

Bac Blanc

Nom et prénom :

Consignes générales

Ce sujet les composé de 3 parties indépendantes valant 15 points chacune. Dans chaque partie, il est possible de sauter une question et des répondre aux suivantes.

Répondre aux questions dans les zones prévues à cet effet. La calculatrice est interdite.

Structures de données (15pt)

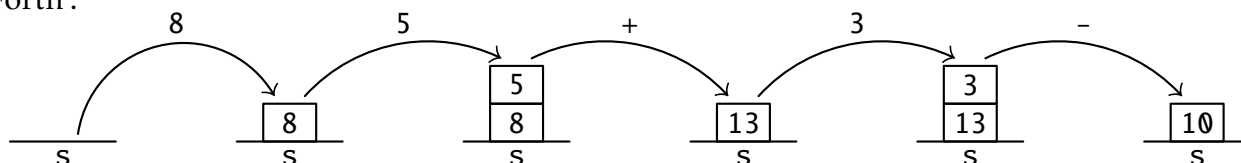
Le langage Forth a été créé au début des années 1960 par Chuck Moore (né en 1938, photo ci-jointe). C'est un langage important du point de vue de l'histoire de l'informatique, d'une part parce qu'il a été développé pour l'informatique embarquée (suivi de satellites, guidage de télescopes, ...), d'autre part parce qu'il est basé sur la structure de pile (stack).



Forth s'inscrit dans le paradigme de programmation impérative et c'est un langage interprété (il n'a pas de compilateur). Un interpréteur Forth occupe typiquement moins de 10 ko...

La pile de Forth est utilisée pour évaluer des expressions algébriques comme : 8 5 + 3 - écrites en notation polonaise inversée (RPN).

L'expression 8 5 + 3 - s'évalue à 10 comme le montrent les états successifs de la pile de Forth :



- On empile la valeur 8.
- On empile la valeur 5.
- On remplace les deux nombres du sommet de la pile (8 et 5) par leur somme 13.
- On empile la valeur 3.
- On remplace les deux nombres du sommet de la pile (13 et 3) par leur différence 10.

1) Que donne l'évaluation par Forth de l'expression 2 3 × 5 +? Vous pourrez représenter les différents états de la pile sur votre copie.

L'interpréteur de Forth est basé sur une pile (stack) dont l'interface, inventée par Alan Turing, est donnée ci-dessous :

Fonction	Description
create_stack()	Renvoie une nouvelle pile vide.
push(s, e)	Rajoute e au sommet de la pile s.
pop(s)	Renvoie l'élément se trouvant au sommet de s, qui ne doit pas être vide, et l'enlève.
empty(s)	Renvoie un booléen indiquant si s est vide ou non.

2) Le mode LIFO permet de décrire la structure d'une pile. Que signifie cet acronyme?

3) Que signifie l'acronyme FIFO et à quelle structure de données associe-t-on ce mode?

La pile de Forth est plus riche que celle de Turing : outre push, pop et empty, elle possède les instructions suivantes :

Fonction	Description	Exemple								
top(s)	Renvoie l'élément au sommet de la pile s sans la modifier.	<table style="display: inline-table; border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">3</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">5</td><td style="border: 1px solid black; padding: 2px;">5</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">s</td><td style="border: 1px solid black; padding: 2px;">s</td></tr> </table>	2	2	3	3	5	5	s	s
2	2									
3	3									
5	5									
s	s									
drop(s)	Supprime l'élément au sommet de la pile s.	<table style="display: inline-table; border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">3</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">5</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">5</td><td style="border: 1px solid black; padding: 2px;">s</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">s</td><td style="border: 1px solid black; padding: 2px;">s</td></tr> </table>	2	3	3	5	5	s	s	s
2	3									
3	5									
5	s									
s	s									
dup(s)	Duplique de la pile s.	<table style="display: inline-table; border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">3</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">5</td><td style="border: 1px solid black; padding: 2px;">5</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">s</td><td style="border: 1px solid black; padding: 2px;">s</td></tr> </table>	2	2	3	3	5	5	s	s
2	2									
3	3									
5	5									
s	s									
swap(s)	Échange le contenu des deux cellules du haut de la pile s.	<table style="display: inline-table; border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">3</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">5</td><td style="border: 1px solid black; padding: 2px;">5</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">s</td><td style="border: 1px solid black; padding: 2px;">s</td></tr> </table>	2	3	3	2	5	5	s	s
2	3									
3	2									
5	5									
s	s									
over(s)	Duplique l'avant-dernier élément de la pile et place la copie en haut de la pile s.	<table style="display: inline-table; border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">3</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">5</td><td style="border: 1px solid black; padding: 2px;">5</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">s</td><td style="border: 1px solid black; padding: 2px;">s</td></tr> </table>	2	3	3	2	5	5	s	s
2	3									
3	2									
5	5									
s	s									
rot(s)	Déplace l'avant-avant-dernier élément de la pile pour le mettre au sommet de s.	<table style="display: inline-table; border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">5</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">5</td><td style="border: 1px solid black; padding: 2px;">3</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">s</td><td style="border: 1px solid black; padding: 2px;">s</td></tr> </table>	2	5	3	2	5	3	s	s
2	5									
3	2									
5	3									
s	s									

