
Exercices Corrigés (série 9)

Exercice 1

Un nombre heureux est un nombre entier qui, lorsqu'on ajoute les carrés de chacun de ses chiffres, puis les carrés des chiffres de ce résultat et ainsi de suite jusqu'à l'obtention d'un nombre à un seul chiffre égal à 1 (un).

Exemple :

N=7 est heureux, puisque :

$$7^2=49$$

$$4^2+9^2=97$$

$$9^2+7^2=130$$

$$1^2+3^2+0^2=10$$

$$1^2+0^2=1$$

On est arrivé à un nombre d'un seul chiffre qui est égal à 1, donc N=7 est heureux

Travail demandé :

Ecrire une fonction heureux(nb) qui permet de déterminer si un nombre entier nb est heureux ou non.

⇒ **Correction**

```
def heureux(nb):
```

```
    etat=False # etat du nombre est ce que c'est heureux ou non par défaut le
```

```
nombre n'est pas heureux
```

```
    limite=False # verifier est ce que le nombre est inférieur à 10
```

```
    nombre=str(nb)
```

```
    while limite==False:
```

```
        s=0
```

```
        for i in nombre:
```

```
            s+=int(i)**2
```

```
        nombre=str(s)
```

```
        if nombre=='1':
```

```
            etat=True
```

```
            break
```

```
        if int(nombre)<10: # si le nombre<10 donc limite=True
```

```
            limite=True
```

```
    return etat
```

Exercice 2

La suite de robinson est définie par :

- $U_0=0$
- U_n se construit en concaténant le nombre d'apparition de chacun des chiffres constituant le terme U_{n-1} suivi du chiffre lui-même, selon l'ordre décroissant des chiffres, pour tout $n>0$.

Exemple :

Pour $n=5$, $U_5=13123110$

En effet :

- $U_0=0$
- $U_1=10$ car il y a une apparition (1) du chiffre 0 dans U_0
- $U_2=1110$ car il y'a une apparition (1) du chiffre 1 et une apparition (1) du chiffre 0 dans U_1
- $U_3=3110$ car il y'a une apparition (3) du chiffre 1 et une apparition (1) du chiffre 0 dans U_2
- $U_4=132110$ car il y'a une apparition (1) du chiffre 3, deux apparition du chiffre 1 et une apparition (1) du chiffre 0 dans U_3
- $U_5=13123110$ car il y'a une apparition (1) du chiffre 3, une apparition du chiffre 2, trois apparitions du chiffre 1 et une apparition (1) du chiffre 0 dans U_4

Travail à faire :

Ecrire une fonction **robinson(n)** qui permet de calculer le $N^{\text{ième}}$ terme de la suite de robinson

⇒ **Correction**

```
def robinson(n):
    U=0
    for i in range(1,n+1):
        ch=str(U)
        L=[0,0,0,0,0,0,0,0,0,0]
        for chiffre in ch:
            L[int(chiffre)]+=1
        res=""
        for j in range(9,-1,-1):
            if L[j]!=0:
                res+=str(L[j])+str(j)
        U=int(res)
        print(U)
    print('Terme : ',n,' est ',U)
```

Exercice 3

Un entier est dit distinct s'il est composé de chiffres distincts (différents).

Ecrire un programme python qui permet de saisir un entier n ($n > 0$), puis de vérifier et d'afficher si cet entier est distinct ou non.

Exemple :

N=1273 est dit distinct car il est formé par les chiffres 1, 2, 7 et 3 qui sont tous distincts, donc, le programme affichera : **cet entier est distinct**

N=1565 est dit non distinct car il est formé par les chiffres 1, 5, 6, 5 qui ne sont pas tous distincts (le chiffre 5 se répète deux fois, donc le programme affichera : **cet entier est non distinct**

⇒ Correction

```
def estdistinct(n):
    etat=True
    if n>0:
        ch=str(n)
        for lettre in ch:
            if ch.count(lettre)>1:
                etat=False
                break
        if etat==True:
            print(n,' est distinct')
        else:
            print(n,' est non distinct')
    else:
        print('le nombre est négatif ')
```