

_____ Le sujet comporte 6 exercices et 9 pages, dont une page d'annexe _____
Aucun document n'est autorisé – Toutes les fonctions de manipulation de nombres, de listes, et d'images disponibles sont rappelées en annexe page 9. Vous pouvez détacher cette annexe pour plus de facilité.

Écrivez votre nom et prénom sur chaque feuille pour éviter qu'elles se perdent.

Exercice 1 Considérons la fonction `mystere` suivante :

```

1 def mystere(s : str) -> str :
2     char = s[0]
3     resultat = char
4     for i in range(1, len(s)):
5         if s[i] != char:
6             char = s[i]
7             resultat = resultat + char
8     return resultat

```

1. Quel est le type du paramètre `s` de cette fonction et de la valeur qu'elle retourne ?

2. Simulez l'exécution de l'appel `mystere("laaalaaa")` en complétant le tableau suivant qui montre l'évolution des variables `resultat`, `char` et `i`.

Vous devez remplir le tableau de telle sorte qu'une seule valeur apparaît par colonne.

<code>char</code>		
<code>resultat</code>		
<code>i</code>		

3. Que retourne l'appel `mystere("laaalaaa")` ?

4. Que retourne l'appel `mystere("ouuups")` ?

5. L'appel `mystere("")` provoque une erreur à l'exécution.

Expliquez l'erreur puis modifiez le code de la fonction pour que l'appel `mystere("")` retourne `""`. Écrivez seulement le code à ajouter à la fonction ci-dessus et indiquez précisément où il doit être inséré.

6. De façon générale, expliquer de manière synthétique en français, comment interpréter la valeur retournée par la fonction `mystere(s)`, pour une chaîne de caractères `s` non vide ?

7. Quelle est la complexité de la fonction `mystere(s)`, en fonction de `n`, la longueur de la chaîne de caractères `s` ?

(ne cocher qu'une seule case)

$\mathcal{O}(1)$ $\mathcal{O}(\log(n))$ $\mathcal{O}(\sqrt{n})$ $\mathcal{O}(n)$ $\mathcal{O}(n^2)$ $\mathcal{O}(n^3)$ $\mathcal{O}(2^n)$

Exercice 2 Expressions booléennes

1. Soit le code suivant :

```
x = 1
y = 2
p = x >= 1 and y < 10
q = p and x == y
r = not (p or q)
```

Donnez les valeurs de `p`, `q`, et `r`

2. Indiquer l'ensemble des valeurs entières de `x` pour lesquelles l'expression Python suivante vaut `True`.

```
x >= 5 and x <= 12 and (x % 4 == 0 or x == 7)
```

Nom:

Prénom:

Groupe:

3. Pour rappel, une année bissextile est une année comportant 366 jours au lieu de 365 pour une année normale (le jour ajouté est le 29 février).

Mathématiquement, une année est bissextile si elle est divisible par 4, sauf cas particuliers : si elle divisible par 100 et qu'elle n'est pas divisible par 400, l'année n'est pas bissextile. Par exemple, 2000, 2004, 2024, 2400 sont bissextils, mais 2001, 2023, 2100, 2200 ne le sont pas.

Écrire une fonction `bissextile(annee:int)->bool` qui retourne `True` quand l'année `annee` est bissextile, et `False` si elle ne l'est pas.

Exercice 3

On souhaite réaliser une fonction `listeVersEntier(L: list) -> int` qui prend en entrée une liste `L` d'entiers positifs strictement inférieurs à 10, et qui retourne le nombre correspondant à l'interprétation décimale de la liste `L`. *Exemples* : `listeVersEntier ([1,8,7,0])` renvoie l'entier 1870, `listeVersEntier ([4,2,5])` renvoie l'entier 425 .

Écrivez la fonction `listeVersEntier(L: list) -> int`

Exercice 4 Un étudiant écrit la fonction `sommeList` ci-dessous qui prend en paramètres deux listes de nombres et qui renvoie une liste dont la valeur à l'indice i est égale à la somme des valeurs d'indice i des listes `L1` et `L2`.