En utilisant uniquement les fonctions empty, push et pop on peut écrire toutes les autres fonctions.

Par exemple :

```
def top(s):
    x = pop(s)
    push(s, x)
    return x
```

```
def drop(s):
    x = pop(s)
```

```
def dup(s):
    x = pop(s)
    push(s, x)
    push(s, x)
```

Nom et prénom :

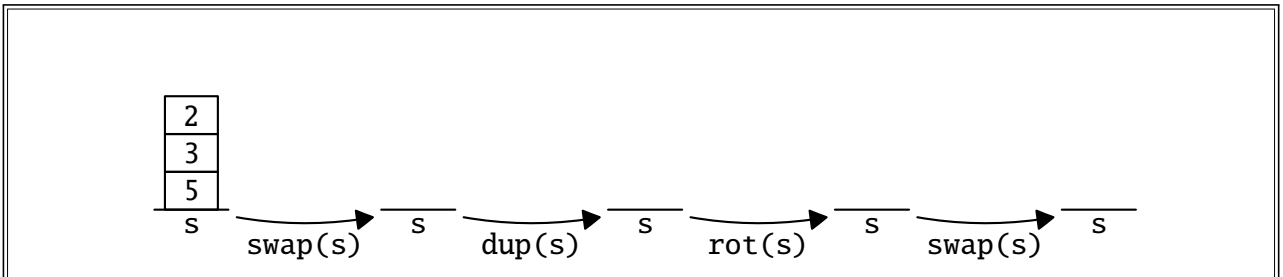
3) Écrire les fonctions swap, over et rot en utilisant uniquement les fonctions empty, push et pop.

<pre>def swap(s):</pre>	<pre>def over(s):</pre>	<pre>def rot(s):</pre>
-------------------------	-------------------------	------------------------

On considère la fonction suivante :

```
def mystere(s):
    swap(s)
    dup(s)
    rot(s)
    swap(s)
```

4) Indiquer les états successifs de la pile ci-dessous lors de l'exécution de cette séquence.



5) Quelle instruction de Forth permet de faire la même chose que cette fonction?

6) L'utilisation de swap et rot permettent d'obtenir différentes configurations pour une pile de 3 éléments. Pour chaque suite d'instructions, représenter l'état final de la pile.

Instructions	Pile	Instructions	Pile																
swap(s) rot(s)	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">5</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">s</td><td style="padding: 2px 5px;">s</td></tr> </table>	2		3		5		s	s	swap(s) swap(s)	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">5</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">s</td><td style="padding: 2px 5px;">s</td></tr> </table>	2		3		5		s	s
2																			
3																			
5																			
s	s																		
2																			
3																			
5																			
s	s																		
rot(s) swap(s)	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">5</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">s</td><td style="padding: 2px 5px;">s</td></tr> </table>	2		3		5		s	s	rot(s) rot(s)	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">5</td><td style="padding: 2px 5px;"> </td></tr> <tr><td style="padding: 2px 5px;">s</td><td style="padding: 2px 5px;">s</td></tr> </table>	2		3		5		s	s
2																			
3																			
5																			
s	s																		
2																			
3																			
5																			
s	s																		

Bases de données (15pt)

Afin de lancer un nouveau service de streaming de musique, vous devez construire une base de données pour les morceaux de votre catalogue. Pour l'instant vous disposez d'une seule table avec les informations des morceaux. Voici un extrait de cette table :

Titre	Durée	Artiste	Album	Piste	CD	Année
Astronomy	384	Blue Öyster Cult	Secret Treaties	8	1	1974
Stone Cold Crazy	136	Queen	Sheer Heart Attack	8	1	1974
Under Pressure	242	Queen and David Bowie	Hot Space	11	2	1982
The Outlaw Torn	589	Metallica	Load	14	1	1996
Fuel	270	Metallica	Reload	1	1	1997
The Memory Remains	279	Metallica and Marianne Faithfull	Reload	2	1	1997
Astronomy	398	Metallica	Garage Inc.	8	1	1998
Stone Cold Crazy	139	Metallica	Garage Inc.	11	2	1998
Fuel	276	Metallica and the San Francisco Symphony	S&M	6	1	1999
The Outlaw Torn	599	Metallica and the San Francisco Symphony	S&M	6	2	1999

Cette table ne convient pas vraiment pour faire une base de données.

