> D'après 2023, sujet zéro A
> Auteurs : Frédéric Junier, Franck Chambon, Denis Quenton

Exécution de programmes, recherche et corrections de bugs

> D'après 2022, Asie, J2, Ex. 5

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

```
def somme(n) :
    total = 0
    for i in range(n):
        total = total + 1 / i
    return total
```

   Lors de l'exécution de `somme(10)` , le message d'erreur `#!py 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` qui prend comme paramètre une liste `L` **non vide** de nombres et renvoie le plus grand nombre de cette liste :

```
def maxi(L) :
    indice = 0
    maximum = 0
    while indice <= len(L):
        if L[indice] > maximum :
            maximum = L[indice]
        indice = indice + 1
    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']` .

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

L'appel `genere(3)` déclenche l'erreur suivante

```
#!py 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` qui prend un entier positif `n` en paramètre et renvoie un entier.

```
def suite(n) :
    if n == 0:
        return 0
    else :
        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 :

```
x = 4
L = []
def modif(x, L):
    x = x + 1
    L.append(2 * x)
    return x, L

print(modif(x, L))
print(x, L)
```

a. Qu'affiche le premier `print` ?

b. Qu'affiche le second `print` ?