

Nom:

Prénom:

Groupe:

Université de Bordeaux
Département Licence, **J1MI1003**

Devoir surveillé du 4 novembre 2015
11h-12h30

_____ 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.

Exercice 1 Considérons le code Python suivant :

```
s = 0
for i in range(1, 3):
    for j in [2, 4, 3]:
        s = s + i * j
```

Simulez l'exécution de ce code en complétant le tableau suivant montrant l'évolution des variables s , i et j .

i	
j	
s	

Exercice 2 Dans la fonction `mystere` suivante, L est une liste d'entiers positifs supérieurs ou égaux à 0.

```
def mystere2(L):
    m = 0
    for i in L:
        if i < m:
            return False
        else:
            m = i
    return True
```

1. Simuler l'exécution de `mystere2(L)` lorsque L est égale à $[0, 2, 3, 7, 10]$ en donnant les valeurs successives des variables m et n dans le tableau suivant :

$L = [0, 2, 3, 7, 10]$

i	
m	

2. Que retourne la fonction lorsque L est égale à $[0, 2, 3, 7, 10]$?

3. Que retourne la fonction si L est plutôt égale à $[5, 3, 1, 7, 11]$?

4. Comment interpréter la valeur retournée par la fonction `mystere2` en général, pour une liste `L` quelconque ?

Exercice 3 Nombres Premiers

Rappel : On dit que i est un diviseur de n si le reste de la division de n par i est égal à 0.

1. Ecrire une fonction `estDiviseur(i,n)` qui retourne `True` si i est un diviseur de n et `False` sinon.

2. Un nombre est dit premier s'il n'a que 2 diviseurs : 1 et lui-même. Donnez la liste des nombres premiers inférieurs à 15.

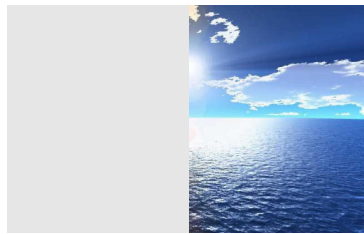
3. Ecrire une fonction `estPremier(n)` qui retourne `True` si n est premier, `False` sinon (on profitera du fait que seuls les nombres strictement inférieurs à n peuvent être diviseurs de n)

4. Ecrire une fonction `nbPremiers(n)` qui retourne le nombre de nombres premiers strictement plus petits que `n`.

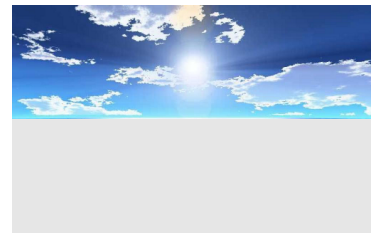
Exercice 4 Il s'agit d'écrire le code de deux fonctions python permettant de masquer la moitié d'une image à l'aide d'une couleur. Voici un exemple :