1) Expliquer pourquoi aucune des colonnes ne peut pas servir de clef primaire.

2) Pourquoi est-ce que cette table est problématique si on veut rajouter des informations sur les artistes, comme leur nationalité?

3) Quel est le problème si on souhaite chercher les morceaux d'un artiste? Vous pourrez prendre l'exemple de Metallica.

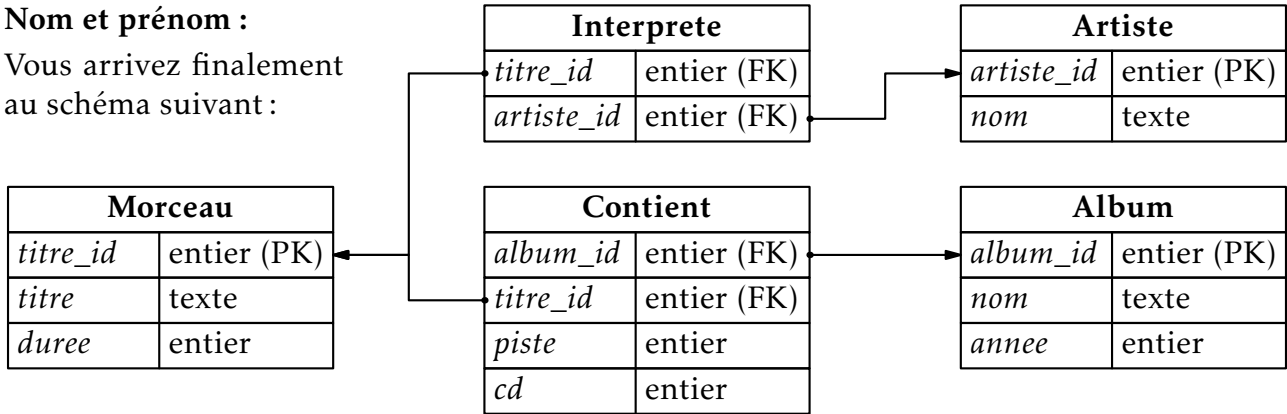
Un ami vous suggère d'utiliser le schéma suivant :

- **Morceau**(titre_id, titre, durée, artiste_id, album, piste, cd, année)
- **Artiste**(artiste_id, nom)

4) Expliquer pourquoi cette représentation ne permet toujours pas de gérer les morceaux fait par deux artistes différents.

Nom et prénom :

Vous arrivez finalement au schéma suivant :



5) Compléter les tables à l'aide des informations déjà disponibles. Un des morceaux n'a pas été remis. Si les noms dépassent, mettre uniquement le début.

titre_id	titre	duree	titre_id	artiste_id	artiste_id	nom
519	Astronomy		519	25		
1219	Astronomy		1219	154		
316	Stone Cold Crazy	136	1319			
1319	Stone Cold Crazy	139	1298	154		
1298		270	1570	154		
1570			1570	318		
401			1125	154		
1125	The Outlaw Torn	589	1591	154		
1591	The Outlaw Torn	599	1591	318		
			316	79		
			401	79		
			401	108		

5) Comment appelle-t-on les clefs primaires de certaines tables apparaissant dans certaines tables, comme dans **Interprete**?

6) Expliquer pourquoi le couple (titre_id, artiste_id) peut servir de clef primaire à **Interprete**.

7) Traduire en langage naturel les requêtes suivantes :

```

-- Requete 1
SELECT titre, duree FROM Morceau WHERE duree > 600 ORDER BY duree DESC;
-- Requete 2
SELECT cd, piste, titre FROM Morceau
JOIN Contient ON Contient.titre_id = Morceau.titre_id
JOIN Album ON Contient.album_id = Album.album_id
WHERE nom = "Garage Inc."
ORDER BY cd, piste;

```

8) Donner les requêtes SQL permettant d'obtenir les résultats suivants :

a) Le nom de l'artiste dont l'identifiant est 200.

b) Le nom de tous les albums sortis entre 1999 et 2010.

c) Le titre et la durée, triés par ordre décroissant de durée, de tous les morceaux de l'artiste dont l'identifiant est 200.

