

UE Algorithmes et programmation

L1/S2 – TD/TP n° 5

Tableaux et matrices – Année 2015

Vous pouvez récupérer du code ici :

<http://lab-sticc.univ-brest.fr/~rodin/FTP/Enseignements/L1/AlgoEtProg>

Sans indication particulière, les fonctions ne doivent pas être écrites sous forme récursives. La récursivité ne sera utilisée que lorsque cela sera indiqué explicitement dans le sujet.

TD

1 TD Exercice 1 : des fonctions sur des tableaux

Le but de cet exercice est d'écrire plusieurs fonctions travaillant sur des tableaux d'entiers.

1. Ecrire une fonction `printTab` permettant d'afficher le contenu d'un tableau d'entiers passé en paramètre.
2. Ecrire une fonction `readTab` prenant un tableau d'entiers en paramètre. Cette fonction permet de saisir (via des `scanf`) des entiers qui seront stockés dans le tableau. Toutes les cases du tableau seront ainsi remplies à l'aide de `scanf` successifs.
3. Ecrire une fonction `recherche` permettant de rechercher dans un tableau (passé en paramètre) une valeur (elle-même passée en paramètre). Cette fonction devra retourner 1 (vrai) ou 0 (faux) en fonction du résultat de la recherche. Si la valeur cherchée est présente dans le tableau, cette fonction devra (à l'aide d'un paramètre d'Entrée/SORTIE) fournir l'indice auquel la valeur se trouve dans le tableau.
4. Ecrire une fonction `combienDeVal` qui compte le nombre cases du tableau passé en paramètre qui ont la valeur `val`.
5. Ecrire un programme principal permettant de tester les 4 fonctions `printTab`, `readTab`, `combienDeVal` et `recherche`. Ce programme principal devra donc contenir la déclaration d'un tableau d'une certaine taille et l'appel à ces différentes fonctions.
6. Ecrire maintenant une fonction `estTrie` prenant un tableau un paramètre. Cette fonction retournera 1 (vrai) si tous les éléments du tableau sont rangés dans l'ordre croissant. Elle retournera 0 (faux) dans le cas contraire.
7. Ecrire une fonction `rechercheDichotomique` permettant de rechercher dans un tableau (passé en paramètre) une valeur (elle-même passée en paramètre). Cette fonction devra retourner 1 (vrai) ou 0 (faux) en fonction du résultat de la recherche. Si la valeur cherchée est présente dans le tableau, cette fonction devra (à l'aide d'un paramètre d'Entrée/SORTIE) fournir l'indice auquel la valeur se trouve dans le tableau.

On considère ici que le tableau passé en paramètre doit être trié par ordre croissant. La recherche dichotomique s'effectue comme suit. Pour rechercher une valeur dans le tableau, on compare cette valeur à la valeur se trouvant au milieu du tableau.

- Si c'est la même, on a trouvé!
 - Si la valeur se trouvant au milieu est plus petite que la valeur cherchée, alors il faut chercher dans la partie droite du tableau.
 - Sinon alors il faut chercher dans la partie gauche du tableau.
- On recommence jusqu'à avoir trouvé la valeur ou bien si l'espace de recherche est nul.

2 TD Exercice 2 : petites fonctions sur des matrices

Le but de cet exercice est d'écrire plusieurs fonctions travaillant sur des matrices d'entiers.

1. Ecrire une fonction `printMat` permettant d'afficher le contenu d'une matrice 3 x 3.
2. Ecrire une fonction `additionMat` permettant de calculer l'addition de 2 matrices 3 x 3.
3. Ecrire une fonction `estSymetriqueMat` permettant de savoir si une matrice CARRÉE 3 x 3 est symétrique ou non. Cette fonction devra retourner 1 (vrai) si la matrice est symétrique ou 0 (faux) sinon.

	<code>a</code>	<code>d</code>	<code>e</code>
est symétrique ou 0 (faux) sinon.	<code>d</code>	<code>b</code>	<code>f</code>
Rappel : une matrice 3 x 3 est symétrique si :	<code>e</code>	<code>f</code>	<code>c</code>
4. Ecrire un programme principal permettant de tester ces fonctions sur les matrices.

TP

1 TP Exercice 1 : des fonctions sur des tableaux

1. Programmer l'Exercice 1 du TD : `printTab`, `readTab`, `estTrie`, `rechercheDichotomique` et un programme principal permettant de tester ces fonctions.
2. Programmer une fonction qui permet d'obtenir la moyenne des valeurs contenues dans un tableau passé en paramètre.
3. Programmer une fonction qui permet d'obtenir le nombre de nombres pairs et le nombre de nombres impairs présents dans un tableau passé en paramètre.
4. Programmer un programme principal permettant de tester les 2 fonctions précédentes.

2 TP Exercice 2 : petites fonctions sur des matrices

1. Programmer la fonction `printMat` vue lors de l'Exercice 2 du TD.
2. Programmer une fonction `readMat` permettant de saisir toutes les cases d'une matrice passée en paramètre. Des `scanf` successifs seront nécessaires.
3. Programmer la fonction `additionMat` vue lors de l'Exercice 2 du TD.
4. programmer une fonction `estNulleMat` permettant de savoir si une matrice 3 x 3 est nulle (ne comporte que de 0 ou non). Cette fonction devra retourner 1 (vrai) si la matrice est nulle ou 0 (faux) sinon.
5. programmer un programme principal permettant de savoir si l'addition de 2 matrices saisies avec `readMat` aboutit à une matrice nulle.

3 TP Exercice 3 :

Eventuellement, programmer les algorithmes restant du TD.