image initiale



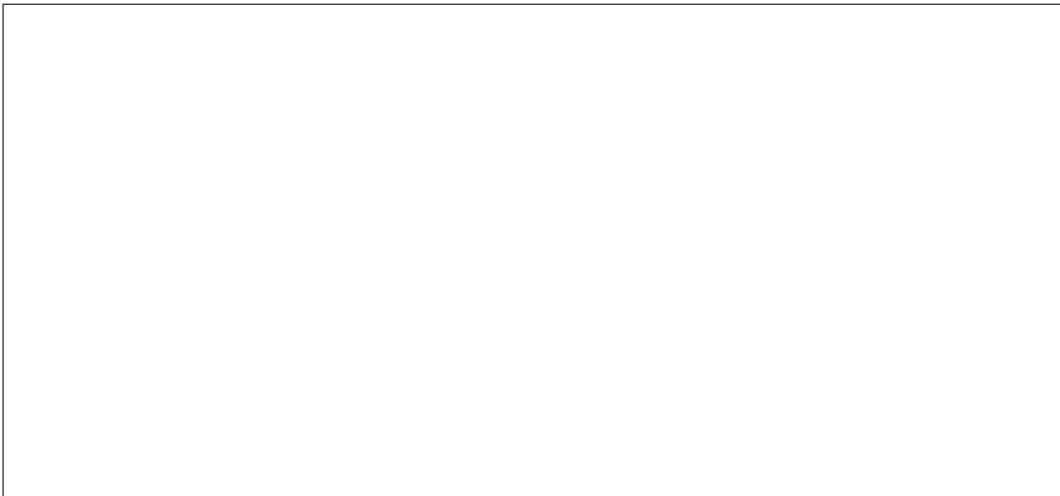
partie gauche masquée



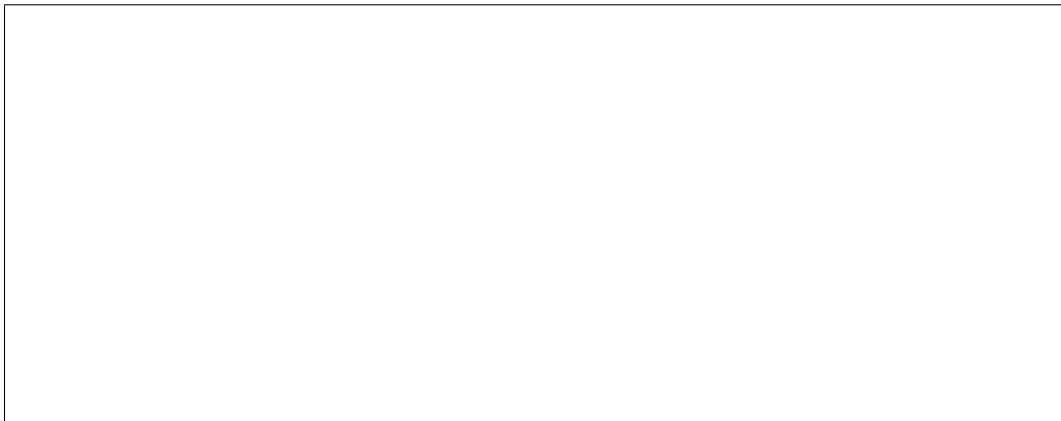
partie inférieure masquée

1. Écrire la fonction `masquerGauche(img, c)` qui remplit de la couleur `c` la moitié gauche de l'image `img` et laisse inchangé le reste de l'image.

2. Écrire la fonction `masquerBas(img, c)` qui remplit de la couleur `c` la moitié inférieure de l'image `img` et laisse inchangé le reste de l'image.



Exercice 5 Écrire la fonction `permutation(img)` qui, pour chaque pixel de l'image `img`, permutera ses composantes r , g et b de telle sorte que la composante rouge devienne la bleue, la verte devienne la rouge et la bleue devienne la verte. Par exemple, la couleur $(128, 32, 255)$ d'un pixel sera modifiée en la couleur $(32, 255, 128)$.



Exercice 6 L'objectif de cet exercice est d'écrire une fonction permettant de taguer une image. On considère deux images de **même taille**.

La première est une image quelconque, par exemple :

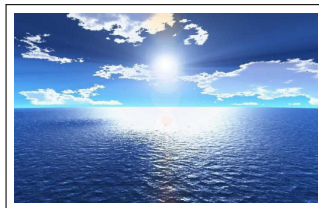


FIGURE 1 – ocean

La deuxième est une image composée d'un fond blanc et d'un texte de couleur c . Par exemple si $c=(0,0,0)$ le texte est noir comme dans l'image ci-dessous :



FIGURE 2 – texte

On souhaite écrire une fonction `ecrireDessus(img1, img2, c)` qui utilisera le texte de couleur c de `img2` pour écrire sur `img1`. Par exemple l'appel `ecrireDessus(ocean, texte, (0,0,0))` transformera l'image `ocean` ainsi :

