

FACULTÉ DES SCIENCES DE GABÈS

DÉPARTEMENT INFORMATIQUE

Algorithmique et Structures de Données 1

TD4: Les tableaux

Sections: LGSI1/LIRIS1

A.U: 2023/2024

Exercice 1

Soit un tableau A de n nombres triés par ordre croissant. Ecrire une procédure qui permet de lire un réel R et l'insérer dans sa bonne position. Le résultat sera un deuxième tableau B de longueur (n+1) et trié également par ordre croissant.

Exercice 2

Ecrire une procédure permettant d'éclater un tableau T contenant n éléments de type entier en deux tableaux :

- TP qui contiendra les éléments positifs de T
- TN qui contiendra les éléments négatifs de T

Exercice 3

Soit T1 et T2 deux tableaux de dimensions N et M et non triés. On vous demande d'écrire les procédures/fonctions suivantes :

1. Une fonction **SAISIE_TAILLE** permettant de lire la dimension DIM ($1 < DIM \leq 50$).
2. Une procédure **SAISIE_TAB** à deux paramètres T et DIM permettant de remplir le tableau T par des entiers.
3. Une procédure **TRI** à deux paramètres T et DIM qui permet de trier un tableau T par ordre croissant (utiliser une procédure **PERMUTER** à deux paramètres X et Y qui permet de permuter X et Y).
4. Une procédure **FUSION** à 5 paramètres T1, T2, F, N et M qui construit un tableau F trié par ordre croissant obtenu à partir les éléments de deux tableaux T1 (dimension N) et T2 (dimension M) qui sont déjà triés par ordre croissant. Le tableau F aura la dimension N+M. Indication : Utiliser trois indices I1, I2 et IF. Comparer T1[I1] et T2[I2] ensuite remplir F[IF] par le plus petit des deux éléments et avancer dans le tableau F et dans le tableau qui a contribué son élément. Lorsque l'un des deux tableaux T1 ou T2 est épuisé, il suffit de recopier les éléments restants de l'autre tableau dans le tableau F.
5. Une procédure **AFFICHE** à deux paramètres T et DIM qui affiche les composantes du tableau T.
6. Ecrire un algorithme principal pour tester les 5 procédures.

Exercice 4

Une association de consommateurs fait une étude comparative de prix entre deux supermarchés. Cette étude a pour but de déterminer :

- le prix d'un panier donné que propose le premier supermarché,
- le prix du panier que propose le second supermarché,
- le prix idéal en prenant pour chacun des produits du panier son prix le plus bas.

On suppose que le panier, objet de l'étude, est formé de N produits ($2 < N < 20$) disponibles dans les deux supermarchés.

On vous demande d'écrire les fonctions suivantes :

1. Une procédure **CHARGEMENT** à deux paramètres T et N permettant de saisir pour les N produits, les prix d'un supermarché. (Les prix sont des réels strictement positifs)
2. Une fonction **SOMME** à deux paramètres T et N permettant de calculer et retourner la somme des éléments d'un tableau de N réels.
3. Une fonction **MINI** à deux paramètres X et Y permettant de retourner le plus petit des deux réels X et Y.
4. Une fonction **IDEAL** à trois paramètres T1, T2 et N permettant de calculer et retourner pour le panier en question, son prix idéal en prenant pour chacun des produits le prix le plus bas.
5. Une procédure **AFFICHE** à trois paramètres T1, T2 et N permettant d'afficher :
 - le prix du panier d'après le premier supermarché,
 - le prix du panier d'après le second supermarché,
 - le prix idéal du même panier.
6. En utilisant les procédures/fonctions définies précédemment, écrire l'algorithme principal qui donne une solution au problème posé par cette association de consommateurs.

Exercice 5

Soit un tableau T de dimension N contenant des valeurs quelconques.
On vous demande d'écrire les procédures/fonctions suivantes :

1. Une procédure **SAISIE_TAB** à deux paramètres T et N permettant de remplir le tableau TAB par des entiers.
2. Une procédure **AFFICHER** à deux paramètres T et N permettant d'afficher les composantes du tableau TAB.
3. Une procédure **RECHERCHE** à deux paramètres T et N permettant de chercher et afficher le nombre d'occurrence du plus petit élément dans T et l'indice de sa première occurrence.
4. Une fonction **DETERMINER** à deux paramètres T et N permettant de retourner le nombre d'éléments distincts du tableau.
5. Une procédure **ELIMINER** à deux paramètres T et N permettant d'éliminer les éléments doubles en faisant des décalages vers la gauche des éléments du tableau sans utiliser un autre tableau.
6. L'algorithme principal permettant d'appeler les fonctions ci-dessus.

Exercice 6

Soit un tableau T de dimension NMAX. On vous demande d'écrire les procédures/fonctions suivantes :

1. Une procédure **SAISIE_TAB** permettant de remplir un tableau par des entiers strictement positifs.
2. Une procédure **AFFICHER** permettant d'afficher les composantes d'un tableau.
3. Ecrire une fonction **TESTER** permettant de vérifier et retourner :

$$\begin{cases} 1 : \text{ si le tableau est croissant,} \\ 2 : \text{ si le tableau est décroissant,} \\ 3 : \text{ si le tableau est constant ou} \\ 4 : \text{ si le tableau est quelconque.} \end{cases}$$

4. Une fonction **DETERMINER** permettant de chercher et retourner le nombre d'occurrence d'un élément donné.
5. Une procédure **TRIER** permettant de trier le tableau.
6. Une procédure **COMPRESSER** permettant de compresser le tableau initial dans un nouveau tableau de la manière suivante : Insérer après chaque élément son nombre d'occurrence. Utiliser les questions 4, 5 et 6.
7. L'algorithme principal permettant d'appeler les procédures/fonctions ci-dessus permettant de saisir la taille, charger T, le tester, le compresser et afficher T1.

