

Quelques programmes en python pour travailler avec les suites

1^{ère} situation

On veut faire un programme servant à calculer les termes d'une suite arithmétique de raison $r = -8$ et de premier terme $u_0 = 24$.

```
1 u=24 #on définit la première valeur de (un)
2 n=0 #on indique le rang associé à cette valeur
3 rang=int(input("n=")) #pour récupérer le rang on pose une question avec input()
4 #et on convertit la réponse obtenue en un entier avec int()
5 while n<rang: #tant que le rang n du terme du moment est encore plus petit
6 #que "rang" le rang qu'on cherche à atteindre
7 u=u-8 #on va calculer le terme suivant
8 n=n+1 #et mettre à jour le rang
9 print("le terme de rang",n," est ",u) #à la fin de la recherche on affichera
10 #le terme obtenu
11 #pour afficher tous les termes calculer il suffit de décaler à droite
12 #la ligne 9
```

Si on demande en plus de donner la somme de tous les termes jusqu'à celui d'un rang donné par l'utilisateur on ajoutera trois lignes au programme :

```
1 u=24 #on définit la première valeur de (un)
2 n=0 #on indique le rang associé à cette valeur
3 somme =u #à ce stade on ne somme qu'un terme de la suite u0
4 rang=int(input("n=")) #pour récupérer le rang on pose une question avec input()
5 #et on convertit la réponse obtenue en un entier avec int()
6 while n<rang: #tant que le rang n du terme du moment est encore plus petit
7 #que "rang" le rang qu'on cherche à atteindre
8 u=u-8 #on va calculer le terme suivant
9 somme=somme+u #la somme est augmentée du nouveau terme de (un)
10 n=n+1 #et mettre à jour le rang
11 print("le terme de rang",n," est ",u) #à la fin de la recherche on
12 #affichera le terme obtenu
13 print("la somme vaut alors ",somme) #on attendra d'avoir affiché tous les
14 #termes avant d'afficher la somme finale
```

Le programme est adaptable à toute suite (arith, géom, quelconque, commençant au rang 0 ou à un autre rang) :

```
1 u=-2 #on définit la première valeur de (un)
2 n=6 #on indique le rang associé à cette valeur
3 somme =u #à ce stade on ne somme qu'un terme de la suite u0
4 print("merci de rentrer un rang valant au moins 6")
5 #pour éviter les erreurs de manipulation
6 rang=int(input("n=")) #pour récupérer le rang on pose une question avec input()
7 #et on convertit la réponse obtenue en un entier avec int()
8 while n<rang: #tant que le rang n du terme du moment est encore plus petit
9 #que "rang" le rang qu'on cherche à atteindre
10 u=3*u+10 #on va calculer le terme suivant
11 somme=somme+u #la somme est augmentée du nouveau terme de (un)
12 n=n+1 #et mettre à jour le rang
13 print("le terme de rang",n," est ",u) #à la fin de la recherche on
14 #affichera le terme obtenu
15 print("la somme vaut alors ",somme) #on attendra d'avoir affiché tous les
16 #termes avant d'afficher la somme finale
```

Là on vient de travailler avec une suite définie de la manière suivante :
$$\begin{cases} u_6 = -2 \\ u_{n+1} = 3u_n + 10 \end{cases}$$

2^{ème} situation

On place 1000€ à la banque et on nous propose deux types de placements :

Formule A : taux fixe de 5%

$$5\% \text{ de } 1000 \text{ c'est } 50 \text{ donc on la caractérisera par } \begin{cases} u_0 = 1000 \\ u_{n+1} = u_n + 50 \end{cases}$$

Formule B : taux fixe de 4%

$$\text{Augmenter une quantité de } 4\% \text{ c'est la multiplier par } \left(1 + \frac{4}{100}\right)$$

$$\text{On la caractérisera par } \begin{cases} v_0 = 1000 \\ v_{n+1} = v_n \left(1 + \frac{4}{100}\right) \end{cases}$$

On se demande quand est ce que la formule B commencera à être plus avantageuse.

```
1 u=1000 #premier terme de (un)
2 v=1000 #premier terme de (vn)
3 n=0 #rang du premier terme
4 while u>=v: #tant que la formule A est plus rentable
5 u=u+50 #on passe au terme suivant pour les deux suites
6 v=v*1.04
7 n=n+1
8 print("apres ",n," années le placement A donne ",u," le placement B donne",v)
9 #petite phrase de conclusion
```

On se demande maintenant quand est ce qu'on dépassera le seuil de 5000€ avec mes deux types de placements.

```
1 u=1000 #définition du premier terme de la suite (un)
2 v=1000 #définition du premier terme de la suite (vn)
3 n=0 #on fixe le rang de ces premiers termes
4
5 while u<5000: #tant que la barre des 5000€ n'est pas dépassée
6 u=u+50 #on va passer d'un terme de (un) au suivant
7 n=n+1 #et bien sur le n augmente à chaque tour
8 print("apres ",n," ans le placement A donne ",u)
9 #on indique pour quel valeur de n le seuil de 5000€ a été dépassé par un terme
10 #de (un)
11 n=0 #remise à zéro
12 print() #on saute une ligne
13 while v<5000: #tant que la barre des 5000€ n'est pas dépassée
14 v=v*1.04 #on va passer d'un terme de (vn) au suivant
15 n=n+1 #et bien sur le n augmente à chaque tour
16 print("apres ",n," ans le placement B donne",v)
17 #on indique pour quel valeur de n le seuil de 5000€ a été dépassé par un terme
18 #de (vn)
```

Les programmes sont disponibles en téléchargement à l'adresse :

<https://www.dimension-k.com/math/1S/Suites-Python.zip>