

Rajalakshmi Engineering College

Name: Mahalakshmi S
Email: 240701301@rajalakshmi.edu.in
Roll no:
Phone: 9566463810
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 : 30

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

```
s = input().strip()
vowels = 'aeiou'

if not s or s[0] in vowels:
    print("False")
else:
    i = 0
    valid = True
    expecting_consonant = True

    while i < len(s):
        if expecting_consonant:
            if s[i] in vowels:
                valid = False
                break
            i += 1
            expecting_consonant = False
        else:
            if s[i] not in vowels:
                valid = False
                break
            while i < len(s) and s[i] in vowels:
                i += 1
            expecting_consonant = True

    print("True" if valid else "False")
```

Status : Correct

Marks : 10/10

2. Problem Statement

Emily is a data analyst working for a company that collects feedback from customers in the form of text messages. As part of her data validation tasks, Emily needs to perform two operations on each message:

Calculate the sum of all the digits mentioned in the message. If the sum of the digits is greater than 9, check whether the sum forms a palindrome number.

Your task is to help Emily automate this process by writing a program that extracts all digits from a given message, calculates their sum, and checks if the sum is a palindrome if it is greater than 9.

Input Format

The input consists of a string *s*, representing the customer message, which may contain letters, digits, spaces, and other characters.

Output Format

The output prints an integer representing the sum of all digits in the string, followed by a space.

If the sum is greater than 9, print "Palindrome" if the sum is a palindrome, otherwise print "Not palindrome".

If the sum is less than or equal to 9, no palindrome check is required.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 12 books 4 pen

Output: 7

Answer

```
s = input().strip()
```

```
# Calculate the sum of digits in the string
digit_sum = sum(int(ch) for ch in s if ch.isdigit())

# Print the sum
print(digit_sum, end="")

# Check for palindrome if sum > 9
if digit_sum > 9:
    if str(digit_sum) == str(digit_sum)[::-1]:
        print(" Palindrome")
    else:
        print(" Not palindrome")
```

Status : Correct

Marks : 10/10

3. 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 World

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 World

Answer

```
string1 = input()
string2 = input()
char1 = input()
char2 = input()
```

```
modified1 = string1.replace(char1, char2)
modified2 = string2.replace(char1, char2)
```

```
print("{} {}".format(modified1, modified2))
```

Status : Correct

Marks : 10/10