9) Les stars étant capricieuses, certaines veulent changer de nom. Donner la requête permettant à 'Maître Gims' de devenir 'Gims' dans la table des artistes.

On rajoute maintenant les tables pour les utilisateurs ;

- **Utilisateur**(util_id, nom, e-mail, adresse)
- **Ecoute**(id_ecoute, titre_id, util_id, date)

12) Expliquer pourquoi le couple (titre_id ,util_id) ne peut pas être une clef primaire.

13) Donner la requête SQL permettant d'ajouter l'utilisateur numéro 2179, qui s'appelle Bob VHS, dont l'email est bob.vhs@hotmail.com et qui habite à New York.

14) Traduire le requête suivante en langage naturel :

```
SELECT COUNT(DISTINCT titre) FROM Morceau
JOIN Ecoute ON Morceau.titre_id = Ecoute.titre_id
WHERE date = "2020-12-12";
```

Programmation orientée objet (15pt)

Dans cette partie, nous voulons réaliser une liste chaînée ordonnée d'entiers. Lorsqu'on ajoute un élément, il est placé de telle sorte à maintenir l'ordre.

Dans le cas d'égalité, on placera la nouvelle valeur le plus tôt possible.

On a déjà les classes suivantes :

```
class Cellule:
    def __init__(self, v, s):
        self.valeur = v
        self.suivant = s

class ListeOrdo:
    def __init__(self):
        self.cellule = None

    def est_vide(self):
        return self.cellule is None

    def ajouter(self, v):
        pass

    def __str__(self):
        if self.est_vide():
            return "nil"
        else:
            c = self.cellule
            res = str(c.valeur)
            while c.suivant is not None:
                c = c.suivant
                res += " -> " + str(c.valeur)
            return res
```

- 1) Comment est le nom particulier des méthodes `__init__` présentes dans les deux classes? À quoi servent-elle?

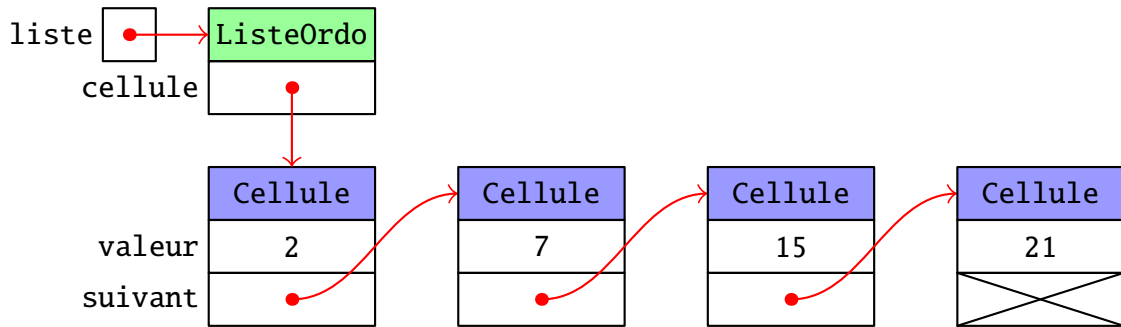
- 2) Comment s'appellent valeur et suivant dans la classe Cellule?

- 3) Quelle instruction permet de créer une nouvelle liste ordonnée `liste0`?

