

Nom :

Prénom :

Groupe :

Le sujet comporte 5 exercices et 9 pages, dont une page d'annexe.

*Aucun document n'est autorisé* : toutes les fonctions de manipulation des graphes disponibles sont rappelées en annexe, page 9, que vous pouvez détacher pour plus de facilités.

**Exercice 1** Voici une fonction Python :

```
def mystere(x,n):  
    r = 1  
    while (x != 0) and (x % n == 0) :  
        x = x // n  
        r = r * n  
    return r
```

1. Simuler l'exécution de `mystere(7,2)` en donnant les valeurs successives des variables `x` et `r` dans le tableau suivant :

$x$	7
$r$	1

Donner la valeur retournée par `mystere(7,2)` :

2. De même pour l'exécution de `mystere(48,2)` :

$x$	48
$r$	1

3. De même pour l'exécution de `mystere(48,3)` :

$x$	48
$r$	1

4. D'une manière générale, expliquer ce que retourne `mystere(x,n)` en fonction de `x` et de `n`.

**Exercice 2** Soit un graphe à 6 sommets, dont voici les listes de voisins :

- voisins de A : [ C, C ]
- voisins de B : [ ]
- voisins de C : [ A, D, A ]
- voisins de D : [ C, E, F ]
- voisins de E : [ D, F ]
- voisins de F : [ E, D, F, F ]

1. Donner le degré de chaque sommet du graphe :

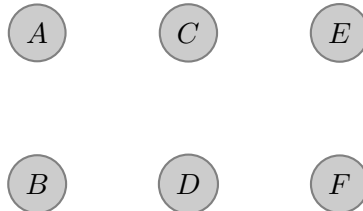
$d(A) =$  ,  $d(B) =$  ,  $d(C) =$  ,  $d(D) =$  ,  $d(E) =$  ,  $d(F) =$

2. Le graphe possède-t-il des boucles ? Si oui combien et pour quel(s) sommet(s) ?

3. Le graphe possède-t-il des arêtes multiples ? Si oui combien et pour quels sommets ?

4. Le graphe possède-t-il des sommets isolés ? Si oui combien et le(s)quel(s) ?

5. Compléter le dessin de ce graphe.



**Exercice 3**

1. Écrire une fonction `sontVoisins(s1,s2)` qui teste si deux sommets `s1` et `s2` sont voisins dans un graphe `G`.

Nom :

Prénom :

Groupe :

2. En utilisant la fonction `sontVoisins` (même si vous n'avez pas su l'écrire, donc), écrire une fonction `chemin2Valide(s1,s2,s3)` qui teste s'il existe une chaîne de longueur 2 passant successivement par les sommets `s1`, `s2` puis `s3`.

3. En utilisant la fonction `sontVoisins`, écrire une fonction `nbVoisinsCommuns(G,s1,s2)` qui compte le nombre de sommets qui sont voisins à la fois à `s1` et à `s2`.

4. En utilisant la fonction `sontVoisins`, écrire une fonction `estUnStable(L)` qui teste si la liste de sommets `L` est un *stable*, c'est-à-dire renvoie `True` s'il n'y a aucune arête entre toute paire de sommets de cette liste `L`, et `False` sinon.

**Exercice 4** Une erreur s'est glissée dans la fonction ci-dessous, censée retourner le degré maximum d'un graphe.

```
def degreMax (G):  
    m = 0  
    for s in listeSommets (G):  
        d = degre (s)  
        if d > m:  
            m = d  
    return m
```

1. Quel est la valeur (erronée) renvoyée par cette fonction?

2. Quel changement doit-on faire pour corriger la fonction? Expliquez.

Nom :

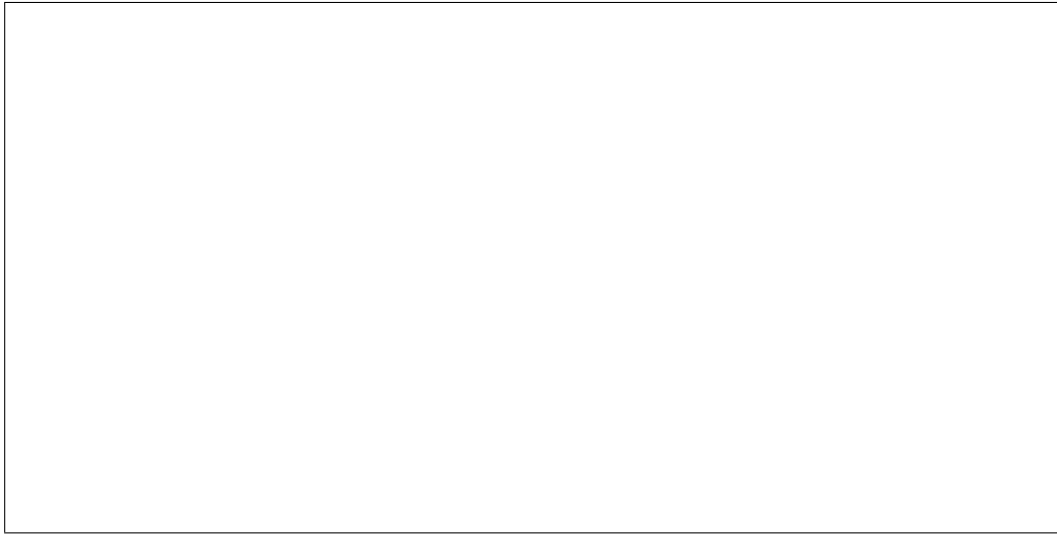
Prénom :

