Correction TP2 python

Exercice 1

1)liste_1 est formée des termes 2^k pour k allant successivement de 1 à 5.
Donc liste_1=[2,4,8,16,32].

2)liste_2 est formée des entiers pairs k allant successivement de 0 à 19.
Donc liste_1=[0,2,4,6,8,10,12,14,16,18].

3)liste_3 est formée des entiers de la forme ij où i est un entier entre 1 et 3 et j un entier entre 1 et 4.
Une liste étant un ensemble ordonné, il faut être vigilant sur l'ordre des éléments qu'on écrit.
On prend $i = 1$, puis $j = 1, 2, 3, 4$. Ensuite, on prend $i = 2$, puis $j = 1, 2, 3, 4$. Etc...
On obtient : liste_3 =[1, 2, 3, 4, 2, 4, 6, 8, 3, 6, 9, 12].

Exercice 2

1)La liste est formée de tous les entiers impairs de 1 à 143.
liste=[k for k in range(1,143) if k%2==1]
ou bien liste=[2*k+1 for k in range(72)].

2)La liste est formée de tous les carrés parfaits entre 1 et 100.
liste=[k**2 for k in range(1,11)]

3)liste=[(k+1)/k for k in range(1,100)]

Exercice 3

1)A est la matrice ligne constituée des termes consécutifs de la suite arithmétique de premier terme 0 et de raison 1/4 en s'arrêtant à 2 sans l'atteindre.
Donc A=([0,0.25,0.5,0.75,1,1.25,1.5,1.75]).

2)B est la matrice ligne constituée des 6 termes consécutifs de la suite arithmétique de premier terme 2 et de dernier terme 10.
Cherchons la raison r de cette suite.
En numérotant ces termes de u_1 à u_6, on a : $u_6 = u_1 + (6 - 1) \times r$, soit : $10 = 2 + 5r$, ce qui mène à $r = 8/5 = 1.6$.
Donc B=([2,3.6,5.2,6.8,8.4,10]).

Exercice 4

L'intervalle [-150,200] a pour longueur 350. Chacun des 50 sous-intervalles a donc pour longueur 350/50=7.
Il suffit donc de construire une suite arithmétique de premier terme -150 et de raison 7, d'où le programme :

```
import numpy as np
subdivision=np.arange(-150,201,7)
```

Exercice 5

1)programme :

```
#input renvoie une chaîne de caractères qu'on convertit en entier
n=int(input("entrer n"))
#construction d'une liste en compréhension
liste=[1/k for k in range(1,n+1)]
#affichage de la somme
print(sum(liste))
```

2)On complète le programme avec :

```
import numpy as np
print(sum(liste)-np.log(n))
```

Pour n très grand, on trouve une limite proche de 0,577.

✓ La limite, notée γ et appelée constante d'Euler vaut $0,5772156649...$

Exercice 6

1)Pour tout entier $k \in [\![1, n]\!]$, on ajoute à la liste L l'élément kL[k-1].
Ainsi, $L[0] = 1$ et $\forall k \in [\![1, n]\!], L[k] = kL[k-1]$.
Une récurrence immédiate montre que $\forall k \in [\![0\, n]\!], L[k] = k!$.
La fonction renvoie donc $n!$

2)On complète le programme :

```
n=int(input(''entrer n''))
liste=[1/f(k) for k in range(0,n+1)]
print(sum(liste))
```

Pour $n = 100$, on obtient : $\sum_{k=0}^{100} \frac{1}{k!} \approx 2.7182$.

On retrouve le fait que $\lim_{n\to+\infty} \sum_{k=0}^{n} \frac{1}{k!} = e$ (série exponentielle).

Exercice 7

2)programme :

```
L=[0,1,2,3,4]
rd.shuffle(L)
print(L)
compteur=0
for k in range(5):
    if L[k]!=k:
        compteur+=1
print(compteur)
```

Exercice 8

1)La fonction chercher renvoie True si x est dans la liste L et False sinon. Cette fonction peut s'écrire plus simplement :

```
def chercher(x,L):
    if x in L:
        return True
    else:
        return False
```

2)a)En écrivant les listes en compréhension, on obtient :
L1=[k**3-10 for k in range(21)]
La commande print(L1) donne : L1=[-10, -9, -2, 17, 54, 115, 206, 333, 502, 719, 990, 1321, 1718, 2187, 2734, 3365, 4086, 4903, 5822, 6849, 7990].

Puis, L2=[8*k**2+19*k for k in range(21)]
La commande print(L2) donne : [0, 27, 70, 129, 204, 295, 402, 525, 664, 819, 990, 1177, 1380, 1599, 1834, 2085, 2352, 2635, 2934, 3249, 3580].

2)b)programme :

```
#Fonction de recherche d'un élément dans une liste
def chercher(x,L):
    for k in range(len(L)):
        if L[k]==x:
            return True
    return False
#Construction des listes L1 et L2 par compréhension
L1=[k**3-10 for k in range(0,21)]
L2=[8*k**2+19*k for k in range(0,21)]
#Construction de la liste L
L=[ ]
for i in range(len(L1)):
    if chercher(L1[i],L2)==True:
        L.append(L1[i])
print(L)
```

On trouve L=[990].

3)Soit $k \in [\![0, 20]\!]$ une solution de l'équation $x^3 - 8x^2 - 19x - 10 = 0$.
On a alors : $k^3 - 8k^2 - 19k - 10 = 0$, puis $k^3 - 10 = 8k^2 + 19k$, c'est-à-dire $L1[k] = L2[k]$.
Comme les listes L1 et L2 n'ont que la valeur 990 en commun, cela entraîne que L1[k]=990.
La commande print(L1.index(990)) renvoie alors k=10.
On vérifie réciproquement que 10 est bien solution de l'équation.

Exercice 9

programme :

```
import numpy.random as rd
#Fonction de recherche d'un élément dans une liste
def chercher(x,L):
    for k in range(len(L)):
        if L[k]==x:
            return True
    return False
#Création et affichage d'une liste alea de 10 chiffres aléatoires
alea=[rd.randint(0,10) for k in range(10)]
print(alea)
#Création de la liste sans_doublons
sans_doublons=[alea[0]]
for k in range(1,len(alea)):
    if chercher(alea[k],sans_doublons)==False:
        sans_doublons.append(alea[k])
#Tri et affichage de la liste sans_doublons
sans_doublons.sort()
print(sans_doublons)
```