

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.

Exo 1:

Program Exo 1:

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.

1. Écrire un sous-programme fonction nommée `min_antidiagonale` qui permet de rechercher le plus petit élément de l'anti-diagonale de A.
2. Insérer cette fonction dans un programme Pascal qui lit une matrice A, appelle la fonction `min_antidiagonale` et qui affiche le minimum de l'anti-diagonale.
3. Réécrire le programme en transformant la fonction en procédure.

Exo 2: 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;
```

```
T[1] := x;
for i := 1 to (N+1) do
  write(T[i]);
```

END.

Exo 2: 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;
```

```
T[k] := x;
for i := 1 to (N+1) do
  write(T[i]);
```

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

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]); 0,5
writeln;
end;

END.

Ex04: 2

1) Function min_antidiagonal (A: Mat; n: integer): real;

Var i: integer; min: real; 0,5

Exo 4: (2)

```
1) Function min_antidiagonale (A: Mat; n: integer): real;  
   Var i: integer; min: real;  
   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];  
     min_antidiagonale := min;  
   End;
```

2) Program Exo4;

USES WinCrt;

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

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

Function min_antidiagonale (A: Mat; n: integer): real;

var i: integer; min: real;

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];

min_antidiagonale := min;

End;

Begin

Read(N);

for i := 1 to n do

for j := 1 to n do

Read(A[i, j]);

Mn := min_antidiagonale(A, n);

Write(Mn);

3) Program Exo4;

USES WinCrt;

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