TD : Boucle répétitive for

1. Exemple corrigé :

   On définit la suite $u_0 = 10$ *et* $u_{n+1} = \sqrt{u_n}$.
   Écrire un programme qui calcule et affiche u_n pour une valeur de n rentré à la main.

```
n=int(input('rentrez n'))      # l'utilisateur rentre la valeur den
u=10                           # on affecte à u la valeur de u0
for k in range (1,n+1):
        u= u**(1/2)            # on calcule le prochain terme avec la relation de récurrence
print(u)                       # on affiche un
```

2. Écrire un programme qui demande un entier naturel *n* (non nul), qui calcule et affiche la somme des *n* premiers entiers non nuls. Essayer avec des valeurs de n et comparer la valeur affichée avec la formule du cours.

3. (a) Écrire un programme qui demande un entier naturel *n* (non nul), qui calcule et affiche n! .
   (b) * Écrire un programme qui demande un entier naturel *n* (non nul) et un entier k inférieur ou égal à n, qui calcule et affiche $\binom{n}{k}$.

4. Écrire un programme qui demande un entier naturel n et qui calcule et affiche la somme des n premiers termes de la suite arithmétique de raison 3 et de premier terme 1.

5. Écrire un programme qui demande un entier naturel n et qui calcule et affiche la somme des n premiers termes de la suite géométrique de raison 3 et de premier terme 1.

6. Soit la suite (u_n) définie sur $\mathbb{N}^*$ par $u_1 = 1$ *et* $u_{n+1} = 3u_n + 2$.

   (a) Écrire un programme qui demande la valeur de *n* et calcule et affiche tous les termes de la suite de u_1 à u_n.
   (b) Écrire un programme qui demande la valeur de *n* et calcule et affiche la somme des n premiers termes de cette suite.

7. Soit la suite (u_n) définie sur $\mathbb{N}$ par : $u_0 = 3$ et $u_1 = 5$, et $u_{n+2} = u_{n+1} + 2u_n$.
   (a) Écrire un programme qui demande la valeur de $n \geq 2$et qui renvoie la valeur de u_n. Calculer ainsi u_{50}.
   (b) Compléter ce programme pour qu'il affiche en plus la somme des n premiers termes de la suite.

8. Écrire un programme qui calcule la somme des 100 premiers entiers impairs.
9. Écrire un programme qui demande un entier n et calcule les deux quantités suivantes

   $S_n = \sum_{k=1}^{n} \frac{2k+1}{k^2(k+1)^2}$ et $T_n = \frac{n(n+2)}{(n+1)^2}$.

   Lancer le programme pour différentes valeurs de n. Que conjecturer ?

10. Écrire un programme qui calcule les deux quantités suivantes :

    $\prod_{k=1}^{5} \frac{2^{2k}}{3^k}$ et $\prod_{k=1}^{10} \left(1 - \frac{1}{k}\right)$

11. Qu'affichent les programmes suivants ? Justifier avec un calcul.

(a)
```
u=1
for k in range(1,15):
    u=u+2
print(u)
```

(b)
```
u=6
for k in range(1,5):
    u=2*u
print(u)
```

(c)
```
u=1
S=0
for k in range(1,15):
    u=u+2
    S=S+u
print(S)
```

(d)
```
u=6
S=0
for k in range(1,5):
    u=2*u
    S=S+u
print(S)
```

12. Écrire sous forme de somme ce qu'affiche les programmes suivants :

(a)

```python
S=0
for k in range(1,100):
  S=S+k*3
print(S)
```

(b)

```python
S=0
for k in range(1,101):
  S=S+(k-1)/(k+3)
print(S)
```

13. Compléter les programmes suivants pour qu'ils calculent et affichent la valeur de u_n lorsque la suite (u_n) est définie par :

(a) $u_0 = 1 \; et \; u_{n+1} = \sqrt{1+u_n}$.

```python
u=...
for k in range(1,n+1):
  u=...
print(u)
```

(b) $u_0 = 2 \; et \; u_{n+1} = \frac{u_n}{u_n+1}$. Ici, on veut afficher spécifiquement u_{100}.

```python
u=...
for k in range(1,...):
  u=...
print(u)
```

14. Exprimer à l'aide d'une somme double ce qu'affiche le programme suivant, la calculer algébriquement :

```python
n=int(input('rentrer n'))
S=0
for k in range(1,n+1):
  for i in range(k,n+1):
    S=S+k/i
print(S)
```

Corrigé

2.

```
n=int(input('rentrer un entier naturel n non nul'))
S=0
for k in range(0,101):
        S=S+k
print(S)
```

3.

(a)

```
n=int(input('rentrer un entier naturel n non nul'))
f=1             # on met 1 car on va multiplier
for k in range(1,n+1):
        f=f*k
```

(b)

```
n=int(input('rentrer un entier naturel n non nul'))
k=int(input('rentrer un entier k inférieur à n))
S=1
f=1
for i in range(0,k):
        S=S*(n-i)           # cela calcule n!/(n-k)!=n(n-1)...(n-(k-1))
        f=f*(i+1)           # cela calcule k
print(S/f)
```

4.

```
n=int(input('rentrer un entier naturel n non nul'))
S=1
for k in range(0,n):
        S=S+3
print(S)
```

5.

```
n=int(input('rentrer un entier naturel n non nul'))
S=1
for k in range(0,n):
        S=S*3
print(S)
```

6.

(a)

```
n=int(input('rentrer un entier naturel n non nul'))
u=1
print(u)
for k in range(1,n+1):
    u=3*u+2
    print(u)            # on veut afficher tous les termes
```

(b)

```
n=int(input('rentrer un entier naturel n non nul'))
```

```
u=1
S=u
for k in range(1,n):
    u=3*u+2
    S=S+u
print(S)
```

7.

(a)

```
n=int(input('rentrer un entier naturel n non nul'))
u=5
v=3
for k in range(0,n):
    w=u
    u=u+2v
    v=w
print(v)
```

(b)

```
n=int(input('rentrer un entier naturel n non nul'))
u=5
v=3
S=u+v
for k in range(0,n):
    w=u                    # la variable w permet de garder en mémoire u
    u=u+2v
    v=w
    S=S+u
print(S)
```

8.

```
S=0

for k in range(0,100):                  # le premier nombre pair est 0=2*0

    S=S+(2*k+1)
```

9.

Les valeurs sont égales pour différentes valeurs de n, donc on peut conjecturer que Sn=Tn pour tout n.