

TP4 Tableaux d'entiers et de caractères

1 Linux

Faites ces manipulations dans un terminal !

1. Préparez un répertoire TP4 dans votre répertoire pour les TP d'algo.
2. Un fichier `alea.c` contenant des fonctions utiles au TP se trouve à l'URL :
`http://www2.lifl.fr/~grisoni/IMA/`
Copiez-le dans votre répertoire TP4 (commande `wget`).
3. Faites une copie de `alea.c` nommée `tp4_a.c` (commande `cp`).

Pour chaque fonction/procédure écrite, vous commenterez en indiquant la fonctionnalité codée, et le type de passage des paramètres (Donnée, Résultat ou Donnée/Résultat).

2 Tableaux d'entiers

Dans le fichier `tp4_a.c`, écrire à la suite des fonctions fournies :

1. Une fonction principale `main` déclarant un tableau de taille 10, puis l'initialisant de la façon suivante (cf le cours) :

```
int t[10]={1,4,10,7,10,-1,-1515,2012,190,200};
```

Compiler avec clang pour éliminer les éventuelles erreurs de syntaxe. Observer le message d'alerte de clang si vous mettez 11 éléments dans le tableau. Observer le message d'erreur de clang si vous déclarez le tableau comme ceci :

```
int size = 10;
int t[size]={1,4,10,7,10,-1,-1515,2012,190,200};
```

puis remettez tout comme initialement (supprimez la déclaration de la variable `size` et `int t[10]={1,4,...}`). Ce qu'il faut retenir c'est qu'un tableau en C (pour l'instant) doit connaître sa taille à la compilation.

2. Un sous-programme affichant les éléments d'un vecteur d'entiers passé en paramètre :

```
void affiche_tableau(int t[], int size){
    ...
}
```

Une fois cette action écrite (elle doit précéder le `main` dans le fichier), on teste dans le `main` sur le tableau déjà déclaré.

Dans la suite les tableaux passés en paramètres le seront sur le modèle précédent (tableau de taille variable, et un entier passant sa taille).

3. 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. Appel dans le `main`, et affichage du tableau modifié.

4. sous-programme réalisant le remplissage d'un vecteur d'entiers passé en paramètre à l'aide de nombres tirés au hasard entre 0 et 400 (un tirage au sort pour chaque case du tableau). La fonction à appeler est `int hasard(int inf, int sup)` qui vous a été fournie. Au début du `main`, on fera attention de bien initialiser le générateur de nombres aléatoires, en appelant la fonction : `initialise_rand()`
5. sous-programme effectuant la somme des éléments d'un vecteur d'entiers passé en paramètre. Tester.
6. sous-programme recherchant le maximum du vecteur d'entiers passé en paramètre. Tester.

3 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 $\text{code}(Z) = \text{code}(z) - 32$ (ou encore ' $z' - 'Z' = 32$).*

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[42]` 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) Pour cet exercice, on testera chaque fonction sur un tableau de taille petite (10 par exemple), avant de lancer sur le tableau de taille 500. **On écrira des fonctions et des actions sur le modèle précédent.**

1. Écrire un programme (un autre, dans un fichier `tp4_b.c`) qui déclare un tableau `tab` de 500 caractères (entre 'a' et 'z', minuscules), et qui l'initialise avec des caractères tirés au hasard. On copie-collera judicieusement les fonctions écrites dans `tp4_a.c`.
2. Modifier le programme pour qu'il affiche chaque caractère du tableau suivi de son code ASCII ainsi que la majuscule correspondante (faire une procédure, bien sûr, et procéder par étapes!) :

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

3. Afficher la plus grande lettre (pour l'ordre alphabétique) ayant été tirée :
La plus grande lettre est : 't'.
4. Demander une lettre à l'utilisateur de taper une lettre et afficher les indices des cases où elle se trouve. On utilisera la fonction de librairie `char getchar(void)` :
On trouve la lettre 'c' dans les cases 35, 196, 354, 488.
5. D'autres questions sur la feuille d'exercices supplémentaires située sur la page du cours.