

**Durée du CC : 1h15**

Prénom ..... Nom : .....

Les réponses sont à apporter directement sur la feuille, en dessous de la question correspondante. Le bénéfice du doute ne sera pas accordé en cas de réponse illisible.

**Si un appareil électronique est manipulé durant le CC, un dossier sera envoyé au conseil disciplinaire, peu importe la justification donnée. Toute sortie est définitive. La fiche-triche est le seul document autorisé.**

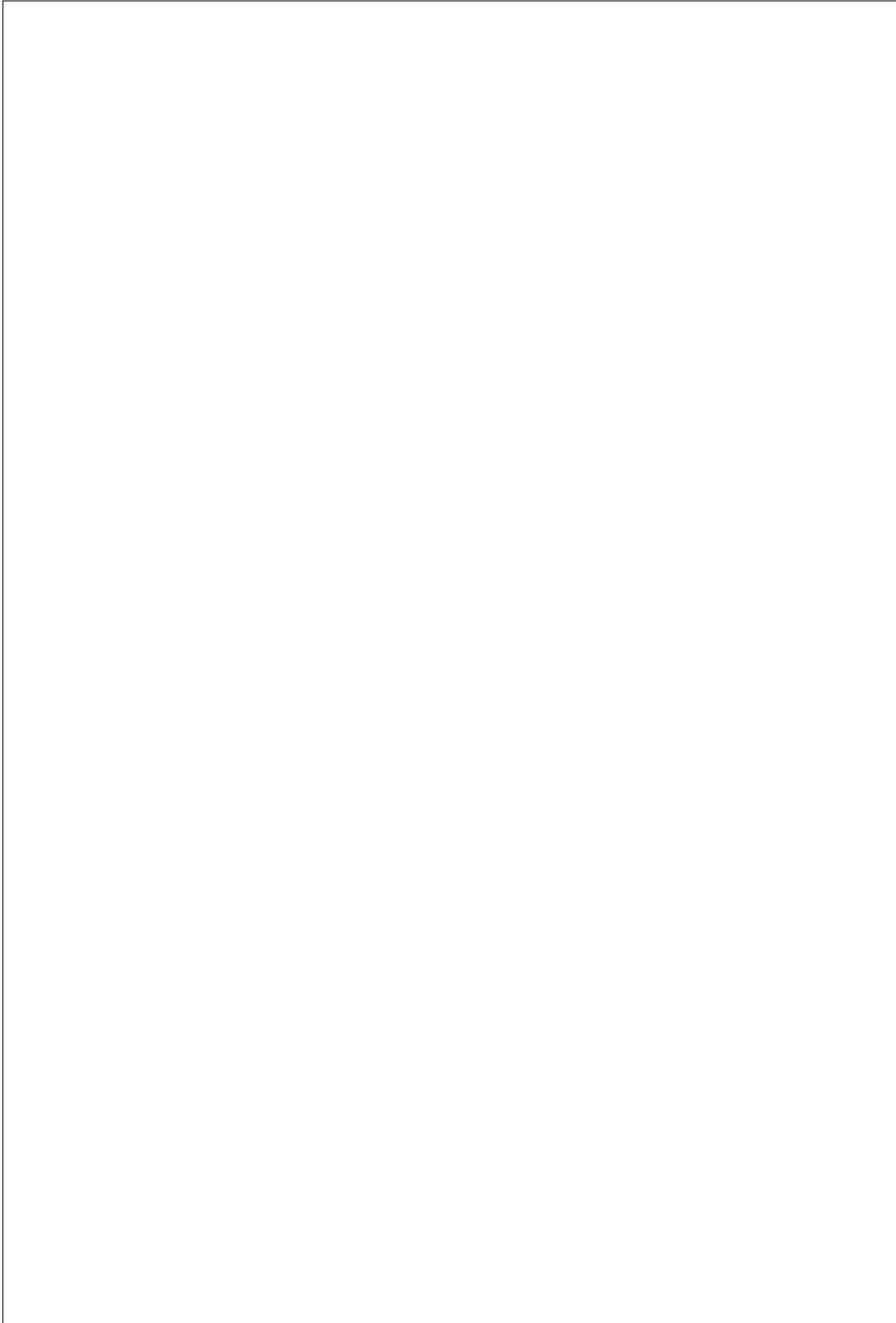
**Question 1**

1. En utilisant une boucle *for*, stockez dans une variable `l1` la liste contenant, dans l'ordre **décroissant**, les carrés de tous les entiers compris entre 0 et 10.

2. En utilisant cette fois une définition par compréhension, définissez une liste `l2` ayant le même contenu.

**Question 2** Écrivez une classe `Promotion` ayant les caractéristiques suivantes :

- son constructeur attendra un dictionnaire, qu'on stockera dans un attribut `notes`. On pourra supposer qu'une clef de `notes` est un nom d'étudiant, et la valeur associée sa note.
  - on fera en sorte que l'affichage d'un objet de type `Promotion` se fasse comme une série de lignes "L'étudiant E a obtenu N", où les clefs/valeurs E/N parcourent l'ensemble du dictionnaire `notes`.
  - elle possèdera une méthode `moyenne()` qui renverra la moyenne des notes de la promotion.
- Il n'est pas demandé d'implémenter la méthode `__eq__()`.



**Question 3** Construisez une classe `Somme` dont le constructeur attend une liste d'entiers `lst`. Une `Somme` doit être itérable. À chaque itération, il faudra avancer d'un cran dans `lst` et renvoyer la somme de tous les entiers vus jusqu'ici. Autrement dit, lors de la  $n$ -ième itération, il faudra renvoyer la somme des  $n$  premiers éléments de `lst`. L'itération devra s'arrêter une fois qu'on aura atteint la fin de `lst`.

Le comportement attendu est donc le suivant :

```
1 somme = Somme([2,4,3])
2 for s in somme:
3     print(s)
4 # affiche 2 puis 6 puis 9
```



**Question 4** Écrivez une fonction `carres()` qui attend en argument un itérable `it`. On supposera que les valeurs produites par `it` sont toujours des nombres – attention, `it` n’est pas forcément une liste!

La fonction `carres()` doit être une fonction génératrice et renvoyer tour-à-tour les carrés des entiers produits par `it`. Cette fonction s’arrêtera quand (et si) `it` s’arrête de produire.

Le comportement attendu est donc le suivant :

```
1 for n in carres([2,4,3]):
2     print(n)
3 # affiche 4 puis 16 puis 9
4
5 # supposons que l'iterable it produise les valeurs 1, 2, 3, 4, ...
6 for n in carres(it):
7     print(n)
8 # affiche 1 puis 4 puis 9 puis 16 puis ...
```

