

## TP4 Tableaux d'entiers et de caractères

### 1 Tableaux d'entiers

Écrire un programme constitué d'une fonction principale appelant les sous-programmes suivants (tous les tableaux considérés auront une taille  $N$  avec  $N$  constante égale à 10).

1. sous-programme réalisant le remplissage d'un vecteur d'entiers passé en paramètre à l'aide d'une unique valeur  $v$  aussi passée en paramètre.
2. sous-programme réalisant le remplissage d'un vecteur d'entiers passé en paramètre à l'aide de nombres tirés au hasard :

```
#include <stdlib.h>
#include <time.h>
...

/*initialisation du générateur*/
void initialise_rand()
{
    srand(time(NULL));
}

/*tire un entier au hasard entre inf et sup */
int hasard(int inf, int sup)
{
    int rando = rand();
    return (inf + (rando % (sup - inf + 1)));
}
```

3. sous-programme affichant les éléments d'un vecteur d'entiers passé en paramètre ;
4. sous-programme effectuant la somme des éléments d'un vecteur d'entiers passé en paramètre ;
5. sous-programme recherchant le maximum du vecteur d'entiers passé en paramètre.

Vous indiquerez en commentaires le type (Donnée, Résultat ou Donnée/Résultat) des paramètres de chaque sous-programme. Vous testerez bien vos procédures et fonctions, elles seront utilisées dans le TP sur les tris.

### 2 Tableaux de caractères (mots)

**Table ASCII** La table ASCII est une table qui fait la correspondance entre un caractère et son entier qui le code en machine. On peut la trouver partout sur le web.

REMARQUE 1 *On remarquera que  $code(Z) = code(z) - 32$ .*

Exemples d'utilisation :

- Pour obtenir un caractère à partir de son code ASCII, on déclare la variable comme un caractère et on lui donne la valeur voulue. Par exemple, on peut faire `char mon_a = 97;`
- Pour imprimer le caractère désigné par le code ASCII 102, on fait `printf("%c", 102)`.
- Pour imprimer le code ASCII du caractère B, on fait `printf("%d", 'B')`.
- Pour remplir un tableau de caractères désignés par leur code, on déclarera le tableau `char tab[MAX]` par exemple, que l'on pourra remplir à l'aide d'instructions de la forme :  
`tab[i] = 67;`

**Exercice sur les tableaux de caractères** (D'après un TD de Michaël Périn, Université de Grenoble 1, 2004)

1. Écrire un programme (un autre) qui déclare un tableau `tab` de 500 caractères (entre 'a' et 'z', minuscules), et qui l'initialise avec des caractères tirés au hasard. On utilisera la fonction `hasard` de la section précédente.

2. Modifier le programme pour qu'il affiche chaque caractère du tableau suivi de son code ASCII ainsi que la majuscule correspondante.

```
c[0] = 'b' code = 98 maj = 'B'
c[1] = ...
```

3. Afficher la plus petite lettre (pour l'ordre alphabétique) ayant été tirée :

```
La plus petite lettre est : 'b'.
```

4. Demander une lettre à l'utilisateur de taper une lettre et afficher les indices des cases où elle se trouve. On utilisera `getchar()` :

```
On trouve la lettre 'c' dans les cases 35, 196, 354, 488.
```

5. Déterminer et afficher le nombre d'apparitions de la lettre donnée par l'utilisateur. On utilisera `getchar()` :

```
On trouve la lettre 'c' 4 fois.
```

6. Déterminer le premier indice d'apparition de chacune des lettres et afficher sous la forme :

```
'a' n'apparaît jamais
'b' apparaît pour la première fois en position 0
'c' apparaît pour la première fois en position 35
...
```

On n'utilisera pas de tableau, et pour chaque lettre on utilisera une boucle `while`.

7. Déclarer et initialiser un tableau `occurrence` qui va nous servir à compter le nombre d'apparitions de chaque lettre. De quelle taille est-il ?

8. Remplir le tableau `occurrences` avec le nombre d'apparition de chaque lettre dans le tableau `tab`, et afficher le résultat ensuite sous la forme :

```
'a' apparaît 0 fois
'b' apparaît 1 fois
'c' apparaît 4 fois
...
```

9. Afficher le contenu du tableau `tab` dans l'ordre alphabétique. On utilisera le tableau `occurrences`, et si le caractère 'a' n'apparaît pas, on n'affichera pas 'a'.