

J1MI2013: Algorithmes et Programmes: feuille 5**Tableaux(2)****Travaux pratiques**

Récupérer depuis le site de l'UE l'archive `tp05.tar.gz` et désarchivez-le. Placez vous ensuite dans le répertoire `tp05`.

Exercice 1.

Dans le fichier `testFiboTableau.c` écrire une fonction `void fibonacciTableau(int t[], int n)` qui place dans le tableau `t` (à partir de la case d'indice 0) les `n` premiers nombres de la suite de Fibonacci¹ (de $F_0 = 1$ à F_{n-1}). La valeur de `n` sera donnée sur la ligne de commande à l'exécution du programme.

Attention : on remplira le tableau directement sans utiliser de fonction écrite précédemment.

Exercice 2. Une suite de nombres est *palindrome* si elle peut être lue indifféremment de la gauche vers la droite ou de la droite vers la gauche.

Dans le fichier `testPalindrome.c` écrire une fonction `estPalindrome(int t[], int n)` qui renvoie `true` si les `n` premiers éléments contenus dans `t` constituent une suite palindrome, et renvoie `false` sinon.

Exemple :

la fonction renvoie `true` si si `t = [1,2,3,3,2,1]` ou `t = [1,2,3,5,3,2,1]`.

Si `t = [1,2,3,4,2,1]` la fonction renvoie `false`.

Exercice 3.

Dans le fichier `testFrequence.c` :

1. écrire une fonction `void tableauFrequence(int t[], int n, int freq[])` qui prend comme paramètre un tableau `t` contenant au maximum 100 nombres naturels inférieurs à 10 et calcule dans le tableau `freq` la fréquence de chaque élément de `t`. C'est-à-dire que l'élément d'indice `i` du tableau `freq` doit correspondre à la fréquence de l'entier `i` dans `t`.

Exemple : si `t=[9,7,4,0,4,3,7,4,2,2,4,9,4,0,2,8]` et `freq` est un tableau d'entiers de taille 10, après l'appel de `tableauFrequence(t, 16, freq)` on doit avoir :
`freq=[2,0,3,1,5,0,0,2,1,2]`.

2. écrire une fonction `afficheFrequence` qui affiche le contenu du tableau `freq` sous forme d'histogramme. Pour l'exemple précédent on aura :

1. définie dans la feuille de TD 2

```
0 :**
1 :
2 :***
3 :*
4 :*****
5 :
6 :
7 :**
8 :*
9 :**
```

3. Un tableau d'entiers t contenant n éléments est un *tableau de permutation* si chaque valeur entre 0 et $n - 1$ est présente dans le tableau une fois.

Exemples : $t = [1,0,3,2,5,4]$ est un tableau de permutation avec n égal à 6.

Mais $t = [1,0,3,0,4]$ n'est pas un tableau de permutation car tous les nombres entre 0 et 5 ne sont pas présents dans t .

Écrire une fonction `estTableauPermutation(int t[], int n)` qui renvoie `true` ou `false` suivant que t est tableau de permutation ou pas. La fonction devra effectuer un unique parcours de t .

Exercice complémentaire 1. Le crible d'Ératosthène est un procédé qui permet de trouver tous les nombres premiers strictement inférieurs à un certain entier naturel donné n . Pour cela, on commence par remplir un tableau t de manière croissante avec les valeurs de 2 à n . Puis, on parcourt t et pour chaque entier k qu'on rencontre, on supprime de t tous les multiples de k . Une fois arrivé à la case qui contenait initialement n , il ne reste plus dans t que les entiers premiers inférieurs ou égaux à n . Par exemple, pour $n=7$, si on remplace les valeurs supprimées par des 0, le crible d'Ératosthène nous donne le tableau $t=[2,3,0,5,0,7]$, et pour $n=15$ il nous donne $t=[2,3,0,5,0,7,0,0,0,11,0,13,0,0]$.

Dans le fichier `testEratosthene.c` compléter la fonction `int eratosthene(int tp[], int n)` qui prend en paramètre un tableau `tp` de taille n , de sorte qu'elle remplisse `tp` de manière croissante avec les entiers premiers inférieurs ou égaux à n et qu'elle renvoie le nombre d'entiers premiers inférieurs ou égaux à n . Cette fonction utilisera un tableau auxiliaire t de taille n pour effectuer le crible d'Ératosthène.

Par exemple, si $n=15$ et `tp` est un tableau de taille n , `eratosthene(tp, n)` renvoie 6 et après son appel on doit avoir `tp=[2,3,5,7,11,13,*,*,*,*,*,*,*,*,*]` où `*` représente des valeurs qu'on ne connaît pas.

À l'aide de la fonction `afficherTableau` déjà implémentée dans le fichier `eratosthene.c`, afficher les entiers premiers inférieurs ou égaux à un entier n . La valeur de n sera donnée sur la ligne de commande à l'exécution du programme.