Par exemple pour les listes `[1, 3, 5]` et `[2, -3, -8]` on obtient la liste `[3, 0, -3]`

exercice 4.1.6 page 30 du poly

```
def sommeList(L1:list, L2:list)->list:
```

```
    fin = len(L1)
    for i in range(fin, -1, -1):
        L[i] = L1[i] + L2
    return L
```

```
sommeList([1, '3, 5], [2, -3, -8])
```

Malheureusement lors de l'appel suivant : `sommeList([1, '3, 5], [2, -3, -8])`, Python Tutor indique 4 erreurs représentées ci-dessous par une capture d'écran. Proposer dans le cadre associé un code permettant de corriger **UNIQUEMENT** l'erreur décrite.

```
1 def sommeList(L1, L2):
2
3     fin = len(L1)
4     for i in range(fin, -1, -1):
5     L[i] = L1[i] + L2
6     return L
7
8 sommeList([1, 3, 5], [2, -3, -8])
IndexError: list index out of range
```

```
1 def sommeList(L1, L2):
2
3     fin = len(L1)
4     for i in range(fin, -1, -1):
5     L[i] = L1[i] + L2
6     return L
7
8 sommeList([1, 3, 5], [2, -3, -8])
NameError: name 'L' is not defined
```

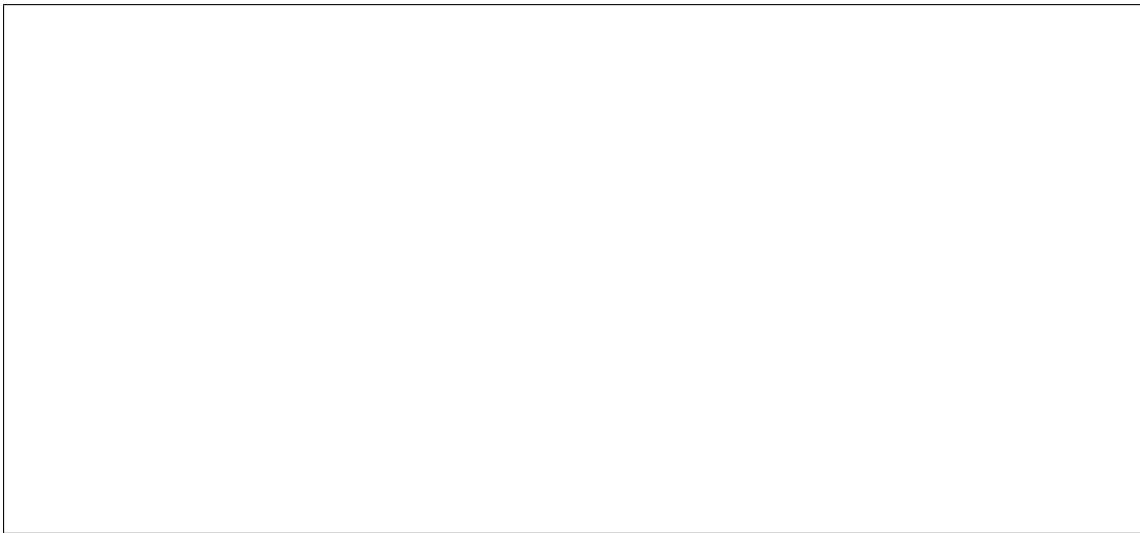
```
1 def sommeList(L1, L2):
2
3     fin = len(L1)
4     for i in range(fin, -1, -1):
5     L[i] = L1[i] + L2
6     return L
7
8 sommeList([1, 3, 5], [2, -3, -8])
TypeError: unsupported operand
type(s) for +: 'int' and 'list'
```

```
1 def sommeList(L1, L2):
2
3     fin = len(L1)
4     for i in range(fin, -1, -1):
5         L[i] = L1[i] + L2
6     return L
7
8 sommeList([1, '3, 5], [2, -3, -8])
File "<string>", line 8
    sommeList([1, '3, 5], [2, -3, -8])
                        ^
SyntaxError: EOL while scanning string literal
```

