

Devoir surveillé n°1 - Durée : 2h - Jeudi 14 novembre 2024

Rédiger sur une copie double. La calculatrice n'est pas autorisée.

Consignes de présentation :

- Nom Prénom lisibles en haut à gauche, classe en haut à droite, DS n°1 au milieu et encadré à la règle.
- Vous laissez ensuite 5 ou 6 centimètres pour mes éventuels commentaires, délimités par deux traits tracés à la règle.
- S'il n'y a pas de marge déjà tracée sur votre feuille, vous en tracez une à 4 cm du bord de la feuille.
- **Les codes Python demandés doivent être écrits très lisiblement, avec une indentation très claire.**

Exercice 1 : On s'intéresse aux matrices parcimonieuses, c'est-à-dire aux matrices dont la plupart des coefficients sont nuls. Une telle matrice M de taille (n, p) pourra être codée par un dictionnaire ayant pour couples clefs/valeurs :

- 'dim' : (n, p)
- (i, j) : M_{ij} pour chaque couple (i, j) tel que $M_{ij} \neq 0$.

Par exemple la matrice $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 15 & 0 \end{pmatrix}$ sera codée par le dictionnaire {'dim' : (2,4), (1,2) : 15}

1. Proposer une fonction d'addition de deux matrices parcimonieuses (que l'on suppose de même dimension). Les deux matrices données en argument seront de type dictionnaire comme expliqué ci-dessus, et le résultat renvoyé devra aussi être un dictionnaire de cette forme.
2. Si la première matrice contient c_1 coefficients non nuls, et la seconde c_2 , quelle est la complexité de cet algorithme ?

Exercice 2 : Introduction

Cet exercice s'intéresse à un système de type "Pass Navigo" (carte d'accès aux transports en communs d'Ile de France), dont le principe est basé sur l'identification par radio fréquence (RFID pour *Radio Frequency Identification*).

Un système RFID est composé de deux entités qui communiquent entre elles : La première est un transpondeur, capable de répondre à une demande venant d'un lecteur. Dans notre exemple il s'agit de la carte Navigo de l'utilisateur. La seconde est une station de base, ou lecteur RFID, qui a pour mission d'identifier le transpondeur, et communiquer avec lui par ondes électromagnétiques.

L'objectif de cet exercice est de déterminer si un voyageur est autorisé ou non à franchir un point de contrôle.

Utilisation des données de la puce pour autoriser ou non le passage

Lorsqu'un utilisateur présente son titre de transport (c'est-à-dire sa carte) face au lecteur d'un point de contrôle, la puce et le lecteur s'identifient mutuellement, puis le lecteur récupère les données de la puce permettant de déterminer si le passage est autorisé ou non. À l'issue de cette récupération, l'ordinateur auquel est relié le lecteur dispose d'un fichier texte 0001.txt semblable à l'exemple ci-dessous.

```

1 49987654
2 1, 3, 2025-08-31
3 2024-10-29, 08:34:15, 4568
4 2024-10-28, 20:21:48, 365
5 2024-10-28, 18:47:54, 987

```

En d'autres termes, ce fichier possède toujours *exactement* la structure suivante :

- La première ligne contient un entier servant à identifier le titre de transport.
- Le réseau de transport est divisé en plusieurs zones numérotées, et le titre n'est valide que dans un ensemble de zones contiguës. La seconde ligne contient les bornes de l'intervalle dans lequel le titre est valide (ici, il s'agit des zones 1 à 3 incluses) ainsi que la date de fin de validité du titre au format *aaaa-mm-jj* (ici, il s'agit du 31 août 2025).
- Les trois lignes suivantes contiennent des données relatives aux trois derniers passages effectués à l'aide du titre : date, heure (au format *hh:mm:ss* sur 24 heures) et identifiant entier du point de passage (gare, arrêt, ...).

On donne le bloc d'instructions utilisé pour lire les deux premières lignes de ce fichier :

```

1 fichier = open('0001.txt', 'r')
2 lignes = fichier.readlines()
3 fichier.close()
4
5 # Ligne 1 : recuperation de l'identifiant du titre
6 id_titre = int(lignes[0])
7
8 # Ligne 2 : recuperation des donnees du titre de transport
9 donnees_titre = lignes[1].rstrip('\n').split(',')
10 zones = [int(donnees_titre[0]), int(donnees_titre[1])]
11 ch_date_fin = donnees_titre[2].split('-')
12 date_fin = [int(ch_date_fin[0]), int(ch_date_fin[1]), int(ch_date_fin[2])]

```

1. Donner les types et les valeurs des variables `id_titre`, `zones` et `date_fin` à l'issue de ces instructions pour le fichier `0001.txt` donné ci-dessus.

