

TP - Méthodologie de la Programmation

25 Juin 2021

Recommandations

Pour votre convenance, l'énoncé est disponible au format .ipynb et .pdf. **Il s'agit du même exercice.**

Vous pouvez réaliser les exercices directement sur jupyter et rendre le fichier .ipynb modifié par vos soins, sans oublier le fichier de sortie. **Pensez bien à indiquer votre nom, prénom et n. d'étudiant.**

Vous devrez envoyer votre travail (exercice et fichier de sortie) sur le mail josue.melka02@univ-paris8.fr. Tout travail rendu après le délai fixé ne sera pas pris en compte.

1 Exercice

Soit le fichier `nombres.txt` contenant plusieurs listes de nombres séparés par des virgules, une par ligne, comme ceci :

```
lst_a : 17, 28, 31, 43, 53, 78, 217, 261, 346, 401, 409, 513, 577
lst_b : 333, 206, 693, 197, 485, 404, 704, 363, 151, 424, 199, 936
lst_c : 111, 174, 19, 267, 293, 354, 367, 391, 408, 445, 452, 492
...
```

L'objectif est de vérifier si ces listes sont ordonnées ou non, et d'enregistrer le résultat dans un fichier.

1.1 Lecture d'une ligne

Écrire une fonction `lire_nombres(ligne)` qui lit une ligne du fichier et retourne un tuple avec le label de la ligne et la liste de nombres correspondant (type `int`).

Exemple:

```
>>> lire_ligne("lst_a : 17, 28, 78, 217, 346, 409, 577, 588, 781\n")
('lst_a', [17, 28, 78, 217, 346, 409, 577, 588, 781])
```

1.2 Test d'ordre

Écrire une fonction `ordonne(liste)` qui retourne un booléen indiquant si une liste de nombres est ordonnée ou non (que ce soit en ordre croissant ou décroissant).

Exemples:

```
>>> ordonne([1, 2, 3, 4])
True
>>> ordonne([3, 2, 4, 1])
False
>>> ordonne([4, 3, 2, 1])
True
```

Note: Vous pouvez réaliser des fonctions auxiliaires `croissant(liste)` et `decroissant(liste)` pour simplifier.

1.3 Lecture du fichier

Lire le fichier ligne par ligne en convertissant les lignes en listes de nombres avec la fonction vue en 1.1.

1.4 Écriture des résultats

Produire les résultats dans le fichier `resultats.txt` sous cette forme.

```
lst_a : Non
lst_b : Non
lst_c : Oui
...
```

Note: Vous pouvez utiliser la fonction vue en 1.2.