

Examen final – Informatique 2

(Durée : 02 heures) ;

Exercice 1 :

Soit V un vecteur de taille N. Ecrire un Programme Pascal qui lit le vecteur V, divise chaque élément de V par le k^{ème} élément et affiche le vecteur résultat V.

Exercice 2 :

Soit l’algorithme suivant qui permet d’insérer une valeur X, introduite par l’utilisateur dans la première position du tableau T.

```

Algorithme insertion
Variables T : tableau [1..50] de réel
n,i :entier
X :réel
Début
Lire(n)
Pour i ← 1 à n faire
    Lire(T[i])
Fin pour
Lire(X)
i ← n
Tant-que i >= 1 faire
    T[i+1] ← T[i]
    i ← i-1
fin tant-que
T[1] ← X
Pour i ← 1 à n+1 faire
    Ecrire(T[i])
Fin pour
Fin.
    
```

- 1- Traduire l’algorithme en programme Pascal.
- 2- Modifier le programme pour insérer la valeur X dans la position K.

NB : les valeurs de X et K sont introduites par l’utilisateur.

Exercice 3 :

Soit une matrice A[n,m] d’entier. Ecrire un programme Pascal qui calcul la matrice B, tel que chaque composante B[i,j] de la matrice B est le factoriel de la composante A[i,j] de la matrice A.

Exemple

$$A = \begin{bmatrix} 3 & 1 & 6 \\ 4 & 7 & 2 \end{bmatrix} ; \quad B = \begin{bmatrix} 3! & 1! & 6! \\ 4! & 7! & 2! \end{bmatrix}$$

Exercice 4 :

Soit A une matrice carré de taille n et de type réel. La séquence d’instructions suivante réalise le calcul de la somme S des composantes de l’anti-diagonale de A.

```

S := 0 ;
For i := 1 to n do
    S := S+A[ i , n-i+1 ] ;
    
```

Eléments de l’anti-diagonale

| | | | | | |
|--------|----------|--------|----------|----------|--------|
| A[1,1] | A[1,2] | A[1,3] | | A[1,n-1] | A[1,n] |
| A[2,1] | A[2,2] | | | A[2,n-1] | A[2,n] |
| A[3,1] | | | A[3,n-2] | | |
| | | | | | |
| | A[n-1,2] | | | | |
| A[n,1] | A[n,2] | | | A[n,n-1] | A[n,n] |

- 1- Ecrire un sous programme fonction nommé min_antidiagonale qui permet de recherché le plus petit élément de l’anti-diagonale de A.
- 2- Insérer cette fonction dans un programme Pascal qui lit une matrice A, appel la fonction min_antidiagonale et qui affiche le minimum de l’anti-diagonale.
- 3- Récrire le programme en transformant la fonction en procédure.

EX02: Q1:

Program Insertion;

USES WINCRT;

VAR T: array[1..50] of real;
N, i: integer; x: real;

Begin

Read(N);

for i := 1 to N do

 Read(T[i]);

Read(x);

i := N;

while i >= 1 do

 Begin

 T[i+1] := T[i];

 i := i - 1;

 end;

T[1] := x;

for i := 1 to (N+1) do

 write(T[i]);

END.

EX02: Q2:

Program Insertion;

USES WINCRT;

VAR T: array[1..50] of real;
N, i, k: integer; x: real;

Begin

Read(N);

for i := 1 to N do

 Read(T[i]);

Read(x);

Read(k);

i := N;

while i >= k do

 Begin

 T[i+1] := T[i];

 i := i - 1;

 end;

T[k] := x;

for i := 1 to (N+1) do

 write(T[i]);

END.

0,5

0,5

0,5

0,5

0,5

0,5

0,5

1

0,5

Exo 1^a

Program Exo 1^a

USES WINCRT;

Var V: array [1..100] of real; } 0,5
i, N, k: integer;

X: real; } 0,5

begin

Read (N); } 0,5

for i := 1 to N do
 Read (V[i]); } 0,5

Read (k); } 0,5

X := V[k]; } 0,5

for i := 1 to N do
 V[i] := V[i] / X; } 1,5

for i := 1 to N do
 write (V[i]); } 0,5

END.

EX03:

Program EX03:

USES WINCRT;

var A, B: array [1..50, 1..50] of integer; (0,5)

F, k, i, j, n, m: integer;

begin ↑ (0,5)

read(n, m);

for i := 1 to n do
 for j := 1 to m do
 read(A[i, j]); (0,5)

for i := 1 to n do
 for j := 1 to m do

begin

F := 1; ↓ (1)

for k := 2 to A[i, j] do

F := F * k; (0,5)

B[i, j] := F; (0,5)

end;

for i := 1 to n do

begin

for j := 1 to m do

write(B[i, j]);

writeln;

end;

END.

(0,5)

Exo 4: (5)

1) Function $\text{min_antidiagonale}(A: \text{Mat}; n: \text{integer}): \text{real};$

Var $i: \text{integer}; \text{min}: \text{real};$

Begin

$\text{min} := A[1, n];$

for $i := 1$ to n do

if $A[i, n-i+1] < \text{min}$ then

$\text{min} := A[i, n-i+1];$

$\text{min_antidiagonale} := \text{min};$

end;

2) Program Exo4;

USES wincrt;

type $\text{Mat} = \text{array}[1..50, 1..50] \text{ of real};$

var $A: \text{Mat}; i, j, N: \text{integer}; Mn: \text{real};$

Function $\text{min_antidiagonale}(A: \text{Mat}; n: \text{integer}): \text{real};$

var $i: \text{integer}; \text{min}: \text{real};$

Begin

$\text{min} := A[1, n];$

for $i := 1$ to n do

if $A[i, n-i+1] < \text{min}$ then

$\text{min} := A[i, n-i+1];$

$\text{min_antidiagonale} := \text{min};$

end;

Begin

read(N);

for $i := 1$ to n do

for $j := 1$ to n do
read($A[i, j]$);

$Mn := \text{min_antidiagonale}(A, n);$

write(Mn);

0,5

0,5

0,5

0,5

1

3) Program Ex04; (15)

USES WINCRT;

type Mat = array [1..50, 1..50] of real;

var A: Mat; i, j, n: integer; Mn: real;

~~function min_anti_diagonale~~

procedure min_anti_diagonale (A: Mat; n: integer; var min: real);

var i: integer;

begin

min := A[1, n];

for i := 1 to n do

if A[i, n-i+1] < min then

min := A[i, n-i+1];

end;

begin

read(n);

for i := 1 to n do

for j := 1 to n do

read(A[i, j]);

min_anti_diagonale(A, n, Mn);

write(Mn);

END.

(0,5)

(1)