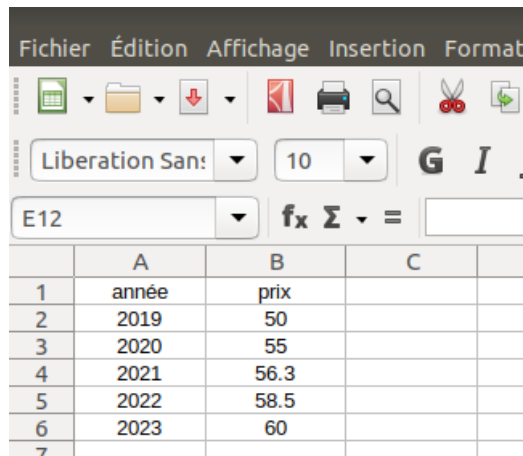


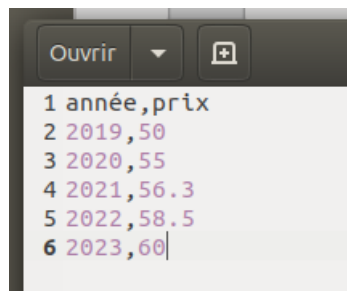
1 Le problème

On dispose d'un fichier au format csv contenant des données structurées dans une table. Voici un aperçu d'un fichier `exemple.csv` ouvert dans un logiciel tableur, ainsi que dans un simple éditeur de texte.



	A	B	C
1	année	prix	
2	2019	50	
3	2020	55	
4	2021	56.3	
5	2022	58.5	
6	2023	60	
7			

exemple.csv ouvert dans LibreOffice



```
1 année,prix
2 2019,50
3 2020,55
4 2021,56.3
5 2022,58.5
6 2023,60
```

exemple.csv ouvert dans Gedit

Afin de pouvoir effectuer des traitements automatiques (éventuellement complexes), on désire récupérer le contenu de ce fichier à l'aide d'un programme écrit en Python.

Chaque ligne du fichier csv sera « enregistrée » dans un tableau (liste-Python).

Par exemple, la 1ère ligne sera la liste `['année', 'prix']`, la 2ème `['2019', '50']`, etc.

Enfin, on enregistrera chacune de ces listes dans un tableau global représentant l'ensemble de la table du fichier csv. Ce tableau sera dans notre exemple la liste suivante :

```
[['année', 'prix'], ['2019', '50'], ...].
```

Le but de ce projet est donc d'écrire une fonction `csv1` qui prend en paramètres d'entrée l'adresse du fichier csv et le séparateur utilisé dans ce fichier, et qui renvoie le tableau de tableaux tel que décrit précédemment.

Exemple :

```
>>> tab = csv1('chemin/vers/exemple.csv', ',')
>>> tab
[['année', 'prix'],
 ['2019', '50'],
 ['2020', '55'],
 ['2021', '56.3'],
 ['2022', '58.5'],
 ['2023', '60']]
```

Questions préliminaires :

1. Expliquer ce qu'est un fichier csv (cf cours SNT ou recherche web).
2. En vous appuyant sur l'exemple proposé et à la variable `tab` obtenue, comment accéder aux descripteurs du fichier csv ?
3. Comment accéder à l'année de la ligne n°4 (ce numéro de ligne fait référence à celui du tableur ou de l'éditeur de texte [*reponse : 2021*]) ?
4. Comment accéder au prix de la ligne n°4 [*reponse : 56.3*], au format numérique (`float`) ?

2 Découpage du projet

On demande dans ce projet d'écrire également quelques fonctions annexes.

2.1 Fonction tranche

Spécification de la fonction `tranche` :

Renvoie une tranche (une sous-chaîne) d'une chaîne de caractères entre les indices `d` (inclu) et `f` (exclu).

Entrées :

- chaîne : str : chaîne à découper.
- d, f : int, int : indice de début (inclu) et indice de fin (exclu).

Sortie : sous-chaîne : str.

Exemple :

```
>>> tranche("bonjour à vous tous", 10, 14)
'vous'
```

Remarque : cette fonction implémente la fonctionnalité intégrée de Python qui s'appelle les slices.

2.2 Fonction indices

Spécification de la fonction `indices` :

Renvoie les indices auxquels on trouve un caractère dans une chaîne.

Entrées :

- chaîne : str : chaîne à analyser.
- car : str : caractère à rechercher.

Sortie : tab : [int] : tableau des indices où on trouve le caractère recherché.

Exemple :

```
>>> indices("bonjour à vous tous", 'o')
[1, 4, 11, 16]
```

Remarque : cette fonction implémente une extension de la méthode `index` des chaînes Python.

2.3 Fonction partage

Spécification de la fonction `partage` :

Découpe une chaîne en sous-chaînes, en utilisant un caractère comme séparateur des sous-chaînes. Ces sous-chaînes sont renvoyées dans un tableau.

Entrées :

- chaîne : str : chaîne à analyser.
- car : str : caractère séparateur des sous-chaînes.

Sortie : tab : [str] : tableau des sous-chaînes.

Exemples :

```
>>> partage("bonjour à vous tous", 'o')
['b', 'nj', 'ur à v', 'us t', 'us']
>>> partage(",chp2,chp3,,chp5,,", ',')
['', 'chp2', 'chp3', '', 'chp5', '', '', '']
```

Remarque : cette fonction implémente la méthode `split` des chaînes Python.