

Feuille de TD n° 8

Exercice 1 : Le drapeau à trois couleurs

- a) Ecrire la fonction `void afficher(char t[], int n)` qui affiche sur une ligne le contenu du tableau t de taille n .
- b) Ecrire la fonction `void permuter(char t[], int i, int j)` qui permute les éléments d'indices i et j du tableau t .
- c) Ecrire la fonction `void remplir_VJR (char t[], int n)` qui remplit le tableau t , de taille n , par des valeurs aléatoires prises dans l'ensemble $\{ 'V', 'J', 'R' \}$.
- d) On considère maintenant la fonction suivante :

```
void trier_en_VJR (char t[ ], int n)
{   int i, j, k ;
    i=0 ;
    j=0 ;
    k=n-1 ;
    while(j<=k)
    {   switch(t[j])
        {
            case 'V' : { permuter(t,i,j) ;   i++ ;   j++ ; break ; }
            case 'J' : { j++ ; break ;}
            case 'R' : { permuter(t,j,k) ;   k-- ;}
        }
    }
    return ;
}
```

Si tab contient J R J J V R V V, quel est le résultat de l'appel `trier_en_VJR(tab, 8)` ?

- e) Ecrire la fonction `main` qui orchestre le tout : remplir un tableau u avec des valeurs aléatoires de $\{ 'V', 'J', 'R' \}$, afficher le contenu de u , "trier_en_VJR" le tableau u , afficher le résultat.

Exercice 2 : Le tri par casiers

Le tri dit "par casiers" peut s'utiliser lorsque le nombre de valeurs différentes à trier est limité. On l'utilise ici sur un tableau de caractères alphabétiques.

On se donne un tableau L dont chacune des N cases contient un caractère alphabétique minuscule. Le nombre de valeurs différentes à trier est donc ici égal à 26.

Le principe du tri par casiers est le suivant : après avoir déclaré et initialisé à 0 les cases d'un tableau C de 26 éléments entiers, on lit les éléments du tableau L . Pour chaque valeur contenue dans L , on incrémente la case correspondante dans le tableau C : on incrémente la case $C[0]$ pour la valeur 'a' , la case $C[1]$ pour la valeur 'b' , ... etc ... et la case $C[25]$ pour la valeur 'z'.

Il suffit ensuite, pour i allant de 0 à 25, de remplir les cases successives du tableau L en y affectant $C[i]$ fois le $(i+1)$ -ème caractère de l'alphabet.

Exemple : si les caractères contenus dans L forment le mot `introduction`, le tableau C contiendra
 0 0 1 1 0 0 0 0 2 0 0 0 0 2 2 0 0 1 0 2 1 0 0 0 0 0

et le tableau L , à la fin du tri, contiendra les caractères formant la chaîne `cdiinnoorttu`.

Ecrire la fonction `void tri_casiers(char t[], int n)` qui effectue ce tri sur le tableau t de taille n .