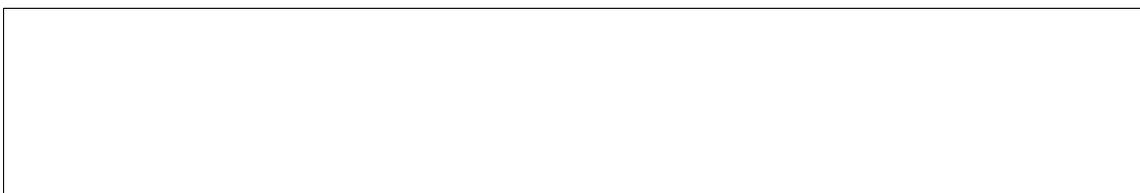
Exercice 5 Chaînes de caractères

1. Extraire le premier mot d'une chaîne de caractères.

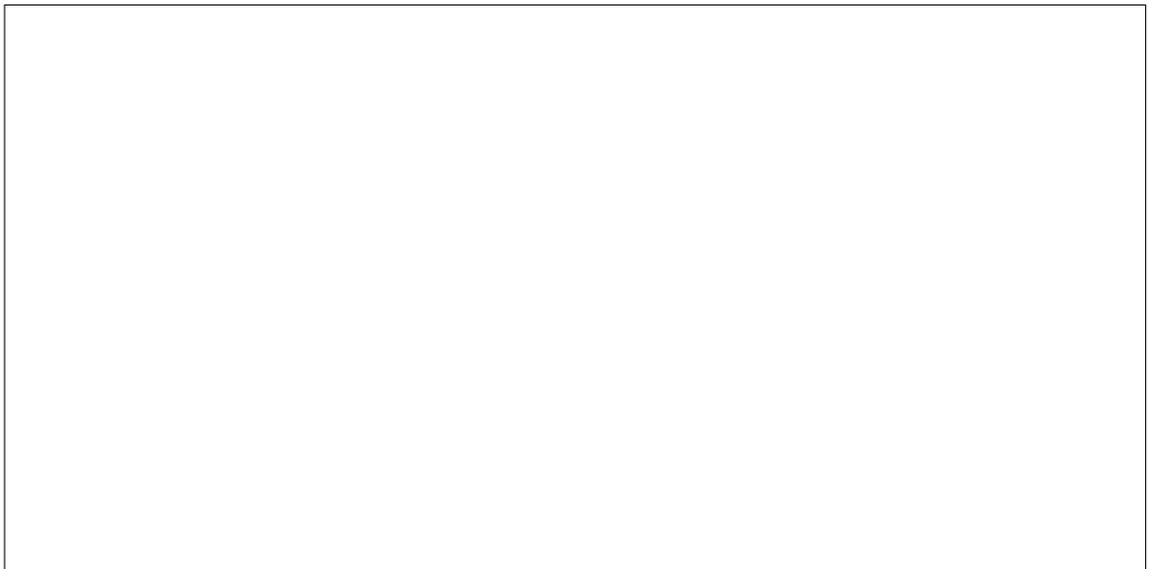
Soit s une chaîne de caractères composée d'un à plusieurs mots séparés par un espace. On veut écrire une fonction `premierMot(s:str) -> str` qui retourne le premier mot rencontré. Par exemple, `premierMot("Hello world")` doit retourner "Hello", `premierMot("Hello")` doit retourner "Hello", `premierMot(" Hello")` doit retourner "". Veillez à commenter votre code.



2. Quelle est la complexité de cette fonction ? Exprimez-la en fonction de la longueur n de la chaîne de caractères s .



3. Que devriez-vous changer dans votre code pour renvoyer le dernier mot au lieu du premier ? `dernierMot("Hello world")` doit retourner "world", `dernierMot("Hello")` doit retourner "Hello", `dernierMot("Hello ")` doit retourner "". Vous pouvez indiquer simplement la ou les lignes de code à changer.

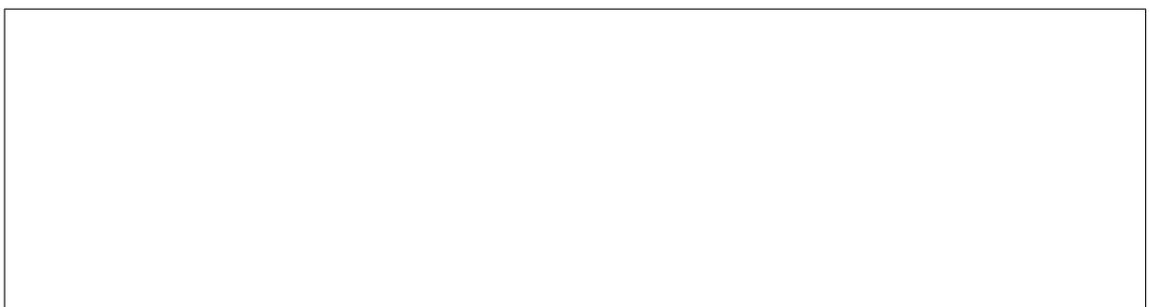


Exercice 6 Dans cet exercice, on réalisera une fonction `DegradeEuclidien(img: image)` qui prend une image en entrée et la remplit d'un dégradé de gris en fonction de la distance à l'origine.

