En algorithmique, on est amené à répéter un certain nombre de fois, la même suite d'opérations. On peut utiliser la boucle **for**. Pour l'utiliser il faut connaître par avance le nombre de fois qu'une instruction doit être effectuée.

Elle prend la forme :

for i **in range**(1,n) :
Instruction 1
Instruction 2

Un exemple de script :

```
for i in range(6):
    print(i)
```

Un autre exemple :

```
n=int(input('n:'))
s=[]
for i in range(1,n+1):
        s=s.append(i)
```

Exercice 1.

1. Entrer dans l'éditeur de Spyder les instructions suivantes :

```
n=int(input(' Donner une valeur de n : ' ))
S=0
for k in range(n):
    S=S+k
print('La somme est :',S)
```

2. Tester pour différents entiers naturels n. A quoi correspond la valeur de S la fin de la boucle ?
3. Modifier les instructions précédentes pour calculer pour un entier naturel n non nul $\sum_{k=1}^{n} k$
4. Proposer un script pour calculer pour un entier naturel n non nul $\sum_{k=1}^{n} k^2$.

Exercice 2.

1. Entrer dans l'éditeur de Spyder les instructions suivantes :

```
n=int(input(' Donner une valeur de n : ' ))
u=0
for k in range(n):
    u=2*u+3
print('La valeur de u est:',u)
```

2. Tester pour différents entiers naturels n.
3. Déterminer le terme général de la suite dont le script précédent calcule les termes.

Exercice 3. Pour chacune des suites définies ici :

$$\left\{\begin{array}{c} u_0 = 6 \\ u_{n+1} = 2u_n - 5 \end{array}\right. \qquad \left\{\begin{array}{c} u_0 = 15 \\ u_{n+1} = \dfrac{u_n^2+2}{2u_n} \end{array}\right. \qquad \left\{\begin{array}{c} u_0 = 1 \\ u_1 = -1 \\ u_{n+2} = -4u_n + 3u_{n+1} \end{array}\right.$$

1. Écrire un script calculant et affichant les termes :
2. Donner u_5 , et vérifier à la main votre résultat.
3. Pour la dernière suite, dire s'il est est possible avec le cours de déterminer le terme général.

Exercice 4.

1. Écrire un script qui demande un entier naturel n, et calcule $n!$ à l'aide d'une boucle **for**.
2. Compléter le script afin d'envoyer un message d'erreur si des valeurs aberrantes de n sont rentrées.

Exercice 5. Créer un script qui calcule $\binom{n}{k}$, pour $n \in \mathbb{N}$ et $0 \leqslant p \leqslant n$.

Exercice 6.

Écrire des programmes calculant dans chaque cas la somme S_n où l'entier n est entré par l'utilisateur.

$$S_n = \sum_{k=0}^{n} k^4 \quad \Bigg| \quad S_n = \sum_{k=1}^{n} \frac{1}{k} \quad \Bigg| \quad S_n = \sum_{k=1}^{n} k^2 \quad \Bigg| \quad S_n = \sum_{k=1}^{n} \frac{1}{2k^6} \quad \Bigg| \quad S_n = \sum_{k=5}^{n-1} \frac{1}{k\sqrt{k}}$$

Avez-vous une idée des éventuelles limites $\lim_{n\to+\infty} S_n$?

Exercice 7. Écrire des programmes calculant dans chaque cas le produit P_n où l'entier n est entré par l'utilisateur.

$$P_n = \prod_{k=1}^{n} \frac{2^{2^k}}{3^k} \quad \Bigg| \quad P_n = \prod_{k=1}^{n} \left(1 + \frac{1}{k^2}\right) \quad \Bigg| \quad P_n = \prod_{k=1}^{n} n^k$$

Vérifier pour quelques valeurs.

Exercice 8.

1. Écrire une suite d'instructions qui, étant donné un entier naturel non nul n, calcule :

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

2. Tester pour différentes valeurs de n et faire une conjecture sur S_n et T_n .
3. Montrer que : $\forall k \in \mathbb{N}^\star$.

$$\frac{1}{k^2} - \frac{1}{(k+1)^2} = \frac{2k+1}{k^2(k+1)^2}$$

4. Prouver la conjecture.

Exercice 9. Soit $n \in \mathbb{N}^\star$, on considère la somme suivante :

$$S_n = \sum_{k=0}^{n} (k^2 + 4k + 4).$$

1. Écrire une suite d'instructions qui, étant donné un entier naturel n, calcule S_n.
2. Calculer algébriquement S_n .