Exercice 7

Dans cet exercice, on considéra :

- Une matrice carré M contenant au maximum longueurmax lignes et longueurmax colonnes, longueurmax étant déclarée comme constante (on prendra par exemple longueurmax = 5)

- Un entier nb saisi au clavier et représentant le nombre de lignes et celui des colonnes de la matrice M, ce nombre doit être strictement positif et ne doit pas dépasser la valeur longueurmax.

Ecrire un algorithme permettant de :

- Saisir au clavier les (nb * nb) valeurs de la matrice.
- Lire un indice ind.
- Calculer la somme des valeurs ayant pour indice de ligne ind.
- Calculer le produit des valeurs ayant pour indice de colonne ind.
- Afficher les résultats.

Exercice 8

Soient M1 et M2 deux matrices à n lignes et m colonnes.

Ecrire une procédure Somme_Matrice qui calcule les éléments de la matrice M3=M1+M2.

Exercice 9

On vous demande d'écrire les procédures/fonctions suivantes :

1. Une procédure SAISIE_NB permettant de saisir un entier entre 0 et 10.
2. Une procédure CHARGEMENT permettant de remplir une matrice.
3. Une procédure MULTIPLICATION qui effectue la multiplication de deux matrices A (dimensions N et M) et B (de dimensions M et P) en une troisième matrice C (dimensions N et P)
La multiplication de deux matrices se fait en multipliant les composantes des deux matrices lignes par colonnes.

$$C_{ij} = \sum_{k=1}^{k=M} A_{ik} * B_{kj} \quad (1)$$

4. Une procédure AFFICHAGE permettant d'afficher les éléments d'une matrice.
5. l'algorithme principal permettant :
 - Saisir les dimensions des deux matrices,
 - charger deux matrices M1 et M2 avec des entiers.
 - multiplier M1 et M2 pour obtenir M3,
 - afficher le contenu des matrices M1, M2 et M3.

Exercice 10

Soit A une matrice carrée de M lignes et M colonnes contenant des valeurs numériques quelconques.

Ecrire un algorithme permettant de :

- saisir les dimensions de la matrice A,
- saisir les éléments de la matrice A et
- vérifier et afficher si la matrice A est symétrique par rapport à la première ou la seconde diagonale.
(N.B : utilisez deux variables booléennes sym1 et sym2)

Exercice 11

Nous souhaitons calculer le rang de chaque étudiant après avoir calculer sa moyenne générale. Nous disposons d'une matrice NOTE (20 lignes, 4 colonnes) dont 20 représente le nombre des étudiants, la première colonne comporte le Numéro d'Inscription de l'Etudiant et les trois autres colonnes correspondent aux notes de l'étudiant dans trois matières. Chaque étudiant dispose de trois notes de même coefficients.

Pour résoudre ce problème, on vous demande d'écrire les procédure/fonctions suivantes :

1. Procédure SAISIE(NOTE : TAB1 ; L, C : entier) permettant de remplir la matrice NOTE par les numéros d'inscriptions des étudiants (N.I.E) et leurs notes par matière.
2. Procédure CALCUL(NOTE : TAB1 ; MOYENNE : TAB2 ; L,C : entier) permettant de calculer la moyenne de chaque étudiant. Les moyennes calculées seront stockées dans la matrice MOYENNE dont La première colonne contient le numéro d'inscription de l'étudiant (N.I.E) et la deuxième contient sa moyenne.
3. Fonction MOYENNE_CLASSE(MOYENNE : TAB2 ; L : entier) permettant de calculer et retourner la moyenne de la classe.

4. Fonction TAUX_REUSSITE(MOYENNE : TAB2 ; L : entier) permettant de calculer et retourner le taux de réussite de la classe sachant que ce dernier est égal au quotient du nombre des étudiants ayant une moyenne ≥ 10 par le nombre total des étudiants.
5. Procédure TRI_MOYENNE(MOYENNE : TAB2 ; L : entier) permettant de trier les moyennes selon l'ordre décroissant en leurs associant les numéros d'inscriptions. C'est le tri de la matrice MOYENNE. (utiliser la méthode tri par sélection).
6. Procédure AFFICHE(MOYENNE : TAB2 ; L : entier) permettant d'afficher les numéros d'inscription et les moyennes des étudiants de la matrice MOYENNE.
7. Fonction RANG(MOYENNE : TAB2 ; L : entier ; NIE : réel) permettant de déterminer et retourner le rang d'un étudiant donné.
8. L'algorithme principal qui fait appel aux procédures/fonctions ci dessus pour :
 - remplir les matrices NOTE et MOYENNE,
 - afficher les numéros d'inscriptions des étudiants et leurs moyennes avant le tri,
 - afficher la moyenne de la classe,
 - afficher le taux de réussite de la classe,
 - afficher les numéros d'inscriptions des étudiants et leurs moyennes après le tri.
 - Saisir un N.I.E et afficher son rang.