

Nom:

Prénom:

Groupe:

Université de Bordeaux

Devoir surveillé du 18 novembre 2021

Département Licence, 4TPU114U / 4TPU115U Informatique 1

9h00 – 10h00

\_\_\_\_\_ Le sujet comporte 6 exercices et 7 pages, dont une page d'annexe \_\_\_\_\_  
**Aucun document n'est autorisé** – Toutes les fonctions de manipulation d'images disponibles sont  
rappelées en annexe page 7. Vous pouvez détacher cette annexe pour plus de facilité.

---

---

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

La correction sera sur plus que 20 points, ne vous inquiétez donc pas de la longueur du sujet.

---

**Exercice 1** Considérons la fonction `mystere` suivante :

```
def mystere(n: int) -> int :  
    s = 0  
    for i in range(n):  
        s = s + 2*i + 1  
    return s
```

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

2. Simulez l'exécution de l'appel `mystere(3)` en complétant le tableau suivant qui montre l'évolution des variables :

i		
s		

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

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

5. Quelle conjecture pouvez-vous émettre sur la valeur retournée par la fonction `mystere(n)`, pour un entier `n` positif? On ne demande pas de prouver cette conjecture.

6. Quelle est la complexité de la fonction `mystere(n)`, en fonction de son entrée `n`?  
(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** Indiquez l'ensemble des valeurs de `x` pour lesquelles l'expression suivante vaut `True`.

```
x >= 2 and x < 12 and (x == 3 or x % 2 != 1)
```

**Exercice 3** Écrire une fonction `estNulleSommeElements(L: list) -> bool` qui retourne `True` si la somme des éléments de la liste de nombres `L` vaut 0 et retourne `False` sinon.

Par exemple, l'appel `estNulleSommeElements([-4,7,5,-6])` retourne `False`, tandis que l'appel `estNulleSommeElements([8,-4,-3,5,-6])` retourne `True`.

**Exercice 4 – Dessin d'image**

- Vous disposez d'une annexe en page 7 contenant des fonctions de manipulation d'images •

1. Écrire en Python une fonction `imageDrapCadre(image)` qui modifie l'image passée en paramètre afin d'obtenir le drapeau Français (un tiers de bleu, un tiers de blanc, un tiers de rouge) à l'intérieur d'un cadre noir d'un pixel d'épaisseur sur toute la bordure.



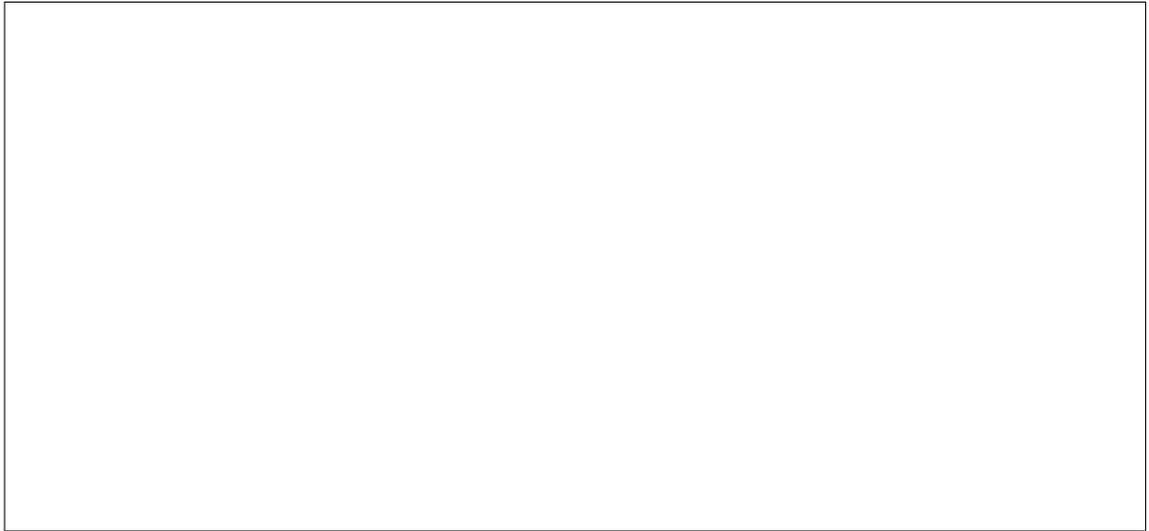
2. Écrire le code qui permet de tester la fonction et d'afficher le résultat produit par cette fonction sur une image de largeur 300 pixels et de hauteur 200 pixels.



