EXERCICE 3 (8 points)

Cet exercice porte sur les bases de numération, la structure de données PILE et la POO.

La civilisation *Maya* est une ancienne civilisation de Mésoamérique principalement connue pour ses avancées dans les domaines de l'écriture, de l'art, de l'architecture, de l'agriculture, des mathématiques et de l'astronomie.

La numération *Maya* est une numération positionnelle de base 20 (dite vigésimale) utilisant trois symboles pour former les “chiffres” :

- une coquille pour le zéro  ,
- un point pour l’unité ●,
- un trait pour la valeur 5 ▬.

Les “chiffres” sont les suivants. Ils utilisent une numération additive :

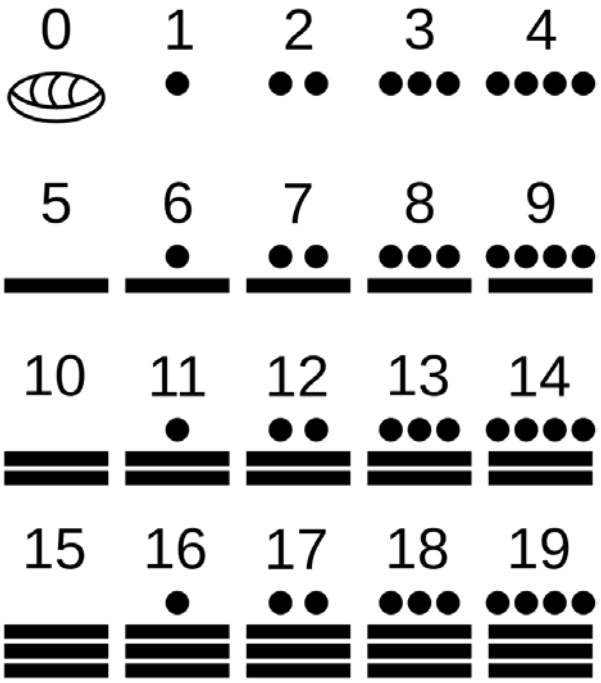
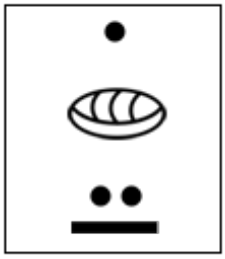


Figure 1. Table des “chiffres” et valeurs correspondantes (source : Wikipédia)

Dans une version simplifiée de ce système, l’écriture d’un nombre se fait par empilement de “chiffres”. Chaque étage correspond à un chiffre de poids 20 fois supérieur au poids du chiffre de l’étage inférieur.

Ainsi la valeur du chiffre de l’étage le plus bas est multipliée par 20^0 soit 1, du second étage par 20^1, du troisième étage par 20^2, et ainsi de suite.

Exemple : la représentation *Maya* de l'entier 407 est la suivante.



1. Compléter le tableau donné en **annexe à rendre avec la copie**.

2. Justifier que l'écriture *Maya* de l'entier 3435 est :

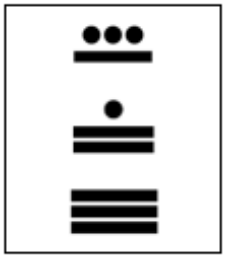


Figure 2. Ecriture *Maya* de l'entier 3435

On **modélise l'écriture d'un entier** dans sa représentation *Maya* par une pile formée de listes. Chacune de ces listes est composée de trois entiers et modélise le "chiffre" d'un étage :

- le premier entier vaut 0 ou 1 suivant s'il y a ou non une coquille ;
- le deuxième représente le nombre de points ;
- le troisième représente le nombre de traits.

Ainsi, la modélisation *Maya* de l'entier 3435 est `[[0, 0, 3], [0, 1, 2], [0, 3, 1]]` et celle de l'entier 407 est `[[0, 2, 1], [1, 0, 0], [0, 1, 0]]`.

Le sommet de la pile se situe en fin de liste.

On dispose de la classe suivante :

```
class Maya:
    def __init__(self):
        self.nombre = []

    def ajouter(self, chiffre):
        """ chiffre est une liste de longueur 3.
        La méthode empile le chiffre au sommet de la pile """
        self.nombre.append(chiffre)

    def retirer(self):
        """ depile et renvoie le chiffre qui etait au sommet de

          la pile """
        if not self.estVide():
            return self.nombre.pop()

    def estVide(self):
        return self.nombre == []

    def nbEtages(self) :
        """ renvoie le nombre de chiffres de la pile """
        ...

    def MayaToDec(self):
        """ renvoie le nombre entier correspondant a la
        modelisation Maya de l'instance courante """
        coeff = 20**...
        ch_Dec = 0
        while ... :
            ch_Maya = ...
            ch_Dec = ch_Dec + (valeurChiffre(ch_Maya)) * coeff
            coeff = ...
        return ch_Dec

    def multiplie(self):
        """ renvoie le resultat de la multiplication par 20 d'un
        nombre en modelisation Maya. """
        ...

    def somme(self, maya2):
        """ ajoute maya2 à l'instance courante et renvoie le
        resultat en modelisation Maya """
        if self.nbEtages() == maya2.nbEtages()
        ...
```

3. Écrire une suite d'instructions permettant de créer une instance, nommée `M`, de la classe `Maya` qui modélise le nombre entier 3435.

