TP 06

Tableaux et chaines de caractères

Savoirs et compétences :

- ❒ *AA.C9 : Choisir un type de données en fonction d'un problème à résoudre*
- ❒ *AA.S11 : Manipulation de quelques structures de données.*

Consignes

Attention : suivez précisément ces instructions. Vous enverrez à votre enseignant un fichier d'extension `.py` (script `Python`) nommé

`tp07_durif_kleim.py`,

où les noms de vos enseignants sont à remplacer par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe, ni majuscule. Dans ce fichier, vous respecterez les consignes suivantes.

- Écrivez d'abord en commentaires (ligne débutant par #), le titre du TP, les noms et prénoms des étudiants du groupe.
- Commencez chaque question par son numéro écrit en commentaires.
- Les questions demandant une réponse écrite seront rédigées en commentaires.
- Les questions demandant une réponse sous forme de fonction ou de script respecteront pointilleusement les noms de variables et de fonctions demandés.

Le fichier produit à la question **??** portera un nom du type `q??_durif_kleim.csv`, où les noms de vos enseignants sont à remplacer par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe, ni majuscule. Pour produire ce fichier, vous respecterez le format csv, notamment en utilisant le séparateur « virgule » (`,`) pour délimiter les cellules.

Activité 1 : Analyse d'un dipôle électrique

Copier et coller le fichier "dipole_electrique.txt" que vous trouverez sur le site de la classe dans votre répertoire personnel. Celui-ci contient des données numériques, obtenues en TP d'électronique, correspondant à différentes mesures de l'intensité traversant un dipôle et de la tension entre ses bornes. On souhaite récupérer ces données et les exploiter.

Ouvrir le fichier "dipole_electrique.txt" à l'aide du bloc-notes et observer son contenu.

Q 1 : Écrire un programme Python que vous appellerez `lit_dipole` et qui prendra en argument une variable de type chaine de caractère qui sera le nom du fichier de données. Cette fonction renverra deux listes notées `I` et `U` l'une contenant les intensités mesurées, et l'autre contenant les tensions mesurées.

Q 2 : Créer une fonction que l'on notera `tracer_dipole` qui prendra en argument deux listes représentant l'intensité I et U. Cette fonction permettra de représenter graphiquement les couples de points (I,U). On sauvegardera la figure avec le nom 'tp06_noms_q02.png'

Q 3 : Conjecturer alors une relation reliant la tension aux bornes du dipôle à l'intensité le traversant.

Activité 2 : Lecture d'un texte

Ouvrir le fichier `complot_contre_lamerique.txt` dans python (on prendra soin de nommer la variable contenant cet objet).

Q 4 : Que fait chacune des méthodes `read()`, `readline()` et `readlines()` ? Quels sont les types des valeurs que chacune des ces fonctions renvoient ?

Q 5 : Que représentent les symboles `\t` et `\n` ?

Q 6 : Écrire une fonction Python `carac(nom_de_fichier)` qui renvoie un tableau contenant le nombre de caractères de chaque ligne du fichier `nom_de_fichier`, retour chariot exclu.

Indication : attention au type de `nom_de_fichier` !

Q 7 : Écrire une fonction Python `compte_carac(carac,nom_de_fichier)` qui renvoie pour caractère donné le nombre d'occurence.

Q 8 : Écrire une fonction Python `stat_carac(carac,nom_de_fichier)` qui renvoie un tableau qui en fonction du numéro de la lettre dans l'alphabet renvoie le nombre d'occurence.

Q 9 : Écrire une fonction Python `trace_stat_carac(carac,nom_de_fichier)` qui trace en fonction du numéro de la lettre dans l'alphabet le nombre d'occurence.

Activité 3 : Résultats de l'Embrunman

Le fichier `embrunman2019.csv` contient les résultats de l'ultratriatlon de Embrun 2019. Les données concernent dans l'ordre :

- le classement ;
- le numéro de dossard ;
-

On pensera d'abord à ouvrir ce fichier dans un éditeur de texte puis dans un tableur afin de bien visualiser ces données.

Quelques rappels sur les chaînes. Il existe un moyen de « découper » des chaînes de caractères en Python : c'est la méthode `split`. Réciproquement, la méthode `join(t)` appliquée à une chaîne `x` permet de concaténer toutes les chaînes du tableau `t`, séparées par `x`. Il existe aussi des outils de conversion de nombres flottants en chaînes de caractère, et vice-versa. Tout cela s'utilise comme suit.

```
>>> s = '123,45,2;1587,45,;45'
>>> s.split(',')
['123', '45', '2;1587', '45', ';45']
>>> s.split(';')
['123,45,2', '1587,45,', '45']
>>> sep = '<'
>>> t = ['GA','BU','ZO','MEU']
>>> sep.join(t)
'GA<BU<ZO<MEU'
>>> str(123.456)
'123.456'
>>> float('456.123')
456.123
```

Enfin, on voudra bien entendu représenter les données contenues dans `embrunman2019.csv` sous forme de tableau à double entrée, c'est-à-dire comme une matrice. On représentera le tableau sous forme de liste Python, ligne par ligne, chaque ligne étant elle même une liste Python.