**Exercice 5** On dispose d'une fonction Python `estVoyelle(c: str) -> bool` qui prend en argument un caractère `c` et retourne le booléen `True` si `c` est une voyelle, et `False` sinon. Par exemple, l'appel `estVoyelle("a")` retourne `True`, l'appel `estVoyelle("b")` retourne `False`.

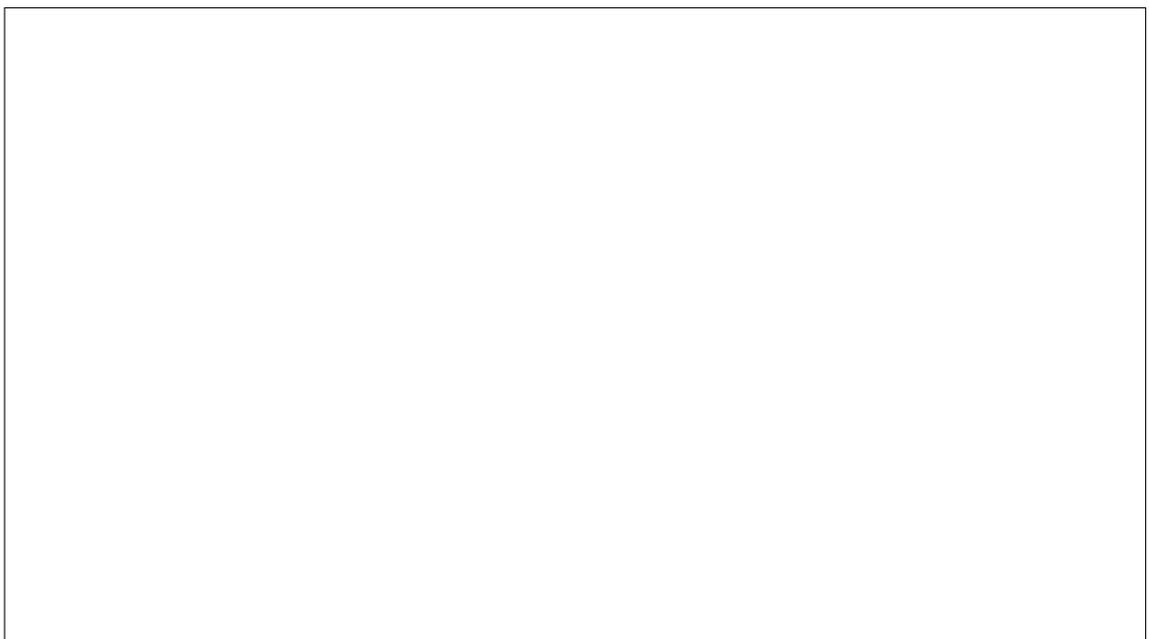
NB : on ne vous demande *pas* d'écrire cette fonction, par contre vous êtes invité-e à *l'utiliser* pour répondre aux deux questions ci-dessous.

1. Écrire une fonction `nbVoyelles(s: str) -> int`, qui prend en argument une chaîne de caractères `s` et retourne le nombre de voyelles apparaissant dans `s`. Par exemple l'appel `nbVoyelle("bonjour")` retourne 3.



2. La traduction en Javanais (un argot dérivé du français) d'un mot s'obtient en insérant devant chaque voyelle du mot les deux lettres "av". Par exemple "bonjour" est traduit par "bavonjavoavur".

Écrire une fonction `javanais(s: str) -> str`, qui prend en argument une chaîne de caractères `s` et retourne la traduction en javanais du mot `s`.



**Exercice 6** – La suite de Fibonacci

Considérons la suite de Fibonacci  $F$  définie par

$$F_0 = 0, \quad F_1 = 1, \quad \text{et pour tout } n \geq 2, \quad F_n = F_{n-1} + F_{n-2}$$

1. Remplissez le tableau suivant avec les 8 premiers termes de la suite de Fibonacci.

$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$
0	1						

2. Écrire en python une fonction `fibonacci(n: int) -> list` prenant pour argument un entier  $n$  (supposé supérieur ou égal à 2) et renvoyant un tableau de longueur  $n$  contenant les  $n$  premiers termes de la suite de Fibonacci.

FIN.



### 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>img = ouvrirImage(nom)</code>	Ouvre le fichier <code>nom</code> et retourne l'image contenue dedans (par exemple <code>ouvrirImage("teapot.png")</code> ).
<code>img = nouvelleImage(large,haut)</code>	Retourne une image de taille <code>large × haut</code> , initialement noire.
<code>afficherImage(img)</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, x, y, (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>