Python String Operations Questions

1. Vowel Counter:
   Write a Python function that takes a string as input. Using a for loop, iterate through each character. If the character is a vowel (a, e, i, o, u, case-insensitive), increment a counter. Store the count of each vowel in a dictionary. Finally, print the total number of vowels found and the dictionary of vowel counts.
2. Palindrome Checker with List:
   Write a Python function that accepts a string. Inside the function, use a for loop to build a new list containing the characters of the string in reverse order. Then, use an if-else statement to check if the original string is a palindrome (reads the same forwards and backward) by comparing it to the string formed from the reversed list. Print whether the string is a palindrome or not.
3. Character Frequency Counter:
   Create a Python function that takes a string. Use a for loop to iterate through the string and populate a dictionary where keys are characters and values are their frequencies. Use an if-else statement to handle whether a character is already in the dictionary or not. Print the resulting character frequency dictionary.
4. Word Length Classification:
   Write a Python function that takes a sentence (string) as input. Split the sentence into words. Using a for loop, iterate through each word. Use if-elif-else statements to classify words as "short" (length < 5), "medium" (5 <= length <= 8), or "long" (length > 8). Store these classified words in separate lists (e.g., short_words, medium_words, long_words). Print all three lists.
5. Substrings with Specific Start/End:
   Given a list of strings, write a function that iterates through each string. If a string starts with a specific prefix (e.g., "py") and ends with a specific suffix (e.g., "thon"), add it to a new list called matching_strings. Use an if statement to check the conditions. Print the matching_strings list.
6. Unique Word Extractor:
   Write a function that takes a paragraph (string) as input. Convert the paragraph to lowercase and remove punctuation. Use a for loop to split the paragraph into words and add each unique word to a set data structure. Finally, convert the set back to a sorted list and print it.
7. Even/Odd Indexed Character Separator:
   Create a Python function that takes a string. Using a for loop with enumerate, iterate through the string. Use an if-else statement to check if the index of a character is even or odd. Store characters at even indices in one list and

characters at odd indices in another list. Print both lists.

8. String Anagram Checker:
   Write a function that takes two strings as input. Convert both strings into lists of characters. Sort both lists. Use an if-else statement to check if the sorted lists are identical. Print whether the two strings are anagrams of each other (contain the same characters with the same frequencies).
9. Missing Character Finder (Alphabet Check):
   Given a string, write a function that determines which lowercase English alphabet letters are not present in the string. Use a set to store all lowercase letters. Iterate through the input string, removing each found letter from the set. Print the remaining letters in the set.
10. Custom String Reversal (Word by Word):
    Write a function that takes a sentence. Use a for loop to iterate through the words of the sentence. Store each word in a list. Then, using another loop (or list slicing), reverse the order of words in the list. Finally, join the reversed words back into a single string and print the result. (Example: "Hello World" -> "World Hello")