



PARTIEL

Exercice 1 :

Que fait la fonction *Mystere* ?

```
void Mystere(char *str) {  
    int len = strlen(str);  
    if (len < 2) return;  
    int i, j = 0;  
    for (i = 1; i < len; i++)  
        if (str[i] != str[i - 1])  
            str[++j] = str[i];  
    str[++j] = '\0';  
}
```

Exercice 2 :

On dispose d'un tableau T (liste contigüe) de taille N contenant des éléments entiers.

1. Écrire une fonction qui permute les deux éléments T[i] et T[j], avec $i, j \in [0 .. N-1]$.
2. Écrire une fonction qui retourne l'indice (position) du plus grand élément du tableau.

On souhaite trier T par ordre croissant en se basant sur le principe suivant : rechercher le plus grand élément, le placer en fin de tableau, recommencer avec le second plus grand, le placer en avant-dernière position et ainsi de suite jusqu'à avoir parcouru la totalité du tableau.

3. Écrire une fonction itérative.
4. Écrire une fonction récursive.

Durée : 1 h 15 mn.

Barème : Exercice 1 (3 points), Exercice 2 (10 points = 1+2+3+4).

Corrigé

Exercice 1:

Mystere : Fonction permettant de supprimer des occurrences consécutives de caractères dans une chaîne de caractères.

Exercice 2 :

1. Permuter deux éléments d'un tableau

```
void permuter(int T[], int i, int j)
{
    int tmp;
    tmp = T[i];
    T[i] = T[j];
    T[j] = tmp;
}
```

2. Renvoie l'indice du plus grand élément du tableau

```
int max-ind(int tab[], int taille)
{
    int i=0, indice_max=0;
    while(i < taille)
    {
        if(tab[i] > tab[indice_max])    indice_max = i;
        i++;
    }
    return indice_max;
}
```

3. Version itérative

```
void tri_selection(int tab[], int taille)
{
    int indice_max;

    for(; taille > 1 ; taille--) // tant qu'il reste des éléments non triés
    {
        indice_max = max-ind(tab, taille);

        permuter(tab, taille-1, indice_max);
        // on échange le dernier élément avec le plus grand
    }
}
```

4. Version Récursive

```
void tri_selection_recuratif(int tab[], int taille)
{
    // un tableau d'un seul élément ou moins n'a pas besoin d'être trié
    if(taille <= 1)        return;

    permuter(tab, taille-1, max-ind(tab, taille));
    // on échange le dernier élément avec le plus grand
    return tri_selection_recuratif(tab, taille-1);
    // on rappelle la fonction en diminuant la taille du tableau
}
```