

Introduction à la programmation

TD 2 - Tableaux

Polytech Marseille - 1ère année

Filière Informatique
Alexandra Bac - Benoît Favre

Exercice 1. On appelle palindrome une suite d'éléments qui se lit de la même manière dans les deux sens (par exemple "AZTTZA" ou "43634").

On déclare une constante N qui sera initialisée comme vous les souhaitez ainsi qu'un tableau T de N caractères qu'on initialisera de manière statique.

- (i) Ecrire un programme testant si le tableau T est un palindrome.
- (ii) Que fait le programme suivant, que calcule-t-il dans le tableau TT :

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 5

int main()
{
    const int p = 0 ; // p doit etre initialise avec 0 ou 1
    const int m = 2*N-p ;
    int i, j ;
    int T[N], TT[m] ;
    srand(time(NULL)) ;

    for(i=0; i<N; i=i+1)
    {
        T[i] = rand() % 20 ;
    }

    i = 0 ; j = m-1 ;
    while(i<N)
    {
        TT[i] = T[i] ;
        TT[j] = T[i] ;
        i = i+1 ;
        j = j-1 ;
    }
    return 0 ;
}
```

Exercice 2. Soit T un tableau de N entiers. Ecrivez un programme qui inverse le contenu de T , c'est-à-dire, par exemple :

$$\text{avant : } T = \boxed{t_0 \mid t_1 \mid t_2 \mid \cdots \mid t_{N-1}}$$

après :

| | | | | |
|-----------|-----------|-----------|---------|-------|
| t_{N-1} | t_{N-2} | t_{N-3} | \dots | t_0 |
|-----------|-----------|-----------|---------|-------|

- (i) Soit en stockant le résultat (les valeurs inversées) dans un autre tableau
- (ii) Soit directement en modifiant le tableau T (sans avoir recours à aucun tableau auxiliaire)

Exercice 3. Ecrivez un programme qui lit au clavier une suite x_0, x_1, x_2, \dots de nombres entiers tels que $0 \leq x_i \leq 20$ et qui calcule et affiche le nombre d'apparitions de chaque valeur dans la suite. La frappe d'un nombre hors de cet intervalle indique la fin de la suite.

Exercice 4.

- (i) Ecrivez un programme qui cherche et imprime la *valeur* du plus petit élément d'un tableau T de N nombres entiers.
- (ii) Ecrivez un programme qui cherche et imprime le *rang* du plus petit élément d'un tableau T de N nombres entiers.

Exercice 5 (RECHERCHE SEQUENTIELLE). Ecrivez un programme qui, étant donné un tableau T de N nombres entiers et un nombre entier X , cherche et imprime le plus petit rang i tel que $T_i = X$, dans les deux cas suivants :

- on est sûr qu'il existe au moins un élément égal à X dans le tableau,
- on n'est pas assuré qu'un tel élément existe.

Exercice 6 (TRI PAR SELECTION/ECHANGE). On souhaite trier les éléments d'un tableau T en utilisant les principes suivants :

- Invariant : à l'étape i , les i premières cases du tableau sont triées et contiennent les i plus petits éléments du tableau
 - A l'étape $i + 1$:
 - Trouver le minimum des cases non triées
 - Le mettre à sa place dans la $i + 1$ ème case
- Pourquoi cet algorithme trie bien le tableau. L'appliquer à

$$T = \begin{array}{|c|c|c|c|c|c|} \hline 7 & 2 & 9 & 8 & 5 & 1 \\ \hline \end{array}$$

puis écrire l'algorithme de tri correspondant. Quelle est sa complexité ?

Exercice 7 (TRI PAR INSERTION). On utilise ensuite un autre principe pour trier les éléments d'un tableau T :

- Invariant : à l'étape i , les i premières cases du tableau contiennent les i premières valeurs de ce tableau triées.
 - A l'étape $i + 1$:
 - On insère le $i + 1$ ème élément à sa place en le comparant progressivement aux éléments précédents.
- Pourquoi cet algorithme trie bien le tableau. L'appliquer à

$$T = \begin{array}{|c|c|c|c|c|c|} \hline 7 & 2 & 9 & 8 & 5 & 1 \\ \hline \end{array}$$

puis écrire l'algorithme de tri correspondant. Quelle est sa complexité ?

Exercice 8. Ecrire un programme (le plus efficace possible) déterminant les éléments d'un tableau T de longueur N présents plusieurs fois dans celui-ci. Votre programme renverra son résultat dans un tableau $Tres$ (alloué au départ de taille suffisante) et une variable $nres$ indiquant le nombre de cases utilisées de $Tres$ (comme $Tres$ est créé d'une taille fixe au début du programme, toutes ses cases ne seront pas forcément utilisées).

Si l'on sait que toutes les éléments de T appartiennent à $\{0 \dots k\}$ est-ce que cela change quelque chose ? Peut-on avoir un algorithme plus rapide ?

Exercice 9 (RECHERCHE DICHOTOMIQUE DANS UN TABLEAU ORDONNE). On considère un tableau T de N nombres entiers deux à deux distincts, rangés par ordre croissant, et un nombre X . Ecrivez un programme qui détermine l'indice exprimant soit le rang de X dans T soit, si X ne figure pas dans T , le rang de l'emplacement dans lequel il faudrait ranger X pour l'insérer dans le tableau, en conservant trié ce dernier.

Principe : considérer deux indices i et j tels que le sous-tableau $[T_i \dots T_j]$ soit seul susceptible de contenir X (initialement, $i = 0$ et $j = N - 1$). En comparant X et l'élément du milieu, déterminer celle des deux moitiés du sous-tableau qui est susceptible de contenir X . Recommencer cette opération jusqu'à déterminer une unique position du tableau.

Estimez le nombre moyen d'opérations faites par ce programme et par celui de l'exercice 5. Estimez le nombre moyen d'opérations faites par l'une et l'autre méthode, lorsque la recherche d'un élément qui ne figure pas dans le tableau doit être suivie de son insertion.

Exercice 10 (FUSION). Etant donnés deux tableaux $T1$ et $T2$ (de tailles respectives $N1$ et $N2$) triés par ordre croissant, écrivez un programme qui fusionne ces tableaux dans un tableau $T3$ trié.

Par exemple :

$$T1 = \boxed{1 \mid 1 \mid 5 \mid 5 \mid 5 \mid 7 \mid 9}$$

$$T2 = \boxed{1 \mid 2 \mid 6}$$

Produiront :

$$T3 = \boxed{1 \mid 1 \mid 1 \mid 2 \mid 5 \mid 5 \mid 5 \mid 6 \mid 7 \mid 9}$$

Remarque : on suppose que l'utilisateur a bien saisi des nombres croissants (on ne le gère pas).

Bonus

Exercice 11. Ecrivez des programmes qui, étant donné un entier N , dessinent un triangle de base N , tout d'abord sous cette forme :

```
*
* *
* * *
* * * *
```

Puis sous celle-ci :

```
  *
 * *
* * *
* * * *
```