

Le sujet comporte 6 exercices et 9 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 9. Vous pouvez détacher cette annexe pour plus de facilité.

Exercice 1 On considère le code suivant :

```
def mystere(s1, s2):
    nb = 0
    for s in listeVoisins(s1):
        marquerSommet(s)
    for s in listeVoisins(s2):
        if estMarqueSommet(s):
            nb = nb + 1
    return nb
```

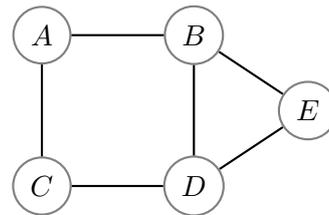


FIGURE 1 – Le graphe *Grrr*

La figure 1 représente un graphe *Grrr* à cinq sommets *a*, *b*, *c*, *d*, *e*, d'étiquettes respectives 'A', 'B', 'C', 'D', 'E'.

1. On suppose qu'aucun sommet du graphe *Grrr* n'est marqué.

Que vaut `mystere(a, d)` ?

Quels sommets sont encore marqués après cet appel ?

2. On suppose qu'aucun sommet du graphe *Grrr* n'est marqué.

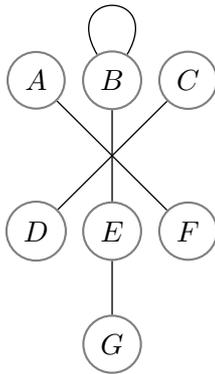
Que vaut `mystere(b, d)` ?

Quels sommets sont encore marqués après cet appel ?

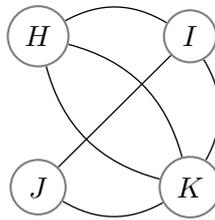
3. De façon générale, que retourne `mystere(u, v)` en fonction des sommets *u* et *v* d'un graphe *G* simple, dont aucun sommet n'est marqué.

Exercice 2 Les questions de cet exercice portent toutes sur ces trois graphes : G_0 , G_1 et G_2 .

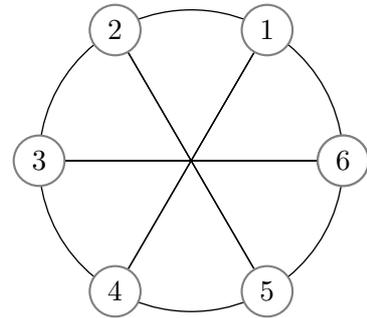
Graphes G_0



Graphes G_1



Graphes G_2



1. Remplir le tableau ci-dessous en n'oubliant pas de justifier votre réponse lorsque c'est nécessaire.

	G_0	G_1	G_2
Est-il connexe? (si non, justifier)			
Est-il simple? (si non, justifier)			
Quel est son nombre chromatique, s'il existe? (justifier)			
Donner un cycle élémentaire de longueur maximale			

Numéro d'anonymat :

Exercice 3 Nombres de Cullen. Pour tout entier $n \geq 0$, le *nombre de Cullen d'indice n* est le nombre $n \times 2^n + 1$.

1. Écrire une fonction `cullen(n)` prenant en paramètre un entier `n` et retournant le nombre de Cullen d'indice `n`. On rappelle que Python peut calculer une puissance a^k avec la notation `a**k`.

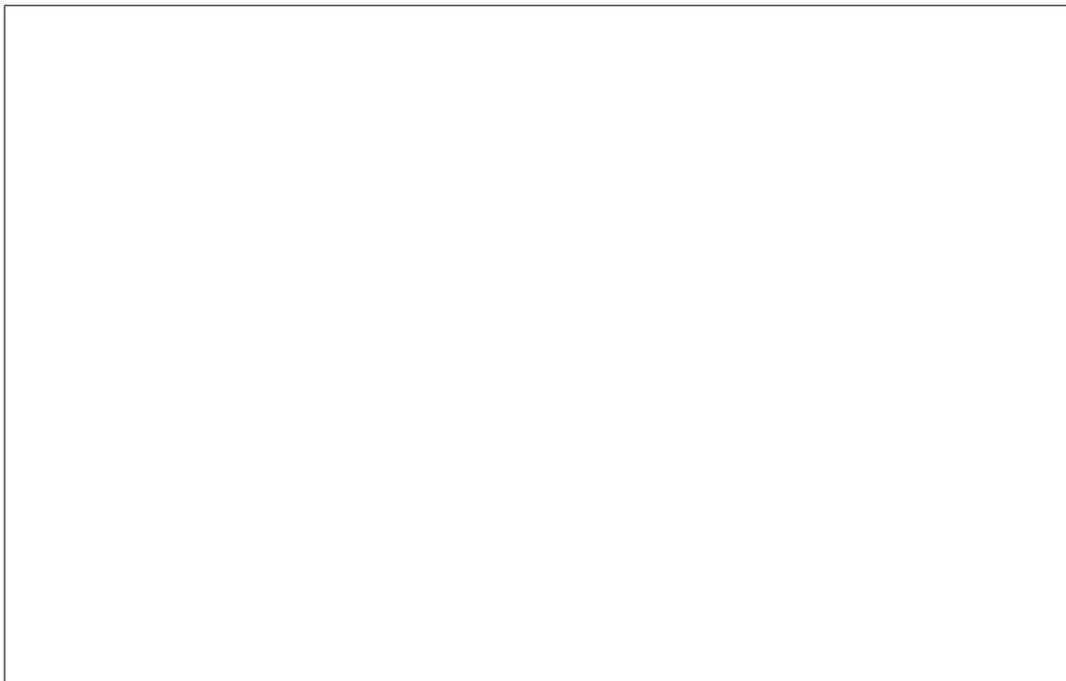
2. Écrire une fonction `indiceCullen(x)` prenant en paramètre un entier `x`, et retournant le plus grand entier n tel que `x` soit supérieur au nombre de Cullen d'indice n .

