Rajalakshmi Engineering College

Name: Jovitha J
Email: 240701219@rajalakshmi.edu.in
Roll no: 240701219
Phone: 7825034021
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 3_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 25

Section 1 : Coding

1. Problem Statement

Write a program to check if a given string is perfect.

A perfect string must satisfy the following conditions:

The string starts with a consonant.The string alternates between consonants and vowels.Each consonant appears exactly once.Vowels can occur consecutively multiple times but should not be followed immediately by a consonant.

If the string satisfies all these conditions, print "True"; otherwise, print "False".

Input Format

The input consists of a string.

Output Format

The output prints "True" if the string is perfect. Otherwise, print "False".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: capacitor
Output: True

Answer

```
def is_perfect_string(s):
    """
    Checks if a given string is perfect according to the defined rules.

    Args:
        s: The input string (lowercase).

    Returns:
        "True" if the string is perfect, "False" otherwise.
    """
    vowels = "aeiou"
    consonants = "bcdfghjklmnpqrstvwxyz"

    # 1. The string starts with a consonant.
    if not s or s[0] not in consonants:
        return "False"

    # Initialize a set to track used consonants.
    used_consonants = set()
    expected_char = 'consonant'  # Start expecting a consonant

    for char in s:
        if expected_char == 'consonant':
            if char not in consonants:
                return "False"  # Not a consonant
            if char in used_consonants:
                return "False"  # Consonant already used
            used_consonants.add(char)
            expected_char = 'vowel'  # Next expected is vowel
        else:  # expected_char == 'vowel'
            if char not in vowels:
                return "False" # Vowel expected but consonant found.

    return "True" # If the loop completes without returning, the string is perfect

def main():
    """
    Main function to take input and call the is_perfect_string function.
    """
    s = input()
    result = is_perfect_string(s)
    print(result)

if __name__ == "__main__":
    main()
```

Status : Partially correct ***Marks : 5/10***

2. Problem Statement

Sarah is a technical writer who is responsible for formatting two important documents. Both documents contain a certain placeholder character that needs to be replaced with another character before they can be finalized. To ensure consistency in formatting, Sarah wants you to help her write a program that processes both documents by replacing the placeholder character with the new one.

Sarah also prefers a neat and structured output, so she wants you to ensure that both modified documents are printed in a single line, separated by a space, using the format() function.

Example

Input:

Hello

World

o

a

Output:

Hella Warld

Explanation:

Here the character 'o' is replaced with 'a' in the concatenated string.

Input Format

The first line contains string1, the first document.

The second line contains string2, the second document.

The third line contains char1, the placeholder character that needs to be replaced.

The fourth line contains char2, the new character that will replace the placeholder.

Output Format

The output displays a single line containing the modified string1 and string2, separated by a space.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: Hello
World
o

a
Output: Hella Warld

Answer

```
# You are using Python
# Input strings and characters
string1 = input().strip()
string2 = input().strip()
char1 = input().strip()
char2 = input().strip()

# Replace placeholder character with the new character in both strings
modified_string1 = string1.replace(char1, char2)
modified_string2 = string2.replace(char1, char2)

# Print both modified strings in a single line separated by a space
print("{} {}".format(modified_string1, modified_string2))
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Raj wants to write a program that takes a list of strings as input and returns the longest word in the list. If there are multiple words with the same length, the program should return the first one encountered.

Help Raj in his task.

Input Format

The input consists of a single line of space-separated strings.

Output Format

The output prints a string representing the longest word in the given list.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: cat dog elephant lion tiger giraffe
Output: elephant

Answer

```
# You are using Python
# Input the list of strings
words = input().split()

# Initialize variables to keep track of the longest word
longest_word = words[0]

# Loop through the words to find the longest one
for word in words:
    if len(word) > len(longest_word):
        longest_word = word

# Output the longest word
print(longest_word)
```

Status : Correct ***Marks : 10/10***