

Nom :

Prénom :

Groupe :

_____ 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 et de graphes disponibles sont rappelées en annexe page 7. Vous pouvez détacher cette annexe pour plus de facilité.

Exercice 1 On considère la fonction suivante :

```
def mystere(img,c):  
    (large,haut) = img.size  
    for i in range(large):  
        j = i//2  
        Image.putpixel(img, (i,j), c)
```

On crée initialement une image `monimage` et l'on appelle la fonction `mystere` sur `monimage` ainsi :

```
monimage = new("RGB", (5,4))  
mystere(monimage, (0,0,255))
```

1. Simulez cette exécution de `mystere` en complétant le tableau suivant (on ne demande donc que l'évolution de `i` et `j` sans encore se soucier de l'effet des appels à `Image.putpixel`)

i	
j	

2. Voici une représentation de l'image `monimage`. Cochez les pixels se retrouvant colorés par l'appel à la fonction `mystere`.

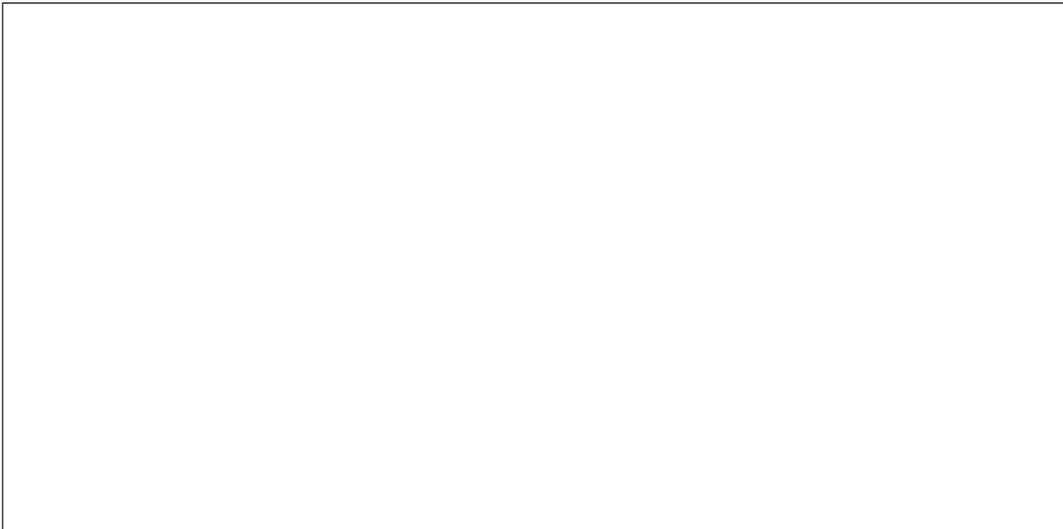
3. De quelle couleur les pixels se trouvent-ils colorés ?

--

Exercice 2 Tracé de bordure

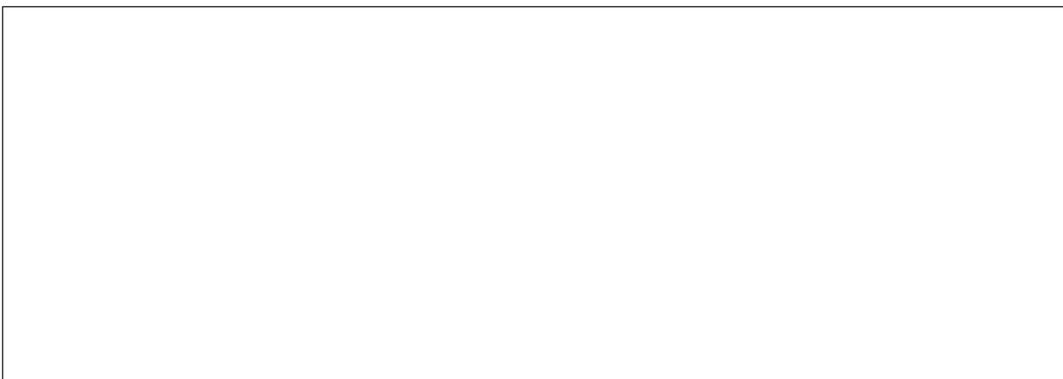
1. Écrire la fonction `bordure(img,c)` qui trace avec une couleur `c` le bord de l'image `img`. Par exemple, avec une image de 5x4 pixels, les pixels du bord de l'image sont ceux marqués d'une croix :

X	X	X	X	X
X				X
X				X
X	X	X	X	X



Exercice 3 Graphes & degrés

1. Écrire une fonction `nbVoisinsDegre(s, d)` qui compte le nombre de sommets de degré `d` qui sont voisins de `s`. Pour simplifier, le graphe est supposé *simple*, c'est-à-dire sans arête multiple ou boucle.



