
OI² — Olympiades d'Informatique de l'Océan Indien

Numérique et Sciences Informatiques (NSI)

classes de première

Durée de l'épreuve : 3 heures

Document autorisé : “Memento Python 3 — 1^{re}” (fourni)

Cette épreuve individuelle se déroule sur poste informatique déconnecté du réseau local de l'établissement et du réseau Internet.

Ce document comporte 5 pages (sans compter cette page de garde). Vérifiez que votre exemplaire est complet avant de commencer.

Certaines énigmes nécessitent d'exploiter des fichiers. Ces fichiers sont fournis avec le sujet.

La personne candidate indique ses réponses aux énigmes à la page 5, à remettre à la fin de l'épreuve.

1 L’auteur caché (22 pts)

Dans le fichier `codes_python.py` la variable `mystere` référence une très longue chaîne de caractères qui contient un message et le nom de son auteur. En ne retenant que les caractères dont les positions sont des nombres premiers¹, tu pourras lire le message et son auteur.

Question 1 : Qui est l’auteur du message ?

2 Le coffre fort (25 pts)

Antoine et Blandine s’échangent régulièrement des messages chiffrés. Ils utilisent le procédé de chiffrement suivant :

1. écrire le message ligne par ligne dans une grille carrée suffisamment grande ;
2. obtenir un message codé en lisant la grille colonne par colonne (chaque colonne de haut en bas) ;
3. rajouter la taille de la grille utilisée au début du message obtenu.

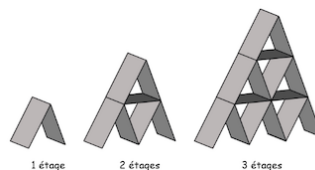
Par exemple si on veut chiffrer `CECI EST UN MESSAGE CHIFFRE DANS UNE GRILLE` on construit la grille 6×6 ci-dessous. Et en lisant colonne par colonne, on obtient le cryptogramme suivant (6 est la taille de la grille utilisée): `6CTSIAGEUAFNRCNGFSIIMERULEECENLSSHDEE`

C E C I E S	Vous interceptez une communication d’Antoine et Blandine. Le message, trop long, est partiellement reproduit ici mais vous êtes donné intégralement dans le fichier <code>codes_python.py</code> :
T U N M E S	
S A G E C H	
I F F R E D	
A N S U N E	
G R I L L E	
A U U I L	EDQU NF D EAATPPMLEUX TOQADR TOLA Q A ORUNO MTUASSUDTB

Question 2 : Quelle est la combinaison du coffre (réponse en chiffres) ? (*Indication :* la taille de la grille utilisée est 17)

3 Le château de cartes (27 pts)

À l’aide de cartes à jouer, on peut réaliser des châteaux comme sur l’image ci-dessous qui montre 3 exemples de châteaux, respectivement de 1 étage utilisant 2 cartes, de 2 étages à l’aide de 7 cartes et de 3 étages nécessitant 15 cartes.



Le record du monde est détenu par un américain avec un château de 54 étages.

- **Question 3A :** Combien a-t-il fallu de cartes pour le réaliser ? (13 pts)
- **Question 3B :** Avec 2026 cartes, le plus grand château qu’on peut réaliser comporte n étages et il reste m cartes inutilisées. Que vaut $n + m$? (14 pts)

¹Un nombre premier est un nombre qui possède exactement 2 diviseurs : 1 et lui-même.

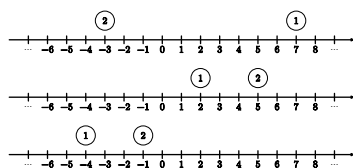
4 Les Puces sauteuses (32 pts)

Deux puces se déplacent par bonds successifs sur le même axe. On recueille pendant un certain laps de temps les sauts effectués par ces puces.

