}
```

Exercice 2: