Informatique - TP 8

TD noté - 8 janvier

NB n°1 : vous enregistrerez votre travail dans votre répertoire réseau sous le titre `TD8-VotreNom.py`.
La première ligne de votre code devra faire apparaître un commentaire dans lequel sera écrit votre nom.
NB n°2 : parmi les questions qui suivent, celles pour lesquelles la réponse attendue n'est pas un code Python, seront traitées sous forme de commentaire dans votre code.
NB n°3 : on ne demande pas de vérifier la validité des arguments fournis aux fonctions demandées dans les exercices.

1. Créer une liste `cubes` constituée des ***cubes*** des entiers de l'intervalle $[\![0;29]\!]$.

2. Écrire une fonction `div11(n)` qui prend en paramètre un entier n et ***renvoie*** la ***somme alternée des chiffres de*** $|n|$: ***chiffre des unités-chiffres des dizaines+chiffre des centaines-etc...***
   Par exemple, `div11(-1345)` doit renvoyer $5-4+3-1=3$...

3. Écrire une fonction `myst(n)` qui prend en paramètre un entier n, calcule `div11(n)`, puis `div11(div11(n))`, etc... et renvoie le premier de ces nombres qui ne s'écrit qu'avec un seul chiffre.
   Par exemple, `myst(5182)` doit renvoyer ***1*** car `div11(5182)` vaut $2-8+1-5=-10$, `div11(-10)` vaut -1 et enfin `div11(-1)` vaut 1.

4. Afficher la liste des valeur de `myst(k)` pour tous les entiers `k` de la liste `cube`.
   Vérifier que cette liste comprend trois fois la valeur 0.

5. On définit la suite $(H_n)_{n\in\mathbb{N}}$ par $\begin{cases} H_0 &= 1 \\ H_{n+1} &= \dfrac{H_n+\frac{3}{H_n}}{2} \end{cases}$.
   Écrire une fonction `her(n)` qui prend en paramètre un entier n et renvoie la valeur de H_n.

6. Afficher sur une même ligne la valeur de H_{10} et de $\sqrt{3}$.
   Remarque ?

7. Importer `numpy` à l'aide de la ligne de code
   ```
   import numpy as np
   ```
   Vous pouvez désormais calculer l'exponentielle d'un nombre r en écrivant `np.exp(r)`.

8. On définit la suite $(u_n)_{n\in\mathbb{N}}$ par $\begin{cases} u_1 &= 0 \\ u_{n+1} &= \dfrac{e^{-u_n}}{n} \end{cases}$.
   Écrire une fonction `somme(n)` qui étant donné un entier $n>0$ renvoie la valeur de $\displaystyle\sum_{k=1}^{n}(-1)^k u_k$.

9. Afficher `somme(50)` puis `somme(100)`.
   Cette somme semble-t-elle converger lorsque $n\to+\infty$?

10. Écrire une fonction `diamond(n)` qui prend pour argument un ***entier*** `n` et a pour effet d'***afficher*** les $2n+1$ lignes décrites par les exemples suivants :

```
>>> diamond(1)
 *
* *
 *
```

```
>>> diamond(2)
  *
 * *
*   *
 * *
  *
```

```
>>> diamond(3)
   *
  * *
 *   *
*     *
 *   *
  * *
   *
```

11. Faire afficher `diamond(3)` et `diamond(4)`.

Correction

```python
# Q1
cubes = [k**3 for k in range(30)]

# Q2
def div11(n):
    s = 0
    i = 1
    for k in str(abs(n))[-1::-1]:
        k = int(k)
        s = s+ i*k
        i = -i
    return s

# Q3
def myst(n):
    res = str(div11(n))
    while len(res)>1:
        res = str(div11(int(res)))
    return int(res)

# Q4
print([myst(k) for k in cubes])

# Q5
def her(n):
    u = 1
    for i in range(n):
        u = (u+3/u)/2
    return u

# Q6
print(her(10),3**0.5)

# Q7
import numpy as np

# Q8
def somme(n):
    u=0
    i=0
    s=0
    while i<n:
        s = s+(-1)**i*u
        i += 1
        u = np.exp(-u)/i
    return s
```

```python
# Q9
print(somme(50))
print(somme(100))
# La somme semble converger.

# Q10
def diamond(n):
    print(n*' '+'*')
    for k in range(1,n+1):
        print((n-k)*' '+'*'+(2*k-1)*' '+'*')
    for k in range(n-1,0,-1):
        print((n-k)*' '+'*'+(2*k-1)*' '+'*')
    print(n*' '+'*')

# Q11
diamond(3)
diamond(4)
```