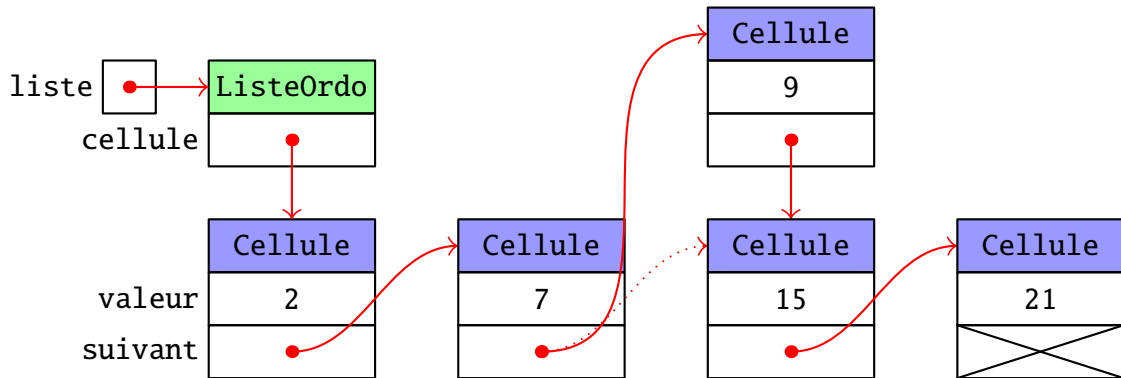
liste0 =

- 4) Juste après sa création, combien de valeurs sont stockées dans une nouvelle liste?

L'objectif est de rajouter le code de la méthode ajouter(**self**, v) qui rajoute la valeur v à la bonne place, tout en maintenant l'ordre. Voici un exemple de liste :



Et voici le nouvel état de la liste après l'appel de liste.ajouter(9) :



Pour les questions suivantes, on considère la valeur finale de liste.

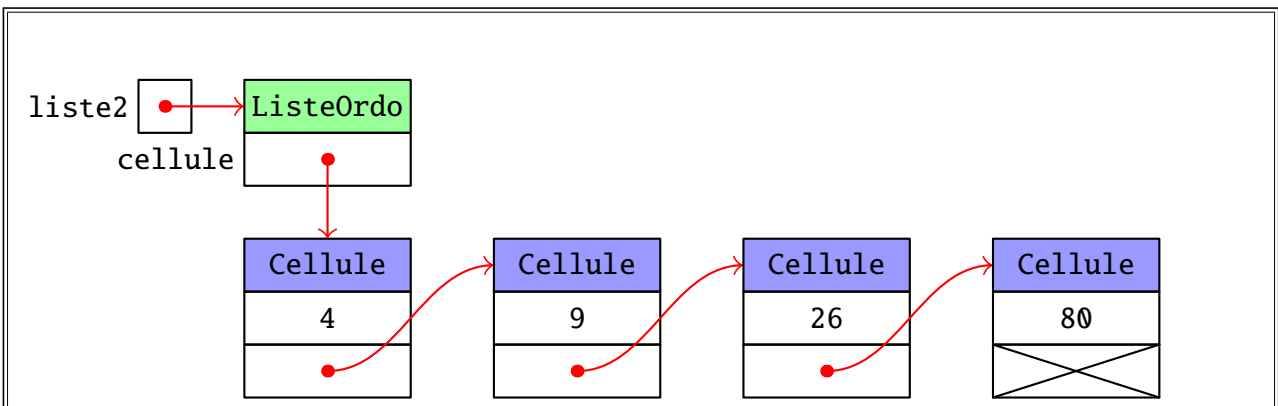
6) Indiquer les réponses données par l'interpréteur Python pour les commandes suivantes :

```
>>> str(liste)

>>> liste.cellule.valeur

>>> liste.cellule.suivant.valeur
```

7) On considère maintenant la liste liste2.



Compléter le schéma ci-dessus après l'exécution des instructions suivantes. Vous dessinerez les nouvelles cellules en mettant les nouvelles flèches nécessaires, tout en barrant celles qui sont modifiées.

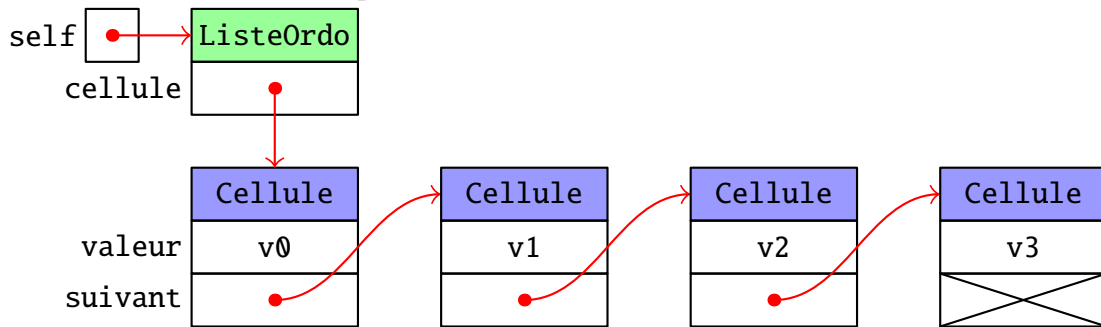
```
>>> liste2.ajouter(15);
>>> liste2.ajouter(3);
>>> liste2.ajouter(99);
```


Nous allons maintenant considérer les différents cas pour l'insertion des éléments. On appellera **self** la liste considérée.