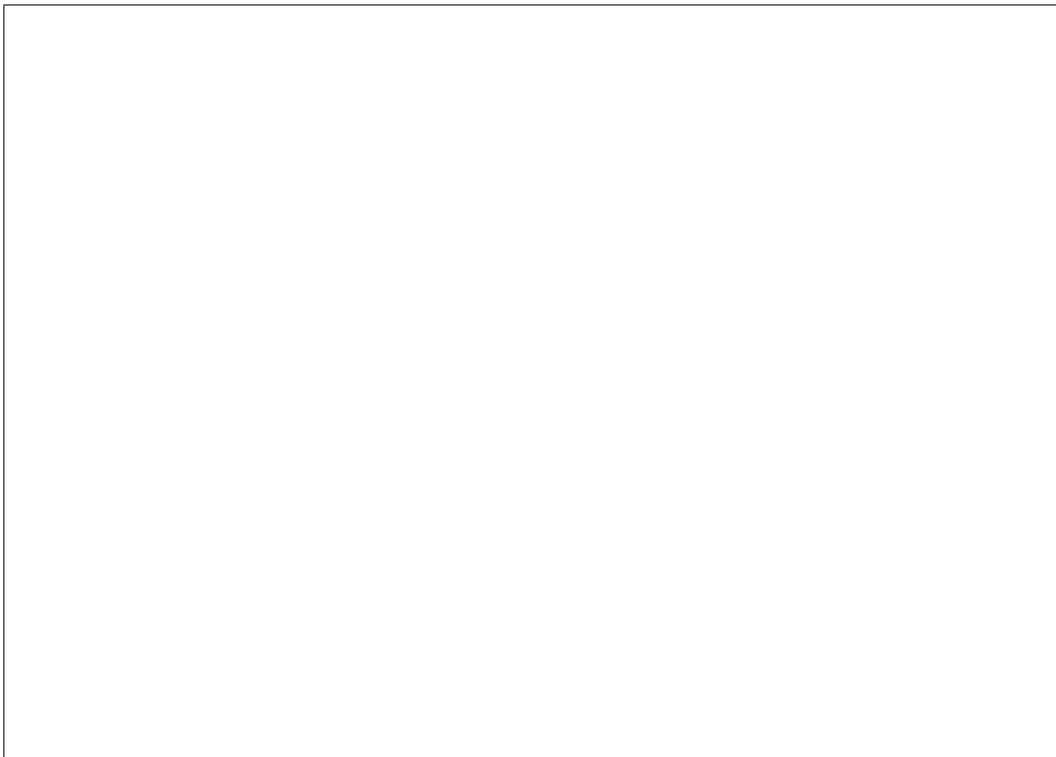


FIGURE 3 – ocean modifiée

L'algorithme utilisé est le suivant :

Pour chaque pixel de coordonnées (x,y) dans `img1`, si le pixel **aux mêmes coordonnées** dans `img2` est de couleur c on remplace la couleur initiale du pixel de coordonnées (x,y) dans `img1` par la couleur c .

Écrire la fonction `ecrireDessus(img1, img2, c)` qui modifie `img1` selon l'algorithme précédent.



Nom:

Prénom:

Groupe:

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

L'argument <code>img</code> est une image	
<code>open(nom)</code>	Ouvre le fichier <i>nom</i> et retourne l'image contenue dedans (par exemple <code>open("teapot.png")</code>).
<code>Image.save(img, nom)</code>	Sauvegarde l'image <i>img</i> dans le fichier <i>nom</i> .
<code>new("RGB", (largeur, hauteur))</code>	Retourne une image de taille <i>large</i> × <i>haut</i> , initialement noire.
<code>Image.show(img)</code>	Affiche l'image <i>img</i> .
<code>(largeur, hauteur) = img.size</code>	Récupère la largeur et la hauteur de <i>img</i> .
<code>Image.putpixel(img, (x,y), (r,g,b))</code>	Peint le pixel (x, y) dans l'image <i>img</i> de la couleur (r, g, b)
<code>Image.getpixel(img, (x,y))</code>	Retourne la couleur du pixel (x, y) dans l'image <i>img</i>