



ALGORITHMIQUE ET STRUCTURES DE DONNÉES EXAMEN FINAL

Exercice 1 :

On dispose d'une chaîne de caractères « chaine1 » et on souhaite y remplacer chaque occurrence d'un caractère c1 par un caractère c2. Le résultat sera donné dans une deuxième chaîne « chaine2 ».

1. Écrire une fonction itérative.
2. Écrire une fonction récursive, ainsi que son premier appel.

Exemple :

Entrée : chaine1 = "/ma / chaine /"

avec c1='/' et c2='\'

Sortie :

chaine2 = "\\ma \\ chaine \\"

Exercice 2 :

Un arbre binaire est une structure hiérarchique où chaque sommet est représenté par :

```
struct node
{
    int data;
    struct node *left;
    struct node *right;
};
```

1. A partir d'un arbre binaire de racine B, implémenter une fonction pour supprimer une feuille d'adresse F .
2. Construire un arbre AVL à partir des clés :

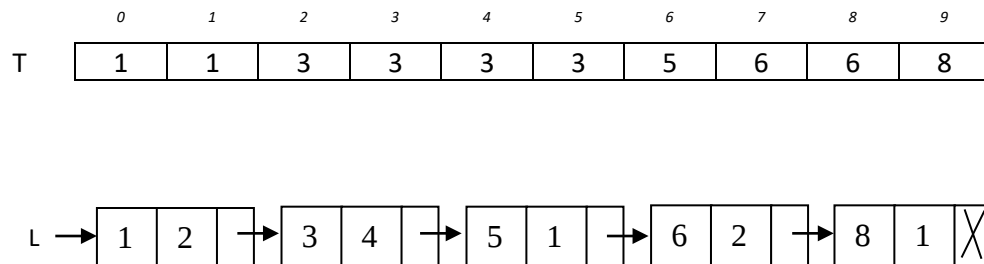
35	20	55	25	10	5	8
----	----	----	----	----	---	---

Examiner la propriété de l'équilibre après chaque ajout.

Exercice 3 :

1. Étant donné un tableau **T** contenant **N** entiers triés en ordre croissant. Écrire une fonction pour retourner le nombre d'occurrence d'une valeur passée en paramètre.
2. On souhaite créer une liste chaînée **L** pour stoker les éléments de **T** ; où chaque cellule contient l'élément, son nombre d'occurrence et un pointeur vers le suivant.

Exemple :



- A. Proposer une structure de donnée pour une cellule de **L**.
- B. Écrire une fonction prenant en entrée **T** et **N** (le tableau et le nombre de ses éléments), et retourne la liste chaînée (précédemment définie).
- C. Écrire une fonction permettant de vérifier si une liste chaînée **L** correspond à un tableau **T** (la liste chaînée **L** a été créée pour **T** selon la stratégie citée ci-avant).

Barème :

Exercice 1 : (4 pts = 1 + 3).

Exercice 2 : (6 pts = 4 + 2).

Exercice 3 : (10 pts = 1 + 1 + 4 + 4).