1. La distance euclidienne entre le point p_1 de coordonnées (x_1, y_1) et le point p_2 de coordonnées (x_2, y_2) se calcule grâce à la formule suivante :

$$D(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Écrivez la fonction `distanceEucliOrigine(x: int, y:int) -> int` qui renvoie la distance euclidienne entre l'origine (de coordonnées $(0, 0)$) et le point de coordonnées (x, y) .

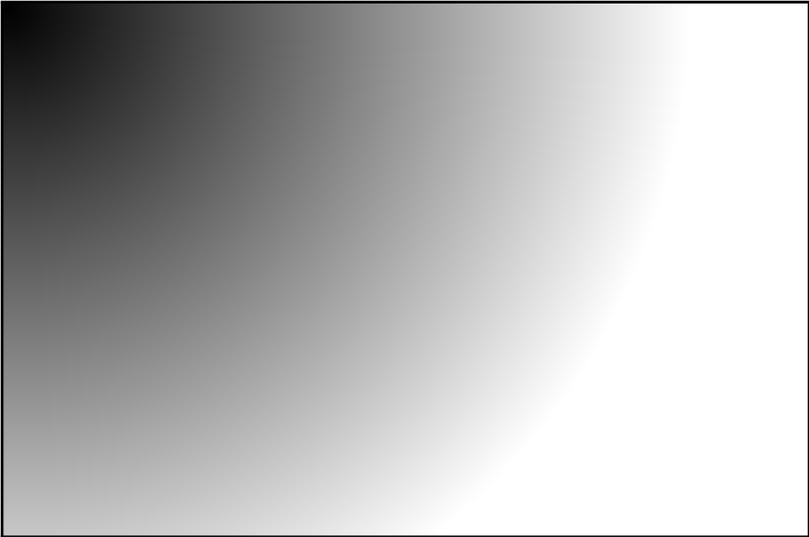


2. En utilisant la fonction `distanceEucliOrigine`, écrivez la fonction `DegradeEuclidien(img: image)`. Le résultat attendu est montré ci-dessous :

Nom:

Prénom:

Groupe:



FIN.

Fonctions mathématiques

Les fonctions mathématiques sont disponibles depuis le module `math`, par exemple la fonction `sqrt` calculant une racine carrée :

```
from math import sqrt
```

Certaines fonctions sont disponibles directement, par exemple l'opérateur puissance :

```
2**4
```

calcule ici 2^4 .

Création d'un tableau

Pour rappel, la syntaxe à utiliser pour créer un tableau `tab` de `n` éléments et contenant initialement que des 0 est :

```
tab = [0] * n
```

Manipulation d'images

Voici un rappel des principales fonctions disponibles pour manipuler les images (à vous de voir celles qui sont utiles pour ce devoir).

Ci-dessous, <code>img</code> est une image	
<code>ouvrirImage(nom:str) -> image</code>	Ouvre le fichier <code>nom</code> et retourne l'image contenue dedans, par exemple : <code>img = ouvrirImage("teapot.png")</code>
<code>nouvelleImage(large:int, haut:int) -> image</code>	Retourne une image de taille <code>large</code> × <code>haut</code> , initialement noire.
<code>afficherImage(img:image)</code>	Affiche l'image <code>img</code> .
<code>L = largeurImage(img)</code> <code>H = hauteurImage(img)</code>	Récupère la largeur de <code>img</code> . Récupère la hauteur de <code>img</code> .
<code>colorierPixel(img:image, x:int, y:int, (r,g,b))</code>	Peint le pixel <code>(x, y)</code> dans l'image <code>img</code> de la couleur <code>(r, g, b)</code>