4. Écrire la méthode `nbEtages` de la classe `Maya`. Celle-ci renvoie le nombre de "chiffres" utilisés pour écrire le nombre correspondant en écriture *Maya*.

De l'écriture *Maya* à l'écriture décimale

5. Écrire une fonction `valeurChiffre` ayant pour paramètre une liste `L`. Celle-ci renvoie la valeur de l'entier associé à la liste `L = [c, p, t]` où `c` (de valeur 0 ou 1) indique la présence d'une coquille, `p` est le nombre de points et `t` le nombre de traits composant un "chiffre" *Maya*.

Exemple :

```
>>> valeurChiffre([0, 2, 3])
>>> 17
>>> valeurChiffre([1, 0, 0])
>>> 0
```

6. Recopier et compléter les lignes 2, 4, 5 et 7 de la méthode `MayaToDec` suivante de la classe `Maya`. Cette méthode renvoie la valeur de l'entier associé à l'objet `Maya`. On pourra utiliser les méthodes `estVide`, `nbEtages` et `retirer`.

```
1 def MayaToDec(self):
2     coeff = 20**...
3     ch_Dec = 0
4     while ... :
5         ch_Maya = ...
6         ch_Dec = ch_Dec + (valeurChiffre(ch_Maya)) * coeff
7         coeff = ...
8     return ch_Dec
```

De l'écriture décimale vers sa modélisation *Maya*

On considère que la fonction `DecToVige` est déjà écrite. Celle-ci prend en paramètre un entier `n` et renvoie la décomposition en base 20 de celui-ci sous la forme d'une liste $[\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_p]$ telle que :

$$n = \mathbf{a}_0 \times 20^0 + \mathbf{a}_1 \times 20^1 + \mathbf{a}_2 \times 20^2 + \ \ldots + \mathbf{a}_p \times 20^p$$

Exemple :

```
>>> DecToVige(3435)
[15, 11, 8]
>>> DecToVige(407)
[7, 0, 1]
```

7. Écrire la fonction `decompChiffre` qui prend en paramètre un entier `n` compris entre 0 et 19 et renvoie la liste `[c, p, t]` où `c` vaut 0 ou 1 et indique la présence ou non d'une coquille, `p` est le nombre de points et `t` le nombre de traits composant le "chiffre" *Maya* correspondant.

Exemple :

```
>>>decompChiffre(17)
[0, 2, 3]
```

```
>>>decompChiffre(0)
[1, 0, 0]
```

8. Écrire la fonction `DecToMaya` qui prend en paramètre un entier `n` et renvoie la modélisation *Maya* d'un objet M de la classe `Maya` correspondant.

Exemple :

```
>>>DecToMaya(3435).nombre
[[0, 0, 3], [0, 1, 2], [0, 3, 1]]
>>>DecToMaya(407).nombre
[[0, 2, 1], [1, 0, 0], [0, 1, 0]]
```

Opérations sur les nombres en modélisation *Maya*

On souhaite additionner des nombres directement à partir de leur modélisation *Maya*.

9. Écrire la méthode `multiplie` de la classe `Maya` qui renvoie le résultat de la multiplication par 20 d'un nombre en modélisation *Maya*.

Exemple :

```
>>> M = Maya()
>>> M.ajouter([0 ,0, 3])
>>> M.ajouter([0, 1, 2])
>>> M.multiplie().nombre
[[1, 0, 0], [0, 0, 3], [0, 1, 2]]
```

On donne la fonction `mystere` suivante :

```
1 def mystere(m1, m2, ret):
2     c = 0
3     p = (m1[1] + m2[1] + ret)%5
4         if m1[1] + m2[1] + ret >= 5:
5             ret = 1
6         else:
7             ret = 0
8     t = (m1[2] + m2[2] + ret)%4
9     if m1[2] + m2[2] + ret < 4:
10        ret = 0
11    else:
12        ret = 1
13    if (m1[0] == 0 and m2[0] == 1) or (p + t = 0 and ret == 1):
14        c = 1
15    return ([c, p, t], ret)
```

10. Donner les résultats renvoyés par les deux appels suivants :

- `mystere([0, 1, 1], [0, 3, 1], 0)`
- `mystere([0, 1, 1], [0, 4, 2], 0)`

11. Écrire une méthode `somme` de la classe `Maya` permettant d'ajouter à l'instance courante un autre nombre `maya2` de même taille en modélisation Maya.

ANNEXE À RENDRE AVEC LA COPIE

Exercice 3 – Question 1.

Étage	Écriture *Maya*	Valeur du “chiffre” de l’étage	Valeur dans la conversion
3		$1 \times 5 + 3 \times 1 = 8$	$8 \times 20^2 = 3200$
2			
1