

Nom :

Prénom :

Groupe :

Le sujet comporte 6 exercices et 6 pages, plus une page d'annexe.

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

Exercice 1 Voici une fonction Python :

```
def f(n,k):  
    c = 1  
    for i in range(k):  
        c = c * (n - i) // (i + 1)  
    return c
```

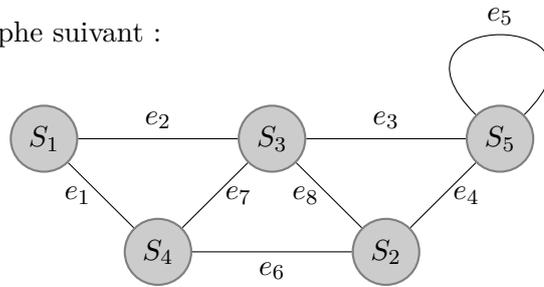
1. Simuler l'exécution de $f(5,3)$ en complétant le tableau suivant :

i	0
c	1

2. Que vaut $f(5,3)$?

3. Écrire une fonction Python $g(n)$ qui calcule $g(n) = \sum_{k=0}^{k=n} f(n, k)$:

Exercice 2 Soit le graphe suivant :



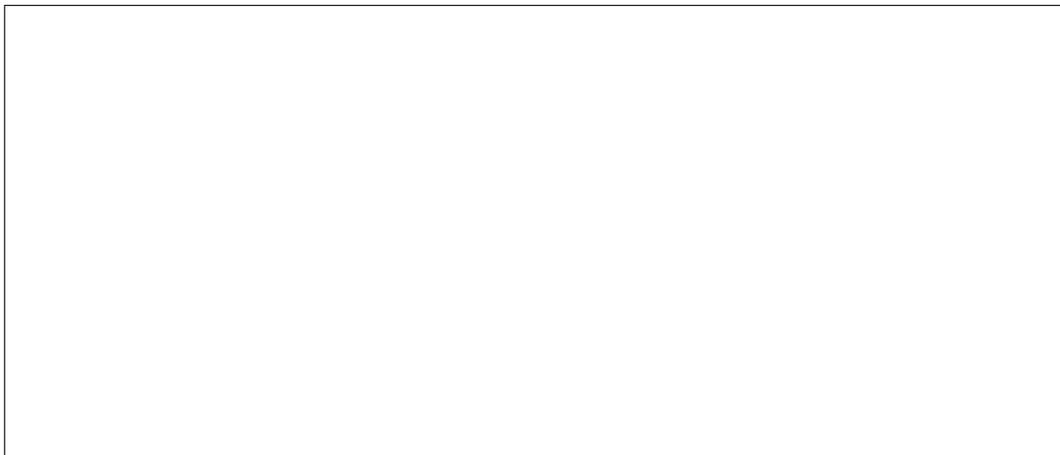
1. Donner un exemple de cycle simple de longueur 5.

2. Est-ce qu'il existe un autre exemple de cycle simple de longueur 5? (autre que le même cycle parcouru dans l'autre sens) Si oui, indiquer lequel, si non, expliquer pourquoi.

Exercice 3 Dans cet exercice tous les graphes sont simples, et tous les cycles aussi. La *maille* d'un graphe G est la longueur du cycle le plus court de G ; on rappelle qu'un graphe G est *cubique* si tous les sommets de G sont de degré 3.

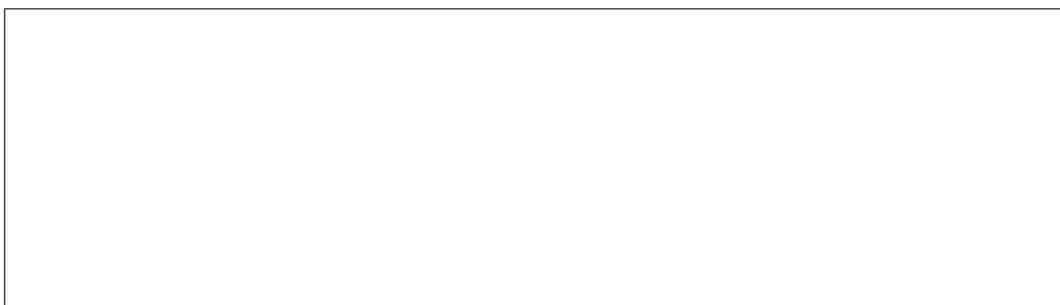
Le but de l'exercice est de construire un graphe cubique de maille 4 possédant le moins de sommets possible.