Exercice 4

1. Écrire une fonction `seuil(img, lim)` qui transforme en noir les pixels de l'image `img` dont la luminosité est inférieure à une limite `lim`. Pour rappel, la luminosité d'un pixel est la moyenne de ses trois composantes rouge, vert et bleu.



2. Écrire une fonction `pixelMaxHaut(img)` qui renvoie la position (x,y) du pixel le plus haut dans l'image dont la luminosité est non nulle (on suppose qu'il en existe au moins un). Si plusieurs pixels répondent à ce critère, on renverra la position de celui qui est situé le plus à gauche.



Exercice 5

1. Écrire une fonction `marqueAvecVoisinNonMarque(G)` qui prend en argument un graphe G et renvoie *un sommet marqué qui a un voisin non-marqué* (si G possède un tel sommet) et renvoie `None` (dans le cas contraire).

2. Écrire une fonction `marquerVoisins(s)` qui prend en argument un sommet s et marque *tous* les voisins de ce sommet.

3. Écrire une fonction `marquerCompConnexe(G, s)` qui prend en argument un graphe G , dont tous les sommets sont démarqués, et un sommet s de ce graphe, et marque *tous* les sommets de la composante connexe de s .

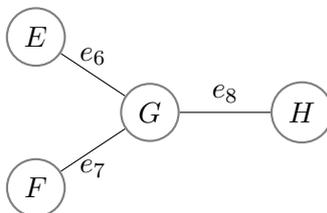
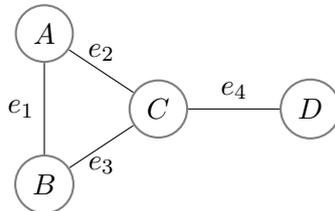
Indication : On utilisera l'algorithme suivant :

- marquer s ,
- tant qu'il existe un sommet marqué qui a un voisin non-marqué,
 - marquer tous les voisins d'un tel sommet.

4. Écrire une suite d'instructions Python permettant de marquer la composante connexe du sommet appelé **Belgique** dans le graphe **Europe**, puis de visualiser le résultat.

Exercice 6 Dans un graphe connexe, un *isthme* est une arête dont la suppression détruit la connexité du graphe. Autrement dit, une arête e est un isthme d'un graphe connexe G si et seulement si le graphe $G \setminus e$ (le graphe G privé de l'arête e) se décompose en deux composantes connexes.

1. Pour chacun des graphes suivants expliciter tous les isthmes. (Il y en a 4 sur l'ensemble des deux graphes).



2. Montrer que dans un arbre, toute arête est un isthme.

3. Soit G un graphe cubique (c'est-à-dire un graphe dont tous les sommets sont de degré 3), connexe, contenant un isthme e . En appliquant la formule de poignées de mains à chacune des composantes connexes, montrer que les deux composantes connexes de $G \setminus e$ ont un nombre impair de sommets.



Numéro d'anonymat :

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 <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>

L'argument <code>G</code> est un graphe	
<code>listeSommets(G)</code>	retourne la <i>liste</i> des <i>sommets</i> de <code>G</code>
<code>nbSommets(G)</code>	retourne le <i>nombre</i> de <i>sommets</i> de <code>G</code>
<code>sommetNom(G, etiquette)</code>	retourne le <i>sommet</i> de <code>G</code> 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 <code>G</code> ; voir page suivante pour les détails

L'argument <code>s</code> est un sommet	
<code>listeVoisins(s)</code>	retourne la <i>liste</i> des <i>voisins</i> de <code>s</code>
<code>degre(s)</code>	retourne le <i>degré</i> de <code>s</code>
<code>nomSommet(s)</code>	retourne le <i>nom</i> (étiquette) de <code>s</code>
<code>colorierSommet(s,c)</code>	colorie <code>s</code> avec la couleur <code>c</code> . Exemples de couleurs : <code>'red'</code> , <code>'green'</code> , <code>'blue'</code> , <code>'white'</code> , <code>'cyan'</code> , <code>'yellow'</code>
<code>couleurSommet(s)</code>	retourne la <i>couleur</i> de <code>s</code>
<code>marquerSommet(s)</code>	marque le sommet <code>s</code>
<code>demarquerSommet(s)</code>	démarque le sommet <code>s</code>
<code>estMarqueSommet(s)</code>	retourne <code>True</code> si <code>s</code> est marqué, <code>False</code> sinon
<code>listeAretesIncidentes(s)</code>	retourne la <i>liste</i> des arêtes <i>incidentes</i> à <code>s</code>