Centre universitaire de Mila

Institut des Sciences et de la Technologie

Département de Mathématique et Informatique

1ère Année LMD MI

**Série de Td N°5**

Module : Algorithmique et structures de données1 Année universitaire : 2018/2019

**Exercice 1**

Ecrire un algorithme qui permet de remplir un vecteur de **n** valeurs entières, puis afficher les valeurs de ce vecteur. **n** est donné par l’utilisateur.

**Exercice 2**

Réalisez un algorithme qui permet de calculer puis afficher la somme, la moyenne, le maximum et le minimum des éléments d’un tableau d’entiers:

***Exemple :***

**Exercice 3**

***SommeT  : 24***

***MoyenneT  : 4,8***

***MaxT  : 9***

***MinT  : 1***

**Calcul**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | 3 | 1 | 7 | 9 |

Ecrire un algorithme qui lit un vecteur de n valeurs entières et vérifier s’il est trié dans l’ordre croissant ou non.

**Exercice 4**

Soit V un vecteur de N valeurs entiers. Construire à partir de ce vecteur deux vecteurs **V\_PAIR** et **V\_IMPAIR**. Avec le premier contenant les éléments pairs de V et le second les éléments impairs de V. puis afficher les éléments des vecteurs V\_PAIR et V\_IMPAIR.

**Exercice 5**

Le calcul des éléments d’une matrice **C**, produit de deux matrices **A** et **B**  respectivement de dimensions (n, r) et (r, m), est donnée par la relation suivante : **cijik \* bkj**

Ecrire un algorithme permettant de construire la matrice **C** produit de deux matrices **A** et **B**.

**Exercice 6**

Ecrire un algorithme qui permet de dire si une matrice est dominée par les nombres pairs, impairs ou équilibrée.

**Solution**

**Exercice 1**

***créer-tab***

Vsupplementaire)e(TireTherT : Tableau d’entiers

n : entier

**Rôle** : Lire et afficher un vecteur ***V***de ***n*** éléments entiers

**Algorithme *créer-tab***

***Const max🡨 50 ;***

V [max]:tableau d’entiers

n : entier ;

i : entier;

**Début**

**Ecrire (”**donnez le nombre d’éléments du tableau: ”**)**

**Lire (n) ;**

**Pour**i allant de 1 à **n faire**

Ecrire **(”**donnez l’éléments N°”,i**)**

Lire (V[i]) ;

**Finpour**

Pour **i** allant de **1** à **n** faire

Ecrire (v[i]) ;

Finpour

**Fin.**

**Exercice 2**

***SommeT  : 24***

***MoyenneT  : 4,8***

***MaxT  : 9***

***MinT  : 1***

***Calcul***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | 3 | 1 | 7 | 9 |

**Algorithme** *Calcul*

***Const max🡨 50 ;***

V [max]:tableau d’entiers

n : entier ;

SommeT, MoyenneT, MaxT, MinT : entier ;

i : entier;

**Début**

Lire (n);

Pouri allant de 1 à n faire

Lire (V[i]) ;

Finpour

SommeT**←T[1];**

MaxT**←** T [1];

MinT**←** T [1]**;**

**Pour**i allant de 2 à n **faire**

**Si** T[i]>Max **alors**

Max← T[i] ;

**Finsi**

**Si** T[i]<Min **alors**

Min← T[i] ;

**Finsi**

SommeT**←**SommeT+ T[i] ;

**Finpour**

MoyenneT**←**SommeT / n ;

**Fin**.

**Exercice 3**

**Rôle** : lire et vérifier le vecteur V est trié dans l’ordre croisant ou non

**Vérifier**

V supplementaire)e(TireTherT: Tableau d’entiers

Rsupplementaire)e(TireTherT : Booléen

n : entier

Algorithme :

Algorithme Ex03

T [100] : tableau de entier ;

n: entier ;

i : entier;

R: booléen;

**Début**

Lire (n);

**Pour** i allant de 1 à **n faire**

Lire (V[i]) ;

**Finpour**

i ← 1; R ← Vrai;

**Tantque** (i ≤ n-1) et (R = Vrai) **faire**

**Si** (V[i] >V [i+1]) **alors**

R ← Faux;

**FinSi**

i ← i + 1;

**Fintantque**

**Si** (R = vrai) **alors**

Ecrire ( ”le tableau est trié dans un ordre croisant”) ;

**Sinon**

Ecrire ( ”le tableau n’est pas trié dans un ordre croisant”) ;

**finsi**

Fin.

**Exercice 4**

**Algorithme** *pair\_impair*

***Const max🡨 50 ;***

V [max]:tableau d’entiers

V\_PAIR [max]:tableau d’entiers //pour les éléments pairs

V\_IMPAIR [max]:tableau d’entiers //pour les éléments impairs

n : entier ;

SommeT, MoyenneT, MaxT, MinT : entier ;

i,j,k : entier;

**Début**

Lire (n);

Pouri allant de 1 à n faire

Lire (V[i]) ;

Finpour

j ← 0 ;

k ← 0 ;

**Pour** i allant de 1 à n **faire**

**Si** (V[i] mod 2 = 0)**alors**

j ← j+1 ;

V\_PAIR [j]← V[i] ;

**sinon**

k ← k+1 ;

V\_IMPAIR [k]← V[i] ;

**Finsi**

**Finpour**

**Ecrire (”**les éléments pairs sont : ”**) ;**

Pour **i** allant de **1** à **j** faire

**ecrire (V\_PAIR [i]) ;**

Finpour

**Ecrire (”**les éléments impairs sont : ”**) ;**

Pour **i** allant de **1** à **k** faire

**ecrire (V\_IMPAIR [i]) ;**

Finpour

**Fin**.

**Exercice 5**

**Rôle** : Construire la matrice **C** produit de deux matrices **A** et **B**

ProduitMat

Csupplementaire)e(TireTherT : Tableau d’entiers

n, r, m: entier

A, B : Tableau d’entier

**Algorithme** **Produit\_Mat**

**Const n ← 5 ; r ← 3 ; m ← 4 ;**

A [n,r] : tableau d’entiers ;

B [r,m] : tableau d’entiers ;

C [n,m] : tableau d’entiers ;

i, j, k, S : entier;

**Début**

**pour** i allant **de** 1 **à** n **faire**

**pour** j **allant** de 1 à **m** **faire**

S ←0;

**pour** k allant **de** 1 **à** r **faire**

S ← S+ A [i,k] \* B[k,j] ;

**FinPour**

C [i,j] ← S;

**Finpour**

**Finpour**

**Fin**.

**Exercice 6 :**

**Algorithme** **Mat\_dominée**

**Const n ← 5 ; m ← 4 ;**

M [n,m] : tableau d’entiers ;

i, j, p,imp : entier;

**Début**

// on suppose que la matrice est déjà lue

p ← 0;

imp ← 0;

**pour** i allant **de** 1 **à** n **faire**

**pour** j **allant** de 1 à **m** **faire**

**si (**M[i][j] mod 2 = 0**)alors**

p ← p +1;

**sinon**

imp ← imp +1;

**Finsi**

**Finpour**

**Finpour**

**Si (**p = imp**) alors**

**Ecrire (**’la matrice est équilibrée’**) ;**

**Sinon**

**Si (**p>imp**)alors**

**Ecrire(‘** la matrice est dominée par les nombres pairs**’) ;**

**Sinon**

**Ecrire(‘** la matrice est dominée par les nombres impairs**’) ;**

**Finsi**

**finsi**

**Fin**.