

Numéro d'anonymat :

Groupe :

Le sujet comporte 4 exercices et 7 pages, dont une page d'annexe.

Aucun document n'est autorisé : toutes les fonctions de manipulation des graphes disponibles sont rappelées en annexe, page 7, que vous pouvez détacher pour plus de facilités.

Exercice 1 Soit la fonction Python suivante :

```
def mystere(v):  
    r = 0  
    while v > 0 :  
        if v % 10 == 0 :  
            r = r + 1  
        v = v // 10  
    return r
```

Rappel : en python, le quotient entier de a par b est noté $a//b$, et le reste de la division entière est noté $a\%b$ (modulo).

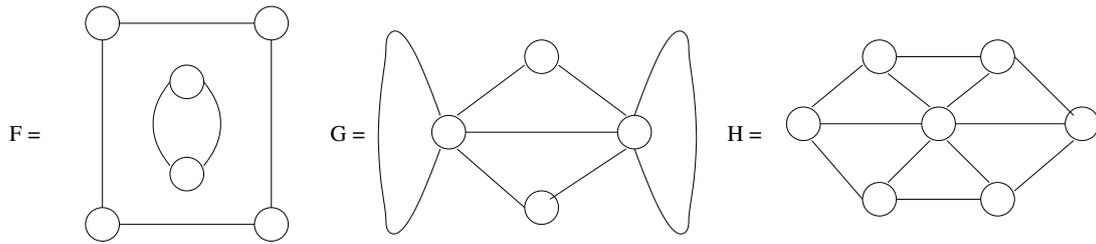
1. Donner les valeurs successives prises par les variables v et r lors de l'appel `mystere(500601)`, ainsi que la valeur retournée.

v	
r	

2. D'une manière générale, indiquer ce que retourne `mystere(v)` en fonction de l'entier strictement positif v .

3. En vous inspirant de la fonction `mystere`, écrire une fonction `plusGrandChiffre(v)` qui retourne le plus grand chiffre contenu dans le nombre entier v . Par exemple, la valeur retournée par `plusGrandChiffre(500601)` est 6.

Exercice 2 On rappelle que K_6 désigne le graphe complet à 6 sommets. On appelle F , G et H les graphes suivants :



1. Remplir le tableau suivant (en répondant par OUI ou par NON lorsque cela est demandé). **Attention**, les réponses fausses enlèvent des points :

Graphe	F	G	H	K_6
Connexe ? Si non, justifier				
Simple ? Si non, justifier				
Eulérien ? (justifier)				
Peut-il être bien colorié ? Si non, justifier				
Si oui, nombre chromatique ? (justifier)				

2. Un graphe composé seulement d'un cycle est-il toujours 2-coloriable ? Justifier.

Numéro d'anonymat :

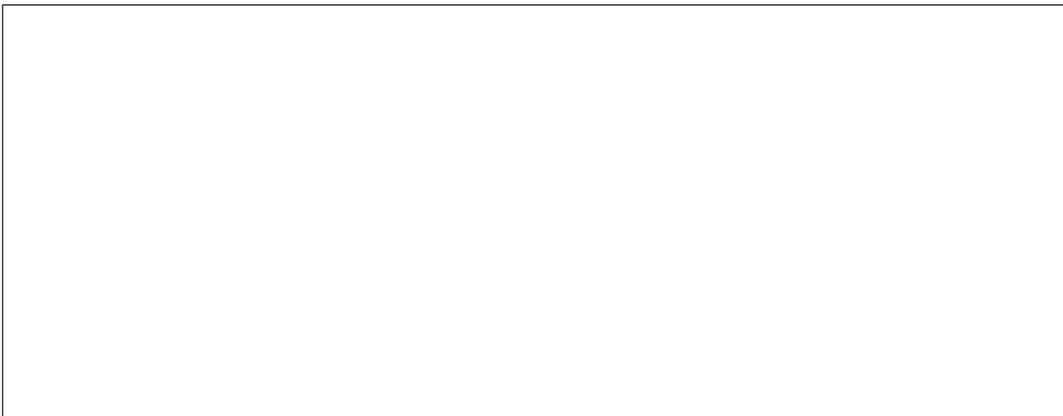
Groupe :

Exercice 3

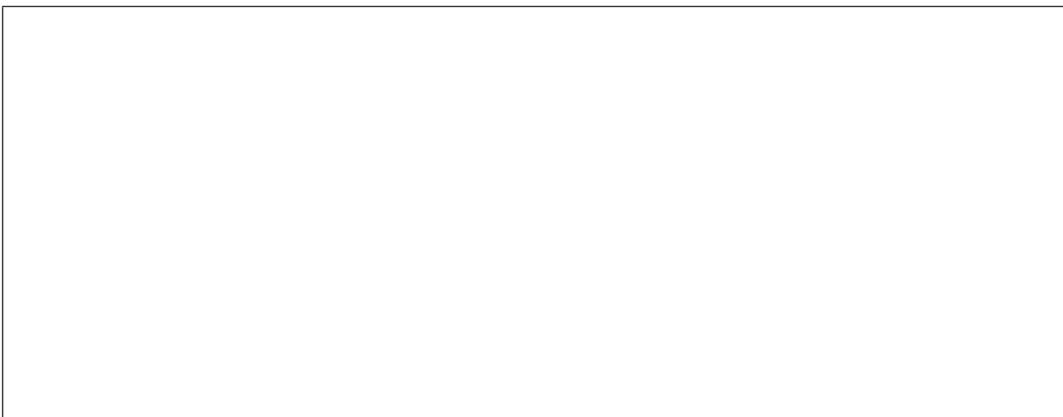
1. Écrire une fonction `compterAretesMarquees(G)` qui renvoie le nombre d'arêtes marquées du graphe `G`.



