Université Sultan Moulay Slimane
Ecole National des Sciences Appliquées
Khouribga

Module : Architecture des ordinateurs et Algorithmique
1ère API / S1 / Année 2018/2019

Feuille de Travaux Dirigés N° 6

Exercice 1

Écrire un algorithme qui demande la saisie des éléments d'un tableau T de 10 nombres réels, puis affiche les éléments de T en ordre croissant et décroissant des indices.

Exercice 2

Écrire un algorithme qui détermine respectivement le plus grand et le plus petit élément d'un tableau T de 10 nombres entiers, ainsi que leurs rangs.

Exercice 3

Écrire un algorithme qui calcule respectivement la somme et la moyenne des éléments d'un tableau T de 10 nombres réels.

Exercice 4

Écrire un algorithme qui détermine respectivement, le nombre des éléments nuls, le nombre des éléments positifs et le nombre des éléments négatifs dans un tableau T de 10 nombres entiers.

Exercice 5

Écrire un algorithme qui saisi un rang k ($1 < k < 10$), puis effectue la division entière de tous les éléments d'un tableau T de 10 entiers par T[k], si ce dernier n'est pas nul.

Exercice 6

Écrire un algorithme qui demande la lecture des éléments d'un tableau T de 10 nombres entiers, puis renverse l'ordre des éléments du tableau T.

Exercice 7

Écrire un algorithme qui teste si un tableau T d'entiers est identique à son renversé (on dit que T est un tableau **palindromique**). Par exemple, le tableau T = {2, 5, 9, -3, 0, -3, 9, 5, 2} est palindromique.

Exercice 8

Écrire un algorithme qui teste si un tableau T est trié dans l'ordre croissant.

Exercice 9

Algorithme de recherche séquentielle ou linéaire (*Linear Search Algorithm*).

Écrire un algorithme qui cherche si un élément e appartient à un tableau T de N éléments. Dans cet algorithme, on parcourt le tableau T (de gauche à droite par exemple), jusqu'à trouver l'élément e. Le coût de cet algorithme est N comparaisons dans le pire des cas. C'est pour cette raison que l'algorithme se nome par « recherche linéaire ».

Exercice 10

Algorithme de recherche dichotomique ou binaire (*Binary Search Algorithm*).

Écrire un algorithme qui cherche si un élément e appartient à un tableau T **trié** de N éléments. Dans cet algorithme, le tableau T étant trié, on procède comme suit : on compare l'élément e avec l'élément qui se trouve au milieu du tableau T (soit m cet élément). Il y a alors trois cas à considérer :

- Si e = m : l'élément e se trouve dans T et l'algorithme se termine.
- Si e < m : on cherche alors de la même manière e dans le demi-tableau gauche de T.
- Si e > m : on cherche alors de la même manière e dans le demi-tableau droit de T.

Le coût de cet algorithme est $Log_2(N)$ comparaisons dans le pire des cas. Par exemple, si le tableau T contient 1024 éléments, alors on fera dans le pire des cas :

- 1024 comparaisons avec l'algorithme de recherche séquentielle.
- $Log_2(1024) = 10$ comparaisons avec l'algorithme de recherche binaire.

Exercice 11

Écrire un algorithme qui effectue le calcul matriciel sur deux matrices A et B carrées d'ordre N :

- Lecture des deux matrices A et B.
- Affichage des deux matrices A et B.
- Calcul de la somme des deux matrices A et B.
- Calcul de la différence des deux matrices A et B.
- Calcul du produit par un scalaire des deux matrices A et B.
- Calcul du produit des deux matrices A et B.
- Calcul de la trace de la matrice A. La trace d'une matrice est la somme de ses éléments diagonaux.
- Calcul de la transposée de la matrice B. La transposée d'une matrice est obtenue en transformant les lignes en colonnes.