EXERCICE 5 (4 points)

Thème : Exécution de programmes, recherche et corrections de bugs

Les questions proposées sont indépendantes les unes des autres.

1. On considère la fonction `somme(n)` qui reçoit en paramètre un entier `n` strictement positif et renvoie le résultat du calcul $1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}$.

```
1|    def somme(n) :
2|        total = 0
3|        for i in range(n) :
4|            total = total  + 1/i
5|        return total
```

Lors de l'exécution de `somme(10)`, le message d'erreur `"ZeroDivisionError: division by zero"` apparait. Identifier le problème et corriger la fonction pour qu'elle effectue le calcul demandé.

2. On considère la fonction `maxi(L)` qui prend comme paramètre une liste `L` de nombres et renvoie le plus grand nombre de cette liste :

```
1|    def maxi(L) :
2|        indice = 0
3|        maximum = 0
4|        while indice <= len(L) :
5|            if L[indice] > maximum :
6|                maximum = L[indice]
7|            indice = indice + 1
8|        return maximum
```

a. Lors de l'exécution de `maxi([2, 4, 9, 1])` une erreur est déclenchée. Identifier et corriger le problème.

b. Le bug précédent est maintenant corrigé. Que renvoie à présent l'exécution de `maxi([-2, -7, -3])` ? Modifier la fonction pour qu'elle renvoie le bon résultat.

3. On souhaite réaliser une fonction qui génère une liste de `n` joueurs identifiés par leur numéro. Par exemple on souhaite que l'appel `genere(3)` renvoie la liste `['Joueur 1', 'Joueur 2', 'Joueur 3']`.

```
1|   def genere(n) :
2|       L = []
3|       for i in range(1, n+1) :
4|           L.append('Joueur '+i)
5|       return L
```

L'appel `genere(3)` déclenche l'erreur suivante : **`TypeError: can only concatenate str (not "int") to str`**.

Expliquer ce message d'erreur et corriger la fonction afin de régler le problème.

4. On considère la fonction `suite(n)` qui reçoit un entier positif et renvoie un entier.

```
1|   def suite(n) :
2|       if n == 0 :
3|           return 0
4|       else :
5|           return 3+2*suite(n-2)
```

a. Quelle valeur renvoie l'appel de `suite(6)` ?

b. Que se passe-t-il si on exécute `suite(7)` ?

5. On considère le code Python ci-dessous :

```
1|   x = 4
2|   L = []
3|   def modif(x, L) :
4|       x = x + 1
5|       L.append(2*x)
6|       return x, L
7|
8|   print(modif(x, L))
9|   print(x, L)
```

a. Qu'affiche le premier `print` ?

b. Qu'affiche le second `print` ?