Groupe :

Des erreurs d'utilisation de fonctions se sont glissées dans la fonction ci-dessous, censée colorier les sommets de degré pair.

```
def colorierPairs (G,c):  
    colorierSommet(listeSommets(G),"white")  
    n = nbSommets(listeSommets(G))  
    for i in range(n):  
        if degre(i) % 2 == 0:  
            colorierSommet(sommetNumero(G,i))
```

Indiquez et expliquez les erreurs.

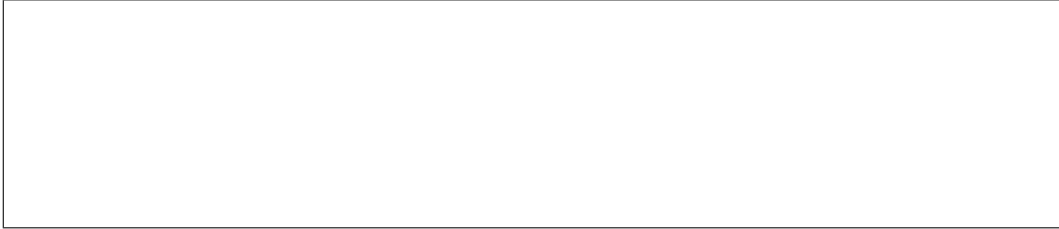


Donnez une version (correcte) bien plus simple de la fonction `colorierPairs` :



**Exercice 5** Un graphe  $G$  est dit *triquadripode* s'il existe un entier  $p$  tel que  $G$  comporte  $2p$  sommets, parmi lesquels  $p$  d'entre eux sont de degré 3 et les  $p$  autres sont de degré 4. Autrement dit,  $G$  a autant de sommets de degré 3 que de sommets de degré 4 et ne possède pas de sommets de degré autre que 3 ou 4.

1. Dessiner un graphe triquadripode à 4 sommets.



2. Soit  $n$  un entier multiple de 4 ( $n = 4p$ ), indiquer comment construire un graphe triquadripode à  $n$  sommets.



3. Exprimer le nombre d'arêtes en fonction de  $n$  dans un graphe triquadripode à  $n$  sommets.



4. En déduire que si  $n$  est un entier non multiple de 4, il n'existe pas de graphe triquadripode à  $n$  sommets.




Nom :

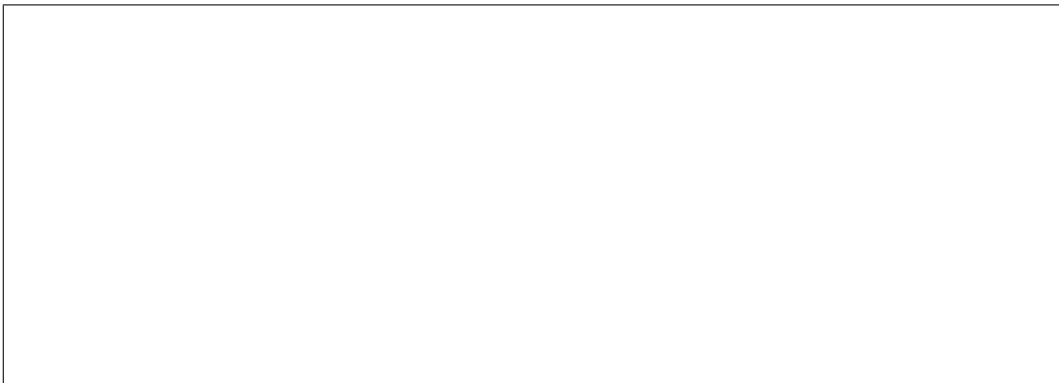
Prénom :

Groupe :

5. Existe-t-il un graphe triquadripode simple à 4 sommets ? Justifier.



6. Question plus difficile : Si  $n$  est un entier multiple de 4, peut-on toujours construire un graphe triquadripode connexe à  $n$  sommets ?







*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.