INTERROGATION DE PYTHON N°1

EXERCICE 1 **10 points**

On considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par $\begin{cases} u_{n+1} = 1 - e^{-u_n} \\ u_0 = 3 \end{cases}$.

1. Compléter la fonction suivante pourqu'elle revoie la valeur de $f(x) = 1 - e^{-x}$ lorsque l'utilisateur entre la valeur du réel strictement positif x.

```
import ............ as ............

def f(x):
  return ............................
```

2. On suppose la fonction précédente correctement. Expliquer **très précisément** le rôle de la fonction suivante.

```
def suite_u(n):
    u=1
    for k in range (1,n):
        u=f(u)
    return u
```

3. Modifier la fonction précédente pour qu'elle renvoie la liste des $n+1$ premiers termes de la suite $(u_n)_{n\in\mathbb{N}}$ lorsque l'utilisateur entre la valeur de n.

```
def suite_liste_u(n):
  ..............
  ..............
  ..............
    ..............
    ..............
  return .......
```

4. On suppose la fonction précédente correctement écrite. Proposer des commandes permettant d'afficher le graphique suivant représentant les 100 premiers termes de la suite $(u_n)_{n\in\mathbb{N}}$:

```
import .............. as ..............

L = ..............
plt.plot(...... , .......)
plt.sow()
```

5. Quelles conjectures peut-on émettre sur la suite $(u_n)_{n\in\mathbb{N}}$?

INTERROGATION PYTHON N°1 : CORRECTION

EXERCICE 1 **10 points**

On considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par $\begin{cases} u_{n+1} = 1 - e^{-u_n} \\ u_0 = 3 \end{cases}$.

```python
import numpy as np

def f(x):
  return 1-np.exp(-x)
```

```python
def suite_u(n):
    u=1
    for k in range (1,n):
        u=f(u)
    return u
```

Rôle de la fonction : étant donné l'entier n, cette fonction calcule et affiche le terme u_{n-1}.
Commentaires : dans la boucle `for k in range(1,n)` la variable `k` parcourt l'intervalle $[\![1;n-1]\!]$ contenant $n-1$ éléments. Donc le dernier terme calculé est bien u_{n-1} et non u_n.

```python
def suite_liste_u(n):
  u=1
  L=[u]
  for k in range(n) :
    u=f(u)
    L.append(u)
  return L
```

Commentaires :

- on ajoute la variable `L` de type liste que l'on initialise à `[`u_0`]`
- dans la boucle `for k in range(n)` la variable `k` parcourt l'intervalle $[\![0;n-1]\!]$ contenant n éléments, par conséquent le dernier terme calculé est u_n
- la commande `L.append(u)` permet d'ajouter au fur et à mesure les termes à la liste `L`, si bien qu'à la fin `L` contient les termes $u_0, \ldots, u_n$ représentant bien les $n+1$ premiers termes de la suite.
- on renvoie `L` par la commande `return L`.

4.

```python
import matlplotlib.pyplot as plt

L =suite_liste_u(99)
plt.plot(L ,'+' )
plt.sow()
```

Commentaires : pour avoir les 100 premiers termes de la suite il faut entre $n = 99$ dans la fonction précédente car les 100 premiers termes sont les termes $u_0, \ldots, u_{99}$.

5. On peut aisément conjecturer que la suite $(u_n)_{n\in\mathbb{N}}$ est :

- bornée par 0 et 1
- strictement décroissante
- tend vers 0.