Par exemple, si **self.est_vide()** est vrai, alors il faut faire :

```
self.cellule = Cellule(v, None)
```

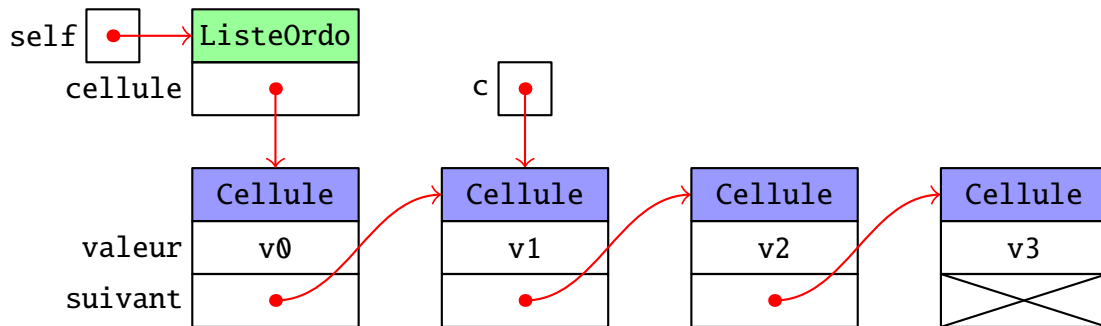
On suppose maintenant que la liste n'est pas vide. Voici une représentation de **self**. Les valeurs **v0** à **v3** sont là pour représenter les valeurs stockées dans la liste mais ne doivent jamais être utilisées dans vos réponses.



8) On suppose que $v \leq \text{self.cellule.valeur}$. Compléter l'instruction pour ajouter **v** :

```
self.cellule =
```

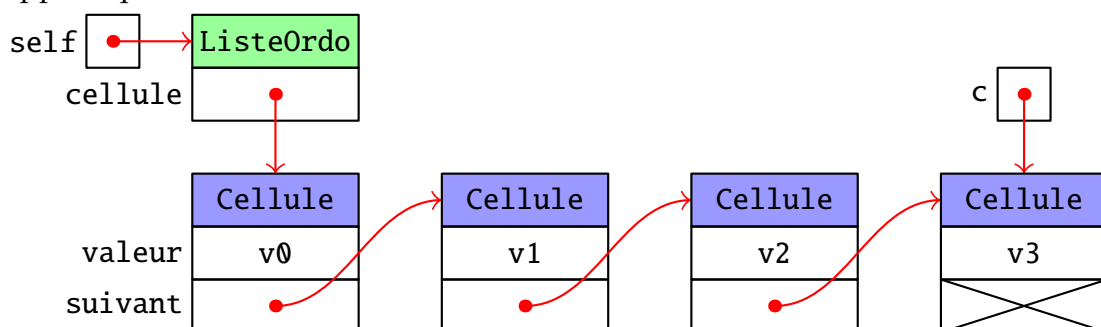
9) On est maintenant en train de parcourir la liste et **c** correspond à la cellule considérée. L'insertion n'a pas pu se faire avant et donc $c.valeur < v$.



a) Quel test permet de savoir s'il faut insérer la valeur juste après **c** ?

b) Si **v** doit être ajouté juste après **c**, donner la ou les instructions pour ajouter **v** :

10) On suppose qu'on est arrivé au bout de la liste.



a) Qu'est-ce que cela veut dire pour v par rapport aux autres valeurs de la liste?

b) Quel test permet de savoir que c est la dernière cellule de la liste?

c) Donner la ou les instructions qui permettent d'ajouter v après c .

Nous avons maintenant tous les éléments permettant de faire la méthode `ajouter(self, v)`.

14) Écrire le code de la fonction `ajouter(self, v)`.

```
def ajouter(self, v):
```

15) Écrire le code de la fonction `compter(self, v)` qui renvoie le nombre d'apparitions de v dans la liste `self`. On exploitera le fait que la liste est ordonnée pour ne pas parcourir tous les éléments lorsque ce n'est pas nécessaire. Pour ceux qui souhaitent faire une version récursive, il est possible d'utiliser `tete(self)` qui renvoie la valeur en tête et `queue(self)` qui renvoie une nouvelle liste pointant sur la queue de la liste.

```
def compter(self, v):
```