Programmation python, session № 1

Master 2 SDS, Université Paris-Saclay 23/01/2024

Exercice: suppression des doublons

Dans cet exercice, vous allez créer un programme qui lit les mots de l'utilisateur jusqu'à ce que celui-ci entre une ligne blanche. Une fois que l'utilisateur a saisi une ligne vierge, votre programme doit afficher exactement une fois chaque mot saisi par l'utilisateur. Les mots doivent être affichés dans l'ordre dans lequel ils ont été saisis. Par exemple, si l'utilisateur entre :

first
second
first
third
second

votre programme devrait alors s'afficher :

first
second
third

Listing 1: une proposition de début de code.

```
1 words = []
2 word = input("Entrez un mot (un blanc pour quitter) :")
```

Exercice: simulation d'une paire de dés

Dans cet exercice, vous allez simuler 1000 lancers de deux dés. Commencer par écrire une fonction qui simule le lancement d'une paire de dés à six faces. Votre fonction ne prendra aucun paramètres. Elle renverra comme seul résultat le total obtenu sur les deux dés. Écrire un programme principal qui utilise votre fonction pour simuler le lancement de deux dés à six faces 1000 fois. Lorsque votre programme s'exécute, il doit compter le nombre de fois que chaque total se produit. Il doit ensuite afficher un tableau résumant ces données. Exprimer la fréquence de chaque total en pourcentage du nombre de lancers effectués. Votre programme doit également afficher le pourcentage attendu par la théorie des probabilités pour chaque total. Un exemple de sortie est présenté ci-dessous.

Total	Simulated	Expected
2	2.9	2.78
3	6.9	5.56
4	9.4	8.33
5	11.9	11.11
6	14.2	13.89
7	14.2	16.67
8	15.0	13.89
9	10.5	11.11
10	7.9	8.33
11	4.5	5.56
12	2.6	2.78

Listing 2: La seule librairie autorisée.

```
1 from random import randrange
```

Exercice : anagrammes

Deux mots sont anagrammes s'ils contiennent les mêmes lettres, mais dans un ordre différent. Par exemple, "evil" et "live" sont des anagrammes car ils contiennent chacun un "e", un "i", un "l" et un "v", un "i", un "l" et un "v". Créer un programme qui lit deux chaînes de caractères de l'utilisateur, détermine s'il s'agit ou non d'anagrammes et communique le résultat. **Attention : Votre programme ne doit faire appel à aucune librairie python. Indication : Penser à utiliser un dictionnaire au lieu d'une liste.**

Exercice : recherche récursive de palindrome

Une chaîne de caractères est un palindrome si elle est identique en avant et en arrière. Par exemple, "anna", "civic", "level" et "hannah" sont tous des exemples de mots palindromes. Écrire une fonction récursive qui détermine si une chaîne de caractères est un palindrome ou non. La chaîne vide est un palindrome, de même que toute chaîne ne contenant qu'un seul caractère. Toute chaîne plus longue est un palindrome si son premier et son dernier caractère correspondent, et si la chaîne formée en retirant le premier et le dernier caractère est également un palindrome. Écrire un programme principal qui lit une chaîne de l'utilisateur et utilise votre fonction récursive pour déterminer s'il s'agit ou non d'un palindrome. Votre programme doit ensuite afficher un message approprié pour l'utilisateur.