

Rajalakshmi Engineering College

Name: Pranav Shanmugam

Email: 240701395@rajalakshmi.edu.in

Roll no: 2116240701395

Phone: 8925357178

Branch: REC

Department: I CSE FD

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

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()
digits=[]
for char in s:
    if char.isdigit():
        digits.append(int(char))
sum_digits=sum(digits)
output=str(sum_digits)
if sum_digits>9:
    if output==output[::-1]:
        output+="Palindrome"
    else:
        output+="Not palindrome"
print(output)
```

Status : Correct

Marks : 10/10

2. 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 perfect(s):
    vowels="aeiouAEIOU"
    con="bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ"
    if not s or s[0] not in con:
        return False
    seen=set()
    prev=""
    for i,c in enumerate(s):
        if c in con:
            if c in seen or prev in vowels and i>1 and s[i-2] in vowels:
                return False
            seen.add(c)
        prev=c
```

```
    prev=c
    elif c in vowels:
        prev=c
    else:
        return False
    return True

print(perfect(input().strip()))
```

Status : Partially correct

Marks : 5/10

3. Problem Statement

You have two strings str1 and str2, both of equal length.

Write a Python program to concatenate the two strings such that the first character of str1 is followed by the first character of str2, the second character of str1 is followed by the second character of str2, and so on.

For example, if str1 is "abc" and str2 is "def", the output should be "adbef".

Input Format

The input consists of two strings in each line.

Output Format

The output displays the concatenated string in the mentioned format.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: abc

def

Output: adbefc

Answer

```
str1=input().strip()
```

```
str2=input().strip()
result=[]
for c1,c2 in zip(str1,str2):
    result.append(c1)
    result.append(c2)
print("".join(result))
```

Status : Correct

Marks : 10/10