2. Écrire une fonction `marquerAretesBicolores(G)` qui marque les arêtes du graphe `G` pour lesquelles les deux sommets incidents sont coloriés avec une couleur distincte. On supposera que `'white'` est aussi une couleur.



3. Écrire une fonction `sommetEstNonMarqueAvecAreteMarquee(s)` qui renvoie `True` si le sommet `s` est non marqué et qu'il est incident à au moins une arête marquée, et renvoie `False` sinon.



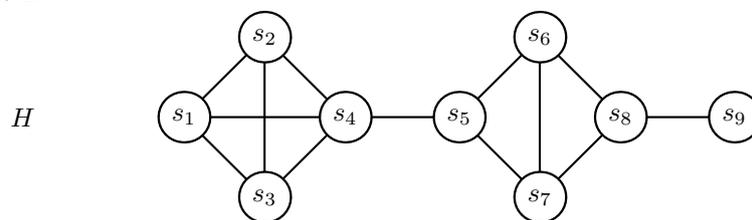
4. Écrire une fonction `listeSommetsNonMarquesAvecAretesMarquees(G)` qui renvoie la liste des sommets du graphe G qui sont non marqués et incidents à au moins une arête marquée. La fonction précédente pourra être utilisée, même si vous n'avez pas su l'écrire.

5. Écrire une fonction `marquerSommetsAvecAretesMarquees(G)` qui marque tous les sommets du graphe G dont au moins une arête incidente est marquée. On pourra utiliser une fonction précédente, même si vous n'avez pas su l'écrire.

Exercice 4 Une **clique** dans un graphe G est un sous-ensemble de sommets C deux-à-deux adjacents, i.e., tel que pour toute paire de sommets distincts de C , il existe une arête dans G reliant les deux sommets, i.e. c'est un sous-graphe complet.

Une clique C est *maximale* si pour tout sommet s qui n'est pas dans C , $C \cup \{s\}$ n'est plus une clique.

Voici le graphe H .



Numéro d'anonymat :

Groupe :

1. Est-ce que $\{s_1, s_2, s_3\}$ est une clique de H ? Une clique maximale? Sinon, trouver dans H une clique maximale contenant $\{s_1, s_2, s_3\}$.

2. Déterminer toutes les cliques maximales du graphe H (il y en a 5).

3. Donner trois cliques disjointes (c'est-à-dire sans aucun sommet commun) qui ensemble couvrent tous les sommets de H .

4. Est-ce qu'un graphe simple peut contenir une clique maximale à un seul sommet? Si c'est le cas, quel est le degré d'un tel sommet?

5. Est-il possible qu'un graphe simple ne possède qu'une seule clique maximale ? Si c'est le cas, de quel(s) graphe(s) s'agit-il ?



FIN.

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

L'argument G est un graphe	
<code>listeSommets(G)</code>	retourne la <i>liste</i> des <i>sommets</i> de G .
<code>nbSommets(G)</code>	retourne le <i>nombre</i> de <i>sommets</i> de G .
<code>sommetNom(G,etiquette)</code>	retourne le <i>sommet</i> de G désigné par son <i>nom</i> (étiquette). Exemple : <code>sommetNom (Europe, 'Italie')</code> .
<code>sommetNumero(G,i)</code>	retourne le <i>sommet</i> numéro i dans G ; la numérotation commence à 0.

L'argument s est un sommet	
<code>listeVoisins(s)</code>	retourne la <i>liste</i> des <i>voisins</i> de s .
<code>degre(s)</code>	retourne le <i>degré</i> de s .
<code>nomSommet(s)</code>	retourne le <i>nom</i> (étiquette) de s .
<code>colorierSommet(s,c)</code>	colorie s avec la couleur c . Exemples de couleurs : 'red', 'green', 'blue', None.
<code>couleurSommet(s)</code>	retourne la <i>couleur</i> de s .
<code>marquerSommet(s)</code> <code>demarquerSommet(s)</code>	marque ou démarque s .
<code>estMarqueSommet(s)</code>	retourne True si s est marqué, False sinon.
<code>listeAretesIncidentes(s)</code>	retourne la <i>liste</i> des arêtes <i>incidentes</i> à s .

L'argument a est une arête	
<code>nomArete(a)</code>	retourne le <i>nom</i> (étiquette) de a .
<code>marquerArete(a)</code> <code>demarquerArete(a)</code>	marque ou démarque a .
<code>estMarqueeArete(a)</code>	retourne True si a est marquée, False sinon.

Arguments : un sommet s et une arête a	
<code>sommetVoisin(s,a)</code>	retourne le voisin de s en suivant l'arête a .