.

NOM	....	PRENOM	....

Exercices

Première approche de la boucle `while`

On propose d'introduire l'usage de la boucle `while` à l'aide de deux exemples :

1. Programmer la fonction `facto` qui prend n en entrée et renvoie la valeur de $n! = n \times (n-1) \times \cdots \times 2 \times 1$:

```
def facto(n) :
        f=1
        while n>0:
                f=f*n
                n=n-1
        return(f)
```

tester et vérifier quelques valeurs :

```
>>> facto(5)
-> 120
>>> facto(10)
-> 3 628 800
>>> facto(20)
-> 2432902008176640000
```

2. La suite géométrique de premier terme $u_0 = 125$ et de raison $q = 0,95$ est décroissante et ses termes semblent se rapprocher de 0 à mesure que n grandit.

On se demande pour quelles valeurs de l'entier n on a $u_n < 10^{-9}$ (un milliadième). Compléter le script suivant afin que son exécution renvoie un tel entier n :

```
u=125
n=0
while u>= 10**(-9):
        u=u*0.95
        n=n+1
print(n)
```

Indiquez alors la valeur de retour obtenue :

```
|
-> 499
```

Démontrer que, si $p > n$ est un entier supérieur à celui obtenu par le programme précédant, alors $0 < u_p < 10^{-9}$

Comme $(u_n)_{n\in\mathbb{N}}$ est géométrique de raison $q = 0.95 \in]0;1[$ avec $u_0 = 125 > 0$ on a, par propriété, que $(u_n)_{n\in\mathbb{N}}$ est décroissante et positive ? Comme $u_n = u_{499} < 10^{-9}$ on peut ainsi écrire :

$$p \geq n \implies u_p \leq u_n = 10^{-9}$$

et ce, pour n'importe quel entier p, d'où si $p > n$ alors alors $0 < u_p < 10^{-9}$

Tant que vous pouvez encore

A partir de l'exemple de l'aide-mémoire (et de Python) résoudre les problèmes suivants (on fournira script, réponse et justifications) :

1. *Howard Philip J. Frye a été cryogénisé par accident le soir du réveillon de l'an 2000. L'appareil qui le retient prisionner ne le libérera que dans 1000 années. Au moment de l'incident, son compte bancaire disposait de 5 centimes et le taux de rémunération était de 2,75%.*

   Les intérêts étant calculés chaque 1er Janvier et le montant du capital étant arrondi au cent supérieur de façon systématique, en quelle année notre héros deviendra-t-il milliardaire (même si congelé) ?

```
frye = 5                # on raisonne en cents
annee = 2000
while frye <= 10**11:
        frye = frye*1.0275
        frye = floor(frye)+1    #arrondi au cent supérieur
        annee=annee +1
frye = frye/100                         #si on veut lire le compte de frye en dollar
print("en l'an :",annee)
```

2. *Deux frères jumeaux Ki et Hi récupèrent 10 000 euros chacun d'une succession. Le premier, Ki, décide de déposer son héritage sur un compte rémunéré 3% (composés) annuel sans condition. Le second, Hi, préfère faire travailler cet argent par leviers financiers (risqués) et en tire chaque année 400 euros (constants fixes).*

   Etablir le nombre d'années qui seraient nécessaires à Ki pour montrer à Hi que son investissement finira par lui conférer un capital supérieur.

```
Ki=10000
Hi=10000
n=0
while Ki<=Hi
        Ki=Ki*1.03
        Hi=Hi+400
        n=n+1
print("au bout de ",n)
```

3. La suite H_n définie par $H_n = \sum_{k=1}^{n} \frac{1}{k}$ dite *harmonique* semble croître très lentement : mettez ce phénomène en évidence puis déterminer le rang du premier terme à partir duquel H_n dépasse 10 (soyez patient).
   Pensez-vous que H_n puisse dépasser 2025 pour n assez grand ?

```
H=1
k=1
while H<=10:
        k=k+1
        H=H+1/k
print("nombre étapes : ", k)

... eh oui, cette suite (on reconnait l'harmonique) finira par dépasser 2025.
... mais avec les ordinateurs du lycée il vous faudra peut-être 2025 années pour cela !
```

4. Dans la série animée *Futurama*, une formule de calcul (*sauvage*) apparaît : $\sum_{n=0}^{+\infty} 2^n \frac{M_0}{2^n(n+1)}$.

(a) Frye ne comprend pas très bien les mathémtiques et lit 8 au lieu de ∞. Sachant qu'on lui a expliqué que $M_0 = 0,6$, quelle valeur va-t-il trouver ?

```
... on a encodé la suite harmonique en TP for-sum (n°6)
... on reprend et on calcule 0.6*sumH(8) :
-> 1.697380952380952
```

(b) Simplifier la formule de calcul mathématiquement dans le cas $M_0 = 1$. Que reconnaît-on ?

$$M_0 = 1 \implies \sum_{n=0}^{+\infty} 2^n \frac{M_0}{2^n(n+1)} = \sum_{n=0}^{+\infty} \frac{M_0}{n+1} \times \frac{2^n}{2^n} = \sum_{n=0}^{+\infty} \frac{M_0}{n+1} = \sum_{n=0}^{+\infty} \frac{1}{n+1} = H_\infty$$

C'est la série harmonique divergente ! Qui d'ailleurs, dans la série, provoque l'effroi de tous... sauf Frye Les notations employées ci-dessus sont à l'anglaise... les français seraient encore plus effrayés en lisant ces lignes que les personnages de la série...

```
... on reconnait la suite harmonique !
... encore une fois, vue en TP for-sum (n°6) et notée Hn
...
```

(c) Programmer une fonction Python qui prend M_0 un `float` et n un `int` et renvoie la valeur de $M_n = \sum_{k=0}^{n} 2^n \frac{M_0}{2^n(n+1)}$.

```
def futuraM(M0,n):
    M=M0
    for k in range(1,n+1):
        M=M+M0/(k+1)
    return(M)
```

Effectuer quelques tests de valeurs que vous reporterez. *tableau de valeurs obtenus avec* $[0.1, 0.4, 1, 2, 3.3]$ *pour valeurs de lignes (les* M_0*) et* $[10, 100, 1000, 10000, 100000]$ *pour valeurs de colonnes (les* n*)* :

```
[[ 0.30198773  0.51972785  0.74864699  0.9787706   1.20901561]
 [ 1.20795094  2.0789114   2.99458794  3.91508241  4.83606245]
 [ 3.01987734  5.19727851  7.48646986  9.78770603 12.09015613]
 [ 6.03975469 10.39455702 14.97293972 19.57541205 24.18031226]
 [ 9.96559524 17.15101908 24.70535054 32.29942989 39.89751523]]
```

(d) Pour $M_0 = 0,6$, à partir de quelle valeur de $n \in \mathbb{N}$ a-t-on $M_n \geq 10$?
facile : reprendre le while de H_n*, démarrer à 0.6 et adapater avec le contexte*

```
M=0.6
n=1
while M<=10:
        n=n+1
        M=M+0.6/n
print("valeur de n : ", n)
```