Informatique - TP 8

TD noté - 8 janvier

NB n°1 : vous enregistrerez votre travail dans votre répertoire réseau sous le titre `TD8-VotreNom.py`.
La première ligne de votre code devra faire apparaître un commentaire dans lequel sera écrit votre nom.
NB n°2 : parmi les questions qui suivent, celles pour lesquelles la réponse attendue n'est pas un code Python, seront traitées sous forme de commentaire dans votre code.
NB n°3 : on ne demande pas de vérifier la validité des arguments fournis aux fonctions demandées dans les exercices.

1. Créer une liste `carres` constituée des ***carrés*** des entiers de l'intervalle $[\![0;29]\!]$.
2. Écrire une fonction `som(n)` qui prend en paramètre un entier n et ***renvoie*** la ***somme des chiffres de*** $|n|$: ***chiffre des unités+chiffres des dizaines+chiffre des centaines+etc...***
   Par exemple, `div11(-1345)` doit renvoyer $5+4+3+1=13$...
3. Écrire une fonction `myst(n)` qui prend en paramètre un entier n, calcule `som(n)`, puis `som(som(n))`, etc... et renvoie le premier de ces nombres qui ne s'écrit qu'avec un seul chiffre.
   Par exemple, `myst(5182)` doit renvoyer ***7*** car `som(5182)` vaut $2+8+1+5=16$, et `som(16)` vaut 7.
4. Afficher la liste des valeur de `myst(k)` pour tous les entiers `k` de la liste `carres`.
   Vérifier que cette liste est périodique, de période 9, à partir du rang 1.
5. On définit la suite $(u_n)_{n\in\mathbb{N}}$ par $\begin{cases} u_0 &= 1 \\ u_{n+1} &= \dfrac{u_n - \frac{2}{u_n}}{2} \end{cases}$.
   Écrire une fonction `suite(n)` qui prend en paramètre un entier n et renvoie la valeur de u_n.
6. Afficher les valeurs de u_k pour $k \in [\![0;19]\!]$.
   La suite u semble-t-elle converger ?
7. Écrire une fonction `moyenne(n)` qui étant donné un entier n renvoie la valeur de $\dfrac{\sum_{k=0}^{n-1} u_k}{n}$, c'est-à-dire la moyenne des n premiers termes de la suite u.
8. Afficher `moyenne(400)`, `moyenne(500)` et `moyenne(600)`.
   Cette moyenne semble-t-elle converger lorsque $n \to +\infty$?
9. Écrire une fonction `sapin(n)` qui prend pour argument un ***entier*** `n` et a pour effet d'***afficher*** les $2n+1$ lignes décrites par les exemples suivants :

```
>>> sapin(1)
 *
***
 *
```

```
>>> sapin(2)
  *
 ***
 ***
*****
  *
```

```
>>> sapin(3)
   *
  ***
  ***
 *****
 *****
*******
   *
```

10. Faire afficher `sapin(3)` et `sapin(8)`.

Correction

```python
# Q1
carres = [k**2 for k in range(30)]

# Q2
def som(n):
    s = 0
    for k in str(abs(n)):
        k = int(k)
        s = s + k
    return s

# Q3
def myst(n):
    res = str(som(n))
    while len(res)>1:
        res = str(som(int(res)))
    return int(res)

# Q4
print([myst(k) for k in carres ])

# Q5
def suite(n):
    u = 1
    for i in range(n):
        u = (u-2/u)/2
    return u

# Q6
print([suite(k) for k in range(20)])
# La suite ne semble pas converger.

# Q7
def moyenne(n):
    i=0
    s=0
    while i<n:
        s = s+suite(i)
        i += 1
    return s/n

# Q8
print(moyenne(400))
print(moyenne(500))
print(moyenne(600))
# La moyenne des termes de la suite ne semble pas non plus converger.
```

```python
# Q9
def sapin(n):
    print(n*' '+'*')
    for k in range(1,n):
        print((n-k)*' '+(2*k+1)*'*')
        print((n-k)*' '+(2*k+1)*'*')
    print((2*n+1)*'*')
    print(n*' '+'*')

# Q10
sapin(3)
sapin(8)
```