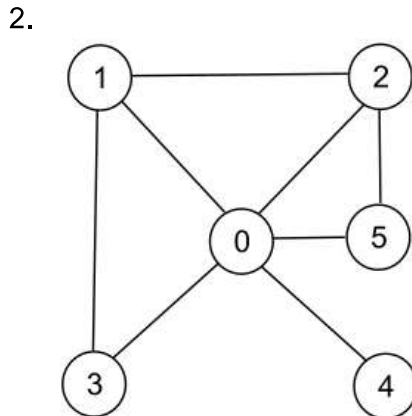


Exercice 1

1.  
voisins = [[1, 2, 3, 4],  
          [0, 2, 3],  
          [0, 1],  
          [0, 1],  
          [0]]



3.  
voisins = [[1, 2, 3, 4, 5],  
          [0, 2, 3],  
          [0, 1, 5],  
          [0, 1],  
          [0],  
          [0, 2]]

4.  
def voisin\_alea (voisins, s) :  
    return voisins[s][random.randrange(len(voisins[s]))]

5.  
La fonction marche\_alea s'appelle elle-même, c'est donc une fonction récursive.

6.  
Cette fonction permet de parcourir une chaîne dans le graphe voisins, cette chaîne commence au sommet i et comporte n sommets. La fonction renvoie le dernier sommet de cette chaîne.

7.  
def simule (voisins, i, n\_tests, n\_pas) :  
    results = [0] \* len (voisins)  
    while n\_tests > 0 :  
        j = marche\_alea(voisins, i, n\_pas)  
        results[j] = results[j] + 1  
        n\_tests = n\_tests - 1  
    return [s/n\_tests for s in results]

8.  
C'est l'ordinateur 0 car dans 32,8% des cas, le virus termine sa course sur l'ordinateur 0.

9.  
Il suffit d'effectuer un parcours en largeur du graphe et de compter le nombre d'itérations nécessaires pour visiter tous les nœuds du graphe.

## Exercice 2

### Partie A

1. 255.255.0.0
2. 172.16.0.0
3. 172.16.255.255
4.  $2^{16} - 2$

### Partie B

5. S1 -> A -> H -> D -> S2
6. S1 -> A -> H -> C -> D -> S2 ou S1 -> A -> B -> C -> D -> S2 ;
- 7.

Pour AHCD :

| Routeur | Réseau destinataire | Passerelle  | Interface  |
|---------|---------------------|-------------|------------|
| H       | L2                  | 53.10.10.10 | 53.10.10.9 |

Pour ABCD :

| Routeur | Réseau destinataire | Passerelle  | Interface   |
|---------|---------------------|-------------|-------------|
| A       | L2                  | 193.55.24.1 | 193.55.24.2 |

8.  
100 Mbit/s =  $10^8$  bit/s soit un coût = 10  
1 Gbit/s =  $10^9$  bit/s soit un coût = 1  
10 Gbit/s =  $10^{10}$  bit/s soit un coût = 0,1
9.  
S1 -> A -> G -> F -> E -> D -> S2 avec un coût =  $1 + 0,1 + 0,1 + 0,1 = 1,3$
10.  
S1 -> A -> H -> F -> E -> D -> S2 avec un coût =  $1 + 1 + 0,1 + 0,1 = 2,2$

## Exercice 3

### Partie A

1.
  - les bases de données relationnelles permettent de gérer la lecture, l'écriture ou la modification des informations contenues dans une base de données (plus difficile avec un fichier texte)
  - les bases de données relationnelles permettent de gérer les autorisations d'accès.
2.  
L'attribut doit avoir une valeur unique pour chaque entrée
3. id\_client et id\_emplacement sont des clés étrangères : id\_client permet de relier (jointure) la table Reservation et la table Client, id\_emplacement permet de relier (jointure) la table Reservation et la table Emplacement.
4.  
Emplacement (id\_emplacement, nom, localisation, tarif\_journalier)

5.

|   |           |    |
|---|-----------|----|
| 1 | myrtille  | A4 |
| 4 | mandarine | B1 |
| 6 | melon     | A2 |

6.

```
SELECT nom, prenom
FROM Client
WHERE ville = 'Strasbourg'
```

7.

```
INSERT INTO Client
VALUES
(42, 'Codd', 'Edgar', '28 rue des Capucines', 'Lyon', 'France', '0555555555')
```

8.

```
SELECT Client.nom, Client.prenom, Reservation.nombre_personne,
Reservation.date_arrivee, Reservation.date_depart,
Emplacement.tarif_journalier
FROM Reservation
JOIN Client ON Client.id_client = Reservation.id_client
JOIN Emplacement ON Emplacement.id_emplacement =
Reservation.id_emplacement
WHERE Reservation.id_reservation = 18
```

## Partie B

9.

self correspond à l'instance courante, on passe cette instance courante en paramètres aux méthodes de la classe.

10.

```
client01 = Client('Codd', 'Edgar', '28 rue des Capucines', 'Lyon',
'France', '0555555555')
```

11.

```
return reservation.nb_jours() * emplacement.tarif_journalier + 2.20
* reservation.nb_jours() * reservation.nombre_personne
```

12.

La variable annee est de type chaîne de caractère (str). A ligne 25, on cherche à comparer la variable annee avec les entiers 2018 et 2024. Il n'est pas possible de comparer un entier et une chaîne de caractères d'où l'erreur.

13.

Il est possible d'écrire à ligne 25 : `2018 <= int(annee) <= 2024`

14.

ligne 28 :

```
if mois not in calendrier :
    return False
```

ligne 30 :

```
if len(numero) != 3 or not que_des_chiffres(numero):
    return False
```