

## Matrice symétrique

\* مصفوفة عناصرها متناظرة بالنسبة للقطر الرئيسي \*

X	A	B	C	D
A	X	E	F	G
B	E	X	H	I
C	F	H	X	J
D	G	I	J	X

### Algorithme Matrice\_Sym

```

Const   N=20
Var     A : tableau [1..N,1..N] d'entiers
        i,j : entier
        Sym : booléen

Début
    pour i ← 1 à N faire
        pour j ← 1 à N faire
            Lire(A[i,j])
        Finpour
    Sym ← vrai
    i ← 1
    TQ (i ≤ N-1) et (Sym) faire
        j ← i+1
        TQ (j ≤ N) et (Sym) faire
            Si (A[j,i] ≠ A[i,j]) alors
                Sym ← faux
            Fsi
            j ← j+1
        FinTQ
        i ← i+1
    FinTQ
    Si (Sym) alors
        Ecrire (" la matrice est symétrique")
    Sinon
        Ecrire (" la matrice n'est pas symétrique")
    FinSi
Fin.

```

```

i ← 2
TQ (i ≤ N) et (Sym) faire
    j ← 1
    TQ (j < i) et (Sym) faire
        Si (A[j,i] ≠ A[i,j]) alors
            Sym ← faux
        Fsi
        j ← j+1
    FinTQ
    i ← i+1
FinTQ

```

استعمال  
المثلث  
السفلي

## Matrice diagonale

\* مصفوفة عناصرها معدومة عدا القطر الرئيسي \*

i = j

X	0	0	0	0
0	X	0	0	0
0	0	X	0	0
0	0	0	X	0
0	0	0	0	X

### Algorithme Matrice\_Diag

```

Const   N=20
Var     A : tableau [1..N,1..N] d'entiers
        i,j : entier
        Diag : booléen

Début
    pour i ← 1 à N faire
        pour j ← 1 à N faire
            Lire(A[i,j])
        Finpour
    Diag ← vrai

    i ← 1
    TQ (i ≤ N) et (Diag) faire
        j ← 1
        TQ (j ≤ N) et (Diag) faire
            Si (A[i,j] ≠ 0) et (i ≠ j) alors
                Diag ← faux
            Fsi
            j ← j+1
        FinTQ
        i ← i+1
    FinTQ

    Si (Diag) alors
        Ecrire (" la matrice est diagonale")
    Sinon
        Ecrire (" la matrice n'est pas diagonale")
    FinSi
Fin.

```

## Matrice Tri-diagonale

\* مصفوفة عناصرها معدومة عدا القطر الرئيسي و القطران المجاوران له \*

X	X	0	0	0
X	X	X	0	0
0	X	X	X	0
0	0	X	X	X
0	0	0	X	X

### Algorithme Matrice\_TRI\_Diag

```

Const    N=20
Var      A : tableau [1..N,1..N] d'entiers
          i,j : entier
          TDiag : booléen

Début
  pour i ← 1 à N faire
    pour j ← 1 à N faire
      Lire(A[i,j])
    Finpour
  Finpour
  TDiag ← vrai

  i ← 1
  TQ (i ≤ N) et (TDiag) faire
    j ← 1
    TQ (j ≤ N) et (TDiag) faire
      Si (A[i,j] ≠ 0) et (i ≠ j) et (i-1 ≠ j) et (i+1 ≠ j) alors
        TDiag ← faux
      Fsi
      j ← j+1
    FinTQ
    i ← i+1
  FinTQ

  Si (TDiag) alors
    Ecrire (" la matrice est Tri-diagonale")
  Sinon
    Ecrire (" la matrice n'est pas Tri-diagonale")
  FinSi
Fin.
```

## Matrice diagonale (diagonale secondaire)

\* مصفوفة عناصرها معدومة عدا القطر الثانوي \*

0	0	0	0	X
0	0	0	X	0
0	0	X	0	0
0	X	0	0	0
X	0	0	0	0

### Algorithme Matrice\_Diag\_Sec

```

Const    N=20
Var      A : tableau [1..N,1..N] d'entiers
          i,j : entier
          SDiag : booléen

Début
  pour i ← 1 à N faire
    pour j ← 1 à N faire
      Lire(A[i,j])
    Finpour
  Finpour
  SDiag ← vrai

  i ← 1
  TQ (i ≤ N) et (SDiag) faire
    j ← 1
    TQ (j ≤ N) et (SDiag) faire
      Si (A[i,j] ≠ 0) et (j ≠ N-i+1) alors
        SDiag ← faux
      Fsi
      j ← j+1
    FinTQ
    i ← i+1
  FinTQ

  Si (SDiag) alors
    Ecrire (" la matrice est diagonale")
  Sinon
    Ecrire (" la matrice n'est pas diagonale")
  FinSi
Fin.
```