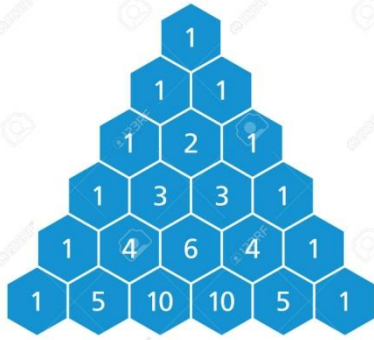


## Triangle de Pascal



\* استعمال مصفوفة وقاعدة المثلث في السطر الأخير \*

1	0	0	0	0	0
1	1	0	0	0	0
1	2	1	0	0	0
1	3	3	1	0	0
1	4	6	4	1	0
1	5	10	10	5	1

### Algorithme PASCAL

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

Début

pour i ← 1 à N faire

P[1,i] ← 0

P[i,1] ← 1

FinPour

pour i ← 2 à N faire

pour j ← 2 à N faire

P[i,j] ← P[i-1,j-1]+P[i-1,j]

Finpour

Finpour

pour i ← 1 à N faire

pour j ← 1 à i faire

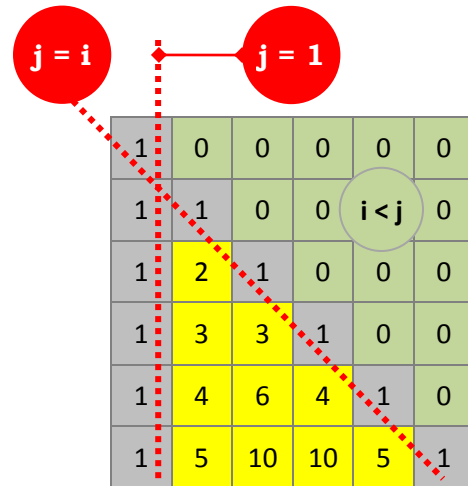
écrire(P[i,j])

Finpour

Finpour

Fin

\* طريقة أخرى \*



### Algorithme PASCAL

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

Début

pour i ← 1 à N faire

pour j ← 1 à N faire

Si (i < j) alors

P[i,j] ← 0

Sinon

Si (i = j) ou (j = 1) alors

P[i,j] ← 1

Sinon

P[i,j] ← P[i-1,j-1]+P[i-1,j]

FinSi

FinSi

Finpour

Finpour

pour i ← 1 à N faire

pour j ← 1 à i faire

écrire(P[i,j])

Finpour

Finpour

Fin

\* استعمال مصفوفة و قاعدة المثلث في العمود الأخير \*

0	0	0	0	0	1
0	0	0	0	1	5
0	0	0	1	4	10
0	0	1	3	6	10
0	1	2	3	4	5
1	1	1	1	1	1

Algorithme **PASCAL**

Const N=20

Var P : tableau [1..N,1..N] d'entiers

i,j : entier

Début

pour i ← N à 1 faire

P[i,1] ← 0

P[N,i] ← 1

FinPour

pour i ← N-1 à 1 faire

pour j ← 2 à N faire

**P[i,j] ← P[i,j-1]+P[i+1,j-1]**

Finpour

Finpour

pour i ← 1 à N faire

pour j ← 1 à i faire

écrire(P[i,j])

Finpour

Finpour

Fin

\* استعمال مصفوفة و قاعدة المثلث في السطر الأول \*

1	5	10	10	5	1
0	1	4	6	4	1
0	0	1	3	3	1
0	0	0	1	2	1
0	0	0	0	1	1
0	0	0	0	0	1

Algorithme **PASCAL**

Const N=20

Var P : tableau [1..N,1..N] d'entiers

i,j : entier

Début

pour i ← 1 à N faire

P[N,i] ← 0

P[i,N] ← 1

FinPour

pour i ← N-1 à 1 faire

pour j ← 1 à N-1 faire

**P[i,j] ← P[i+1,j]+P[i+1,j+1]**

Finpour

Finpour

pour i ← 1 à N faire

pour j ← 1 à i faire

écrire(P[i,j])

Finpour

Finpour

Fin

\* استعمال مصفوفة و قاعدة المثلث في العمود الأول \*

1	0	0	0	0	0
5	1	0	0	0	0
10	4	1	0	0	0
10	6	3	1	0	0
5	4	3	2	1	0
1	1	1	1	1	1

Algorithme **PASCAL**

Const N=20

Var P : tableau [1..N,1..N] d'entiers

i,j : entier

Début

pour i ← 1 à N faire

P[i,N] ← 0

P[N,i] ← 1

FinPour

pour i ← N-1 à 1 faire

pour j ← N-1 à 1 faire

**P[i,j] ← P[i,j+1]+P[i+1,j+1]**

Finpour

Finpour

pour i ← 1 à N faire

pour j ← 1 à i faire

écrire(P[i,j])

Finpour

Finpour

Fin