

Corrigé

Code de partage avec Capytale : a01b-2328751

Boucle *while* ou « tant que »

Exercice 1 - dépassement d'un seuil, poids de l'ourson

A la naissance, un ourson d'un an pèse 15,7 kg. Chaque année, son poids augmente de 50%, auquel s'ajoutent 3,8 kg. Ecrire un programme qui permet de déterminer en quelle année l'ours aura atteint ou dépassé 270 kg.

On fait grossir l'ourson « tant que » le poids visé n'est pas atteint. Il s'agit en fait d'une suite arithmético-géométrique (dont on pourrait trouver une formule explicite).

La variable `age` n'est pas indispensable pour le calcul mais pour la réponse, elle sert simplement de compteur pour les années.

```
poids=15.7
age=1
while poids<270 :
    age=age+1
    poids=poids*1.5+3.8

print('A', age, "ans l'ourson pèse",
      poids, 'kg')
```

Exercice 2 - dépassement d'un seuil, propagation d'un virus

Un virus se propage dans une population de la manière suivante :

Au jour 0, une personne est nouvellement contaminée et tous les jours on enregistre 1,6 fois plus de nouveaux cas que le jour précédent.

On suppose qu'une même personne ne peut être contaminée deux fois.

Au bout de combien de jours le million de cas (en cumulé) sera dépassé ?

```
cas=1
cumul=1
jour=0
while cumul<1e6 : # la condition porte
    sur le cumul
    jour=jour+1 # on passe au jour
    suivant
    cas=cas*1.6 # évolution du nombre
    de cas quotidiens
    cumul=cumul+cas # on fait évoluer
    le cumul
    print(cas) # si on veut voir l'
    évolution

print(" Le jour", jour, ", il y a",
      round(cas), "cas et le cumul
      atteint", round(cumul))
```

Il faut distinguer ici les termes de la suite : $u_0 = 0$, puis $u_{n+1} = 1,6u_n$ du cumul. Le nombre de cas total au jour n vaut $\sum_{k=0}^n u_k$

Nous pouvons résoudre ce problème par le calcul, mais ici la boucle `while` est adapté à ce type de problème (seuil) : tant que le seuil ne sera pas atteint ou dépassé, on calculera les valeurs au jour suivant.

On peut donc proposer le programme ci-contre.

La variable `jour` sert uniquement de compteur, mais elle n'est pas utilisée pour le calcul de la suite ou du cumul.

On trouve finalement que le million est dépassé au 28^{ème} jour.

Listes

Exercice 3 - faire le tri

1. a) Que fait le programme suivant ?

```
def minimum(L) :
    min=L[0]
    indice=0
    for i in range(1,len(L)):
        if L[i]<min :
            min=L[i]
            indice=i
    return [indice, min]
```

Ce programme définit une fonction qui prend en objet une liste et renvoie sous la forme d'une liste de deux éléments : le premier indice de la liste pour laquelle le minimum est atteint (ne pas oublier que la liste commence à l'indice 0) et la valeur du minimum. On peut la tester dans la console avec `minimum([1,-5,11])` et on obtiendra `[1,-5]`

- b) Combien d'opérations ou tests effectués ce programme ?

Pour chaque valeur de la liste, le programme effectue un test et selon le résultat aucune ou deux affectations. Donc si n est la taille de la liste, le programme effectue en n et $3n$ opérations (une affectation ou un test étant comptés comme opération).

2. a) A l'aide de la fonction `minimum`, écrire un programme qui échange la première valeur d'une liste (numérique) avec le minimum.

Grâce à la fonction `minimum` définie ci-dessus, on obtient le minimum d'une liste et un indice pour lequel il est atteint. Dès lors il suffit de faire l'échange entre la première valeur et celle-ci. Attention on a recours à une variable auxiliaire pour ne pas écraser les valeurs initiales. On définit donc une fonction `echange` qui prend en objet une liste et renvoie la même liste, à l'exception des deux valeurs échangées. Dans un premier temps, on récupère les informations à l'aide de la fonction `minimum` puis on effectue l'échange grâce à la variable auxiliaire `m`

```
def echange(L) :
    indice_min=minimum(L)[0]
    m=L[indice_min]
    L[indice_min]=L[0]
    L[0]=m
    return L
```

- b) A l'aide de la fonction `minimum`, écrire un programme qui trie les valeurs d'une liste (numérique) par ordre croissant.
- c) Combien d'opérations ou tests effectués ce dernier programme ?