On souhaite placer le contenu des trois dernières lignes dans un tableau d'entiers nommé `passages`, dont chaque ligne corresponde à un passage (dans l'ordre dans lequel ils apparaissent dans le fichier) et dont les colonnes soient définies comme suit :

Indice	0	1	2	3	4	5	6
Contenu	Année	Mois	Jour	Heures	Minutes	Secondes	Point de passage

2. Écrire le bloc d'instructions à exécuter à la suite des opérations précédentes pour construire le tableau `passages` à partir des lignes 3, 4 et 5 contenues dans la liste `lignes`. (le tableau `passages` sera donc une liste de longueur 3, dont chaque élément est une liste de longueur 7).

Les données relatives au lecteur sont décrites par les variables suivantes (on utilise une initiale majuscule pour bien différencier les données du lecteur de celles du titre ; le type est indiqué entre parenthèses) :

- **Zone** (entier) : indique la zone dans laquelle se trouve le lecteur ;
- **Id_point** (entier) : indique l'identifiant du point de passage où se trouve le lecteur ;
- **Liste_noire** (liste d'entiers) : contient les identifiants des titres ayant été déclarés perdus, volés ou détériorés par leurs propriétaires, et devant donc être refusés ;
- **Maintenant** (liste de six entiers) : contient la date et l'heure au format [année, mois, jour, heures, minutes, secondes].

Le passage doit être autorisé si les conditions suivantes sont *toutes* vérifiées :

- l'identifiant du titre n'est pas dans la liste noire du lecteur ;
- la zone du lecteur appartient à l'intervalle de validité du titre ;
- la date du jour est antérieure à la date de fin de validité du titre ;
- si l'une des trois dernières validations a été effectuée au même point de passage que celui où est installé le lecteur, elle doit avoir été effectuée il y a plus de 450 secondes (ceci afin de décourager l'utilisation frauduleuse d'un même titre par plusieurs voyageurs).

Enfin, lorsqu'un passage est refusé, un message apparaît sur un afficheur LCD pour donner la raison du refus. Cet afficheur possède une seule ligne et il faut donc définir des priorités au cas où plusieurs des conditions ci-dessus ne seraient pas remplies. L'ordre de priorité et les messages correspondants sont donnés ci-dessous :

- "Titre refuse" si l'identifiant est dans la liste noire ;
- "Non valide dans cette zone" si le lecteur est hors des zones de validité du titre ;
- "Titre expire" si la date de fin de validité du titre est dépassée ;
- "Titre déjà valide" si le titre a déjà été validé dans le même lieu il y a moins de 450 secondes.

3. Écrire une fonction `estAvant(date1,date2)` prenant pour arguments deux dates au format [année, mois, jour] (donc sous forme de listes de trois entiers chacune) et retournant `True` si `date1` est antérieure ou égale à `date2`, et `False` sinon.
4. Écrire une fonction `nbSecondesEntre(heure1,heure2)` prenant pour arguments deux horaires au format [heures, minutes, secondes] (donc sous forme de listes de trois entiers chacun) et retournant le nombre de secondes séparant les deux instants. Le résultat devra être positif si `heure1` est *postérieure* à `heure2`.
5. Écrire alors une fonction `testPassage`, dont les arguments sont à préciser, retournant la valeur `True` si le passage est autorisé et la valeur `False` sinon et, dans ce dernier cas, affichant à l'écran le message correspondant aux règles de priorité ci-dessus. On pourra utiliser toutes les variables définies dans cette section et appeler les fonctions définies dans les deux questions précédentes.

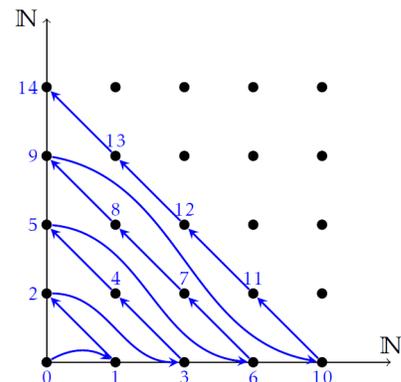
Exercice 3 :

En cours de mathématiques, vous démontrez cette année que l'on peut établir une bijection entre l'ensemble $\mathbb{N} \times \mathbb{N}$ et l'ensemble \mathbb{N} .

Une manière de faire est d'attribuer un numéro à chaque couple $(x, y) \in \mathbb{N}^2$, en suivant le procédé suggéré par la figure ci-contre.

Par exemple le couple $(0,0)$ aura le numéro 0, le couple $(1,0)$ aura le numéro 1, ..., le couple $(2,1)$ aura le numéro 7, etc.

Il n'est pas question ici de déterminer explicitement l'expression de cette bijection.



1. Rédiger une fonction *récursive* `numero` qui retourne le numéro du point de coordonnées (x, y) .
2. Rédiger la fonction réciproque `couple`, là encore de façon *récursive*. (à un numéro entier elle doit associer le couple d'entiers naturels correspondants)