

**Exercice 1 :**

- N= 2

Etape	N	P	Q	I	Som	écran
1	2	2	1	/	/	/
6	2	2	1	/	/	2

- N= 3

Etape	N	P	Q	I	Som	écran
1	3	3	2	/	/	/
2	3	3	2	1	0	/
3	3	3	2	2	3	/
3	3	3	2	3	6	/
4	3	6	1	3	6	/
6	3	6	1	3	6	6

- N= 4

Etape	N	P	Q	I	Som	écran
1	4	4	3	/	/	/
2	4	4	3	1	0	/
3	4	4	3	2	4	/
3	4	4	3	3	8	/
3	4	4	3	4	12	/
4	4	12	2	4	12	/
2	4	12	2	1	0	/
3	4	12	2	2	12	/
3	4	12	2	3	24	/
4	4	24	1	3	24	/
6	4	24	1	3	24	24

- **Rôle** : Calculer le factoriel d'un nombre « N »

```

PROGRAM CALCUL ;
VAR
    N, P, Q, I, Som : INTEGER ;
BEGIN
    READ(N) ;
    Q :=N - 1 ;
    P := N + 1 ;
    WHILE ( Q > 1) DO
    BEGIN
        Som :=0 ;
        I :=1 ;
        WHILE (I <= Q) DO
        BEGIN
            Som :=Som+P ;
            I :=I+1;
        END ;
        P :=Som ;
        Q :=Q - 1;
    END ;
    IF (P <= 1) THEN
        WRITE (P - Q)
    ELSE
        WRITE (P) ;
    END.

```

**Exercice 2 :**

**Algorithme exo2**

**Constantes :**  
N=20

**Variables :**  
Tab [1..N] : tableau d'entiers  
I : entier  
T : booléen

**Début**

```

    Pour I allant de 1 à N faire
    |   Lire (Tab [I])
    Fin pour
    I ← 1
    T ← vrai
    Tant que ((I <N) et (T=vrai)) faire
    |   Si (Tab [I] > Tab [I+1]) alors
    |   |   T ← faux
    |   Fin si
    |   I ← I+1
    Fin TQ
    Si (T= vrai) alors
    |   Ecrire ('Tableau trié')
    |   Sinon
    |   Ecrire ('Tableau non trié')
    Fin si

```

**Fin**

**Exercice 3**

```

Algorithme exo3
Constantes :
    N= 4
Variables :
    Mat [1..N, 1..N] : matrice d'entiers
    l, J, max: entier
Début
    Pour l allant de 1 à N faire
        Pour J allant de 1 à N faire
            Lire (Mat[l, J])
        Fin pour
    Fin pour
    Max ← Mat[1, 1]
    Pour l allant de 2 à N faire
        Si (Mat [l, l] > Max) alors
            Max ← Mat [l, l]
        Fin si
    Fin pour
    Ecrire (Max)
Fin.
    
```

```

Algorithme exo3
Constantes :
    N= 4
Variables :
    Mat [1..N, 1..N] : matrice d'entiers
    l, J, max: entier
Début
    Pour l allant de 1 à N faire
        Pour J allant de 1 à N faire
            Lire (Mat[l, J])
        Fin pour
    Fin pour
    Max ← Mat[1, 1]
    Pour l allant de 1 à N faire
        Pour J allant de 1 à N faire
            Si ((Mat[l, J] > Max) et (l=J)) alors
                Max ← Mat[l, J]
            Fin si
        Fin pour
    Fin pour
    Ecrire (Max)
Fin.
    
```