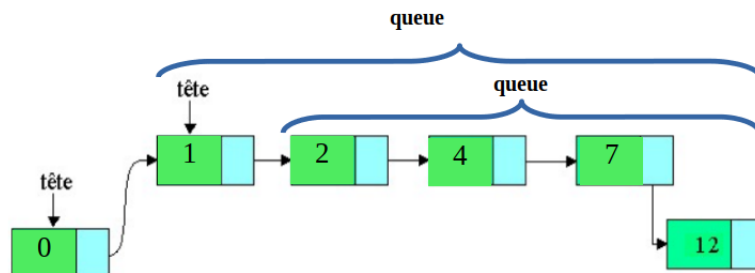


Exercice 4

POO - Listes

Le principe de la liste chaînée

Le schéma ci-dessous illustre le principe d'une liste chaînée : chaque cellule est constitué d'une valeur (la tête) et d'une queue qui fait référence à la cellule suivante.



On peut représenter la liste ci-dessus par : [0]->[1]->[2]->[4]->[7]->[12]

On a choisit de représenter une liste à l'aide des deux classes ci-dessous :

```
class Cellule:
    def __init__(self, tete = None, queue = None):
        self.tete = tete
        self.queue = queue

class Liste:
    def __init__(self, c = None):
        self.cellule = c

    # renvoie True si la liste est vide
    def est_vide(self):
        return self.cellule is None
```

Le but de cet exercice est de créer une fonction qui permet d'insérer une valeur dans la liste au bon endroit de sorte que la liste soit ordonnée dans le sens croissant.

? QUESTION 1: Création de listes

1. On souhaite créer une liste vide `liste1`. Quelle instruction faut-il écrire pour cela?
2. Pour créer la première cellule de la liste, il suffit d'exécuter cette instruction :

```
liste1.cellule = Cellule(0, None)
```

Recopier et compléter la ligne de code ci-dessous qui permet d'obtenir la liste: [0]->[1]->[2]->[4]->[7]->[12]

```
liste1.cellule.queue =
```

? QUESTION 2: Accès aux valeurs de la liste

On suppose créée la liste1 : [0]->[1]->[2]->[4]->[7]->[12]

L'instruction `liste1.cellule` permet d'accéder à la première cellule de la liste.

1. Quel affichage produit cette instruction:

```
print(liste1.cellule.tete)
```

2. Expliquer pourquoi l'instruction `print(liste1.cellule.queue)` produit l'affichage `<__main__.Cellule object >?`

3. Quel affichage produit l'instruction : `print(liste1.cellule.queue.tete)?`

4. Écrire l'instruction permettant de faire afficher la valeur 7 de liste1.

? QUESTION 3: Parcours de liste

On suppose créée la liste1 : [0]->[1]->[2]->[4]->[7]->[12]

1. Que contient la variable `tmp` à l'issue de la boucle `while` de ce programme?

```
tmp = liste1.cellule # on accède à la première cellule
while tmp is not None:
    tmp = tmp.queue
```

2. Que contient `tmp.tete` à l'issue de la boucle `while` de ce programme?

```
tmp = liste1.cellule # on accède à la première cellule
while tmp.queue is not None:
    tmp = tmp.queue
```

La fonction ci-dessous est sensée réaliser l'insertion souhaitée.

```
def insert(liste, val):
    if liste.est_vide():
        liste.cellule = ???
    elif liste.cellule.tete >= val:
        liste.cellule = Cellule(val, liste.cellule)
    else:
        tmp=liste.cellule
        while tmp.tete <= val and tmp.queue is not None:
            tmp1 = tmp
            tmp = ???
        if tmp.tete >= val:
            tmp1.queue = ???
        else:
            tmp.queue = ???
```

? QUESTION 4: La fonction insertion

1. Recopier le programme ci-dessus en remplaçant les ??? par les bonnes instructions.

2. Écrire la fonction `insert` en tant que méthode de la classe `Liste`.