

Python - L2 MIASHS Examen de 2nde session

Romain Tavenard

8 juin 2016, 9h-10h

Pour cet examen de seconde session, tous les documents sont autorisés et l'accès à internet n'est pas interdit à l'exception stricte des moyens de discussion de type e-mails, messagerie instantanée, etc. Toute contrevenance à cette règle simple entraînera la note de 0 sans prise en compte d'aucune excuse que ce soit. Les questions de cet examen ne sont pas triées par ordre de difficulté, n'hésitez pas à commencer par celles qui vous semblent les plus accessibles.

Questions indépendantes

1. Écrivez une fonction qui retourne le terme de rang n (où n est un argument de la fonction) de la suite définie par :

$$\forall n \geq 1, u_n = 5 \cdot u_{n-1}^2 + 3$$

Lors de l'appel de cette fonction, on pourra (au choix) soit spécifier la valeur de u_0 , soit laisser le programme utiliser la valeur par défaut qui sera alors $u_0 = 0$.

2. Écrivez une fonction `count_occ` qui prenne en entrée une liste de chaînes de caractères et une chaîne de caractère (que l'on appellera `motif`) et retourne le nombre de chaînes de la liste qui contiennent au moins une fois le `motif`. Les comparaisons devront être insensibles à la casse. On fournit ci-après deux exemples d'exécution de la fonction :

```
> print(count_occ(["Romain", "Jean", "Paul", "Marie"], "MA"))
2
> print(count_occ(["Romain", "Jean", "Paul", "Marie"], "je"))
1
```

Questions liées

Si vous ne traitez pas la question 3, vous pouvez tout de même proposer une solution pour les questions suivantes reposant sur un appel à la fonction demandée

pour la question 3 : vous ne serez pas doublement pénalisé(e).

3. Écrivez une fonction `count_occ_lst_motifs` qui prenne en entrée une liste (notée `lst`) de chaînes de caractères et une liste de motifs (chaque motif étant lui-même une chaîne de caractères) et retourne un dictionnaire indiquant, pour chaque motif, le nombre de chaînes de la liste `lst` qui le contiennent. On fournit ci-après un exemple d'exécution de la fonction :

```
> lst = ["Romain", "Jean", "Paul", "Marie"]
> l_motifs = ["MA", "je"]
> print(count_occ_lst_motifs(lst, l_motifs))
{'MA': 2, 'je': 1}
```

4. Écrivez une fonction `most_frequent_motif` qui prenne en entrée une liste (notée `lst`) de chaînes de caractères et une liste de motifs (chaque motif étant lui-même une chaîne de caractères) et retourne un le motif que l'on retrouve dans le plus grand nombre de chaînes de la liste `lst`. Cette fonction pourra faire appel à la fonction `count_occ_lst_motifs` de la question 3.
5. Écrivez une fonction `write_motif_frequency_csv` qui prenne en entrée un nom de fichier `filename`, une liste (notée `lst`) de chaînes de caractères et une liste de motifs (chaque motif étant lui-même une chaîne de caractères) et écrit au format CSV, dans le fichier dont le nom est stocké dans `filename`, pour chaque motif, le nombre de chaînes de la liste `lst` qui le contiennent. Cette fonction pourra faire appel à la fonction `count_occ_lst_motifs` de la question 3. On fournit ci-après un exemple de contenu du fichier CSV créé :

```
motif;frequency
MA;2
je;1
```