1. Il existe un seul graphe (simple) cubique possédant quatre sommets; expliquer pourquoi et dessiner ce graphe :

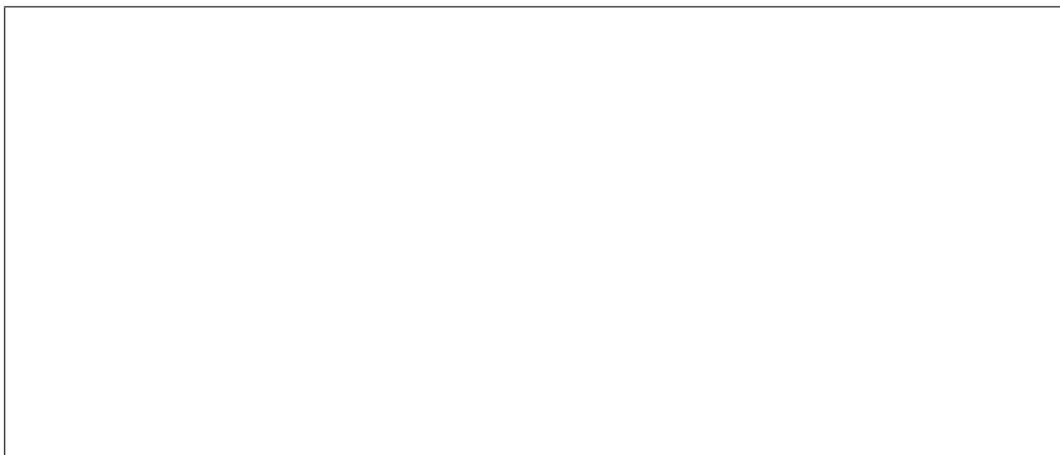


2. Quelle est la maille du graphe précédent?

3. Il n'existe pas de graphe cubique (simple) possédant cinq sommets; expliquer pourquoi :

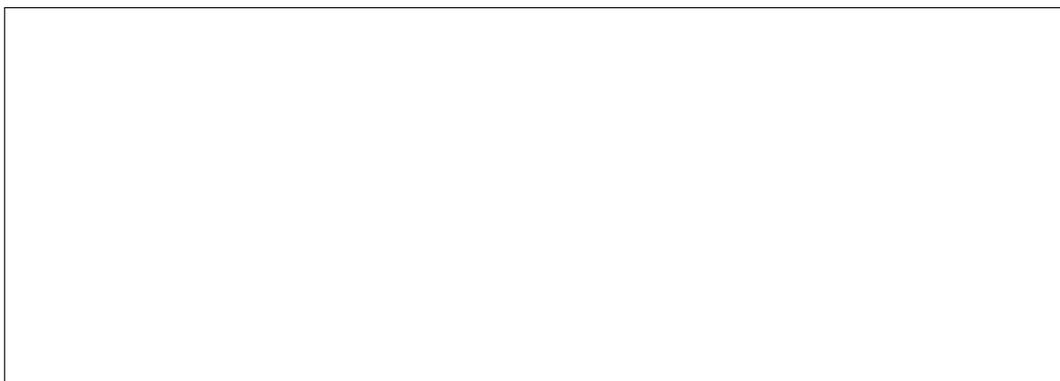


4. Dessiner un graphe cubique de maille 4 et possédant six sommets :



Exercice 4 Justifier votre réponse par un contre-exemple ou une explication :

1. Est-ce qu'un graphe non simple possède forcément un cycle ?



2. Est-ce qu'un graphe possédant un cycle est forcément non simple ?



3. Est-ce qu'un graphe simple dont les n sommets sont tous de degré $n - 1$ est forcément connexe ?



Exercice 5 Voici des expressions python faisant appel aux fonctions de la bibliothèque de graphes, dont la documentation est rappelée en annexe. Indiquer lesquelles sont correctes (il y en a très peu), et pour les autres, pourquoi elles ne le sont pas, en quelques mots. Comme d'habitude, G est un graphe (non vide), s est un sommet, c est une couleur.

1. `colorierSommet(listeVoisins(s),c)`

2. `degre(G)`

3. `marquerSommet(sommetNumero(G,0))`

4. `nbSommets(listeSommets(G))`

5. `colorierSommet(G,c)`

Exercice 6

1. Écrire une fonction Python `sommetsTousColories(G)` qui retourne `True` si tous les sommets du graphe G sont coloriés (c-à-d de couleur différente de `None`), et `False` sinon :

2. Écrire une fonction Python `nbSommetsMarques(G)` qui retourne le nombre de sommets marqués dans le graphe G :

3. Écrire une fonction Python `sommetMarque(G)` qui retourne un sommet marqué s'il en existe un, et `None` sinon :

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.