Par exemple, les deux séries suivantes indiquent que la puce n°1 a effectué un saut vers la droite de 7 cm puis un autre de 5 cm vers la gauche et un troisième de 6 cm toujours vers la gauche. Pendant le même laps de temps la puce n°2 a enchainé un saut vers la gauche de 3 cm puis un de 8 cm vers la droite et enfin un troisième saut vers la gauche de 6 cm :

```
sauts_puce_1 = [7, -5, -6]
sauts_puce_2 = [-3, 8, -6]
```

La figure suivante montre les positions des deux puces après chacun des sauts :



On suppose que les puces partent du même point à l'instant 0, et effectuent leurs sauts à la même

vitesse de 1 saut par seconde. On cherche à savoir à quel moment, après le premier saut, elles ont été le plus proche l'une de l'autre et à quelle distance. On ne mesure la distance entre les puces qu'une fois qu'elles sont au sol.

Par exemple, nos deux puces sont à une distance de 10 à l'instant 1, puis à une distance de 3 à l'instant 2 et enfin à une distance de 3 à l'instant 3. La réponse attendue ici est donc 3 pour la distance minimale et 2 pour l'instant (qui est le premier où la distance minimale a été atteinte).

Question 4 : À partir des deux listes `sauts_1` et `sauts_2` fournies dans le fichier `codes_python.py` déterminer d la distance minimale entre les deux puces et l'instant t à laquelle cette distance est atteinte pour la première fois. Donner la valeur de $d + t$.

5 La BNS (Banque Nationale Shadok) (37 pts)

Les Shadoks comptaient en base 4 avec les “chiffres” suivants : GA qui vaut 0, BU qui vaut 1, ZO qui vaut 2 et MEU qui vaut 3.

La banque des Shadoks comporte 1000 clients, vous disposez d'un fichier `05_donnees.txt` qui liste ces clients et la somme en *Soudok* (la monnaie des Shadoks) que chacun a déposée en banque. Ci-contre un extrait de ce fichier :

```
Client11: BU BU BU MEU BU
Client12: BU GA MEU GA ZO
Client13: MEU BU GA ZO
Client14: ZO BU GA GA BU
```

En écriture décimale, le `Client11` possède 349 *Soudok*, le `Client12` possède 306. Pour chacune des questions suivantes, vous donnerez la réponse en écriture décimale :

- **Question 5A :** Combien de Soudok possède le `Client13` ? (7 pts)
- **Question 5B :** Quel est le montant total en banque ? (30 pts)

Aide : dans le fichier `codes_python.py` une fonction permettant de lire le fichier et de récupérer la liste des montants *soudok* vous est fournie.

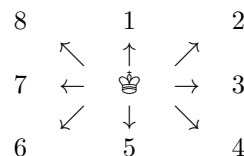
6 Look & Say (50 pts)

La suite auto-descriptive de J. H. Conway est une suite de lignes de chiffres :

- la première ligne est constituée d'un seul chiffre : 1
- puis chaque ligne est l'écriture en chiffres de la description de la ligne précédente, ainsi :
 - la deuxième ligne décrit la première où on lit : “un 1” ce qui donne 11 en chiffres ;
 - pour la troisième, on lit la deuxième : “deux 1” soit 21 en chiffres ;
 - pour la quatrième, on lit la troisième : “un 2, un 1” soit 1211 en chiffres
 - etc.
- **Question 6A** : Quelle est la ligne 5 ? (10 pts)
- **Question 6B** : Combien de 1 à la ligne 40 ? (40 pts)

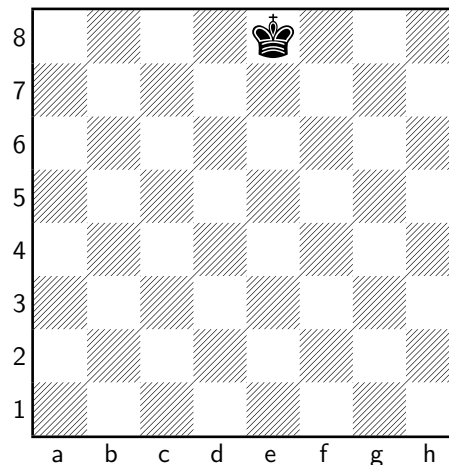
7 Le Roi en balade (52 pts)

Le jeu d'Échecs se joue sur un échiquier de 8×8 cases. Une des pièces du jeu est le Roi qui peut se déplacer d'une case dans une des 8 directions. On numérote ces directions de 1 à 8 comme indiqué sur la figure de droite :



Attendant la prochaine partie, le Roi noir, positionné sur sa case en **e8** (voir le schéma ci-contre qui donne le nom de chacune des cases), fait les cents pas suivant une liste de déplacements aléatoires. On s'intéresse au nombre de cases visitées pendant cette balade. Lorsqu'un déplacement provoque une sortie de l'échiquier, le Roi ne bouge pas et passe au déplacement suivant.

