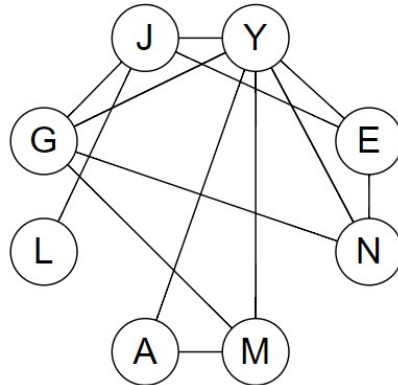


Exercice 2 : Partie A : Matrice d'adjacence

1. Graphe :



2.

```
# sommets :      G, J, Y, E, N, M, A, L
matrice_adj = [[0, 1, 1, 0, 1, 1, 0, 0], # G
               [1, 0, 1, 1, 0, 0, 0, 1], # J
               [1, 1, 0, 1, 1, 1, 1, 0], # Y
               [0, 1, 1, 0, 1, 0, 0, 0], # E
               [1, 0, 1, 1, 0, 0, 0, 0], # N
               [1, 0, 1, 0, 0, 0, 1, 0], # M
               [0, 0, 1, 0, 0, 1, 0, 0], # A
               [0, 1, 0, 0, 0, 0, 0, 0]] # L
```

3. Avec sommets = ['G', 'J', 'Y', 'E', 'N', 'M', 'A', 'L']:

position(sommets, 'G') renvoie 0

position(sommets, 'Z') renvoie None

4.

```
def nb_amis(L, m, s):
    pos_s = position(L, s)
    if pos_s == None:
        return None
    amis = 0
    for i in range(len(m)):
        amis += m[i][pos_s]
    return amis
```

5. nb_amis(sommets, matrice_adj, 'G') renvoie 4.

Exercice 2 : Partie B : Dictionnaire de listes d'adjacence

6. Dans un dictionnaire, **c** représente une clé et **v** la valeur qui lui est associée.

7.

```
graphe = {'G' : ['J', 'Y', 'N', 'M'],  
          'J' : ['G', 'Y', 'E', 'L'],  
          'Y' : ['G', 'J', 'E', 'N', 'M', 'A'],  
          'E' : ['J', 'Y', 'N'],  
          'N' : ['G', 'Y', 'E'],  
          'M' : ['G', 'Y', 'A'],  
          'A' : ['Y', 'M'],  
          'L' : ['J']  
        }
```

8.

```
def nb_amis(d, s):  
    return len(d[s])
```

9. Le cercle d'amis de Lou est Jade, Gabriel, Nino, Yanis et Emma.

10.

```
def parcours_en_profondeur(d, s, visites = []):  
    visites += [s]  
    for v in d[s]:  
        if v not in visites:  
            parcours_en_profondeur(d, v)  
    return visites
```