2. Écrire une fonction `tousSommetsDegre(G, d)` qui renvoie le booléen `True` si tous les sommets du graphe `G` sont de degré `d` et qui renvoie `False` sinon.

Exercice 4 Un programmeur débutant, utilisant la bibliothèque `bibV3`, a écrit la fonction suivante pour tester si un graphe admet au moins une arête dont les deux sommets extrémités sont colorés l'un en bleu et l'autre en rouge :

```
def bleuRouge(G):  
    for s in listeSommets(G):  
        for t in listeVoisins(G):  
            if couleurSommet(s,t) = (bleu,rouge):  
                return True  
    return False
```

1. Cette fonction comporte 5 erreurs : cochez-les ci-dessus et expliquez en quelques mots.

2. Donnez une version corrigée de la fonction `bleuRouge`.

Exercice 5 Luminosité

Écrire une fonction `luminosite(img)` qui calcule la luminosité moyenne d'une image (entre 0 et 255), c'est-à-dire la moyenne sur l'ensemble de l'image de la luminosité des pixels, la luminosité d'un pixel étant la moyenne des trois composantes rouge, vert et bleu.



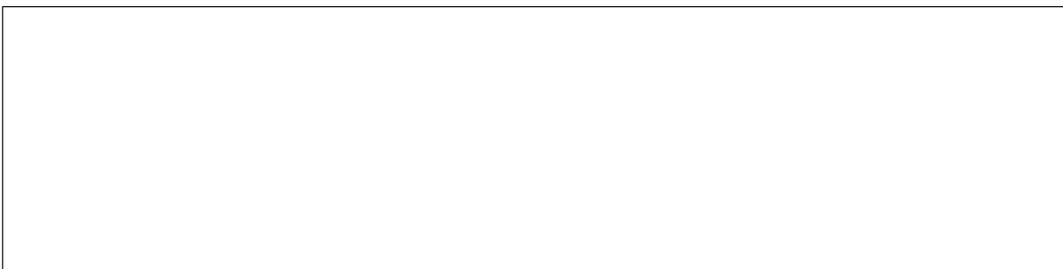
TSVP

Exercice 6 Un reporter a pris plusieurs photographies d'un paysage. Elles sont de même hauteur mais montrent des points de vue décalés. Il souhaite maintenant les mettre côte à côte pour obtenir une image panoramique de ce paysage (sans se soucier des problèmes de perspective).

1. Écrivez une fonction `assembler(imgGauche, imgDroite)` qui retourne une nouvelle image correspondant à l'assemblage horizontal des images `imgGauche` et `imgDroite`.



2. Écrivez une fonction `assemblerTrois(imgGauche, imgCentre, imgDroite)` qui assemble trois images horizontalement, en utilisant la fonction précédente (il n'est donc pas nécessaire d'avoir traité la question précédente pour traiter celle-ci).



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

L'argument <i>img</i> 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>

L'argument <i>G</i> est un graphe	
<code>listeSommets(G)</code>	retourne la <i>liste</i> des <i>sommets</i> de <i>G</i>
<code>nbSommets(G)</code>	retourne le <i>nombre</i> de <i>sommets</i> de <i>G</i>
<code>sommetNom(G, etiquette)</code>	retourne le <i>sommet</i> de <i>G</i> désigné par son <i>nom</i> (<i>etiquette</i>). Exemple : <code>sommetNom(Europe, 'Italie')</code>
<code>dessinerGraphe(G)</code> ou simplement <code>dessiner(G)</code>	demande (très poliment) au logiciel <i>Graphviz</i> de dessiner le graphe <i>G</i> ; voir page suivante pour les détails

L'argument <i>s</i> est un sommet	
<code>listeVoisins(s)</code>	retourne la <i>liste</i> des <i>voisins</i> de <i>s</i>
<code>degre(s)</code>	retourne le <i>degré</i> de <i>s</i>
<code>nomSommet(s)</code>	retourne le <i>nom</i> (étiquette) de <i>s</i>
<code>colorierSommet(s,c)</code>	colorie <i>s</i> avec la couleur <i>c</i> . Exemples de couleurs : 'red', 'green', 'blue', 'white', 'cyan', 'yellow'
<code>couleurSommet(s)</code>	retourne la <i>couleur</i> de <i>s</i>