- **Question 7A** : avec la liste des déplacements `cent_pas` du fichier `codes_python.py` combien de cases le Roi va-t-il visiter ? (23 pts)
- **Question 7B** : donne le nom de la case la plus visitée – on garantit qu'il n'y en a qu'une (29 pts).



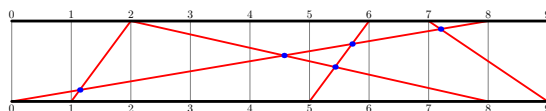
8 Méli-mélo à Tokyo (55 pts)

La ville de Tokyo est réputée pour la complexité de son réseau électrique aérien. Certaines de ses rues sont surchargées de câbles électriques qui s'entremêlent.



Partie A. Intersections sur une rue

On vous donne une liste de câbles électriques qui traversent une rue orientée ouest-est. Chaque câble est donné sous forme d'un couple d'entiers (x, y) où x est la position de départ du câble sur la façade nord et y la position d'arrivée du câble sur la façade sud. Par exemple, la liste $[(2, 8), (8, 0), (2, 1), (6, 5), (7, 9)]$ correspond à la disposition de câbles suivante :



On constate qu'il y a au total 5 intersections de câbles au-dessus de cette rue (marquées par des points sur la figure). Un point de départ (ou d'arrivée) partagé entre deux câbles ne compte pas comme une intersection.

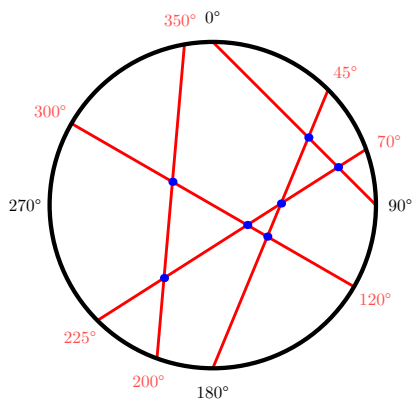
Question 8A : une liste `rue` de câbles est définie dans le fichier `codes_python.py` ; combien compte-t-on d'intersections de câbles 2 à 2 au-dessus de cette rue ? (24 pts)

Partie B. Intersections sur une place circulaire

On s'intéresse maintenant à une place circulaire traversée par des câbles électriques. Un câble est donné encore une fois sous forme d'un couple de valeurs (x, y) représentant les points de départ et d'arrivée du câble. Cette fois-ci cependant x et y désignent l'orientation magnétique (en degrés) par rapport au cap nord. Ainsi la direction Nord a pour valeur 0, l'Est a pour valeur 90, le Sud 180 et l'Ouest 270. Par exemple, la liste $[(0, 90), (45, 180), (200, 350), (225, 70), (300, 120)]$ correspond à la disposition de câbles de la figure de droite :

On constate que ces 5 câbles forment au total 7 intersections de câbles au-dessus de la place (marquées par des points sur la figure). Un

point de départ (ou d'arrivée) partagé entre deux câbles ne compte pas comme une intersection.



Question 8B : une liste `place` de câbles est définie dans le fichier `codes_python.py` ; combien compte-t-on d'intersections de câbles 2 à 2 au-dessus de cette place ? (31 pts)

Vos réponses aux énigmes

• Nom :

• Prénom :

Réponse 1	
Réponse 2	
Réponse 3A	
Réponse 3B	
Réponse 4	
Réponse 5A	
Réponse 5B	
Réponse 6A	
Réponse 6B	
Réponse 7A	
Réponse 7B	
Réponse 8A	
Réponse 8B	