

Rajalakshmi Engineering College

Name: SIVASURIYAN .
Email: 240701518@rajalakshmi.edu.in
Roll no:
Phone: 7010603376
Branch: REC
Department: I CSE FF
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

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

```
# You are using Python
a=input()
b=input()
d=input()
e=input()

print((a+" "+b).replace(d,e))
```

Status : Correct

Marks : 10/10

2. Problem Statement

Gina is working on a data analysis task where she needs to extract sublists from a given list of integers and find the median of each sublist. For each median found, she also needs to determine its negative index in the original list.

Help Gina by writing a program that performs these tasks.

Note: The median is the middle value in the sorted list of numbers, or the first value of the two middle values if the list has an even number of elements.

Example

Input

10

1 2 3 4 5 7 8 9 10 11

3

1 5

2 6

3 10

Output

3 : -8

4 : -7

7 : -5

Explanation

For the first range (1 to 5), the sublist is [1, 2, 3, 4, 5]. The median is 3, and its negative index in the original list is -8.

For the second range (2 to 6), the sublist is [2, 3, 4, 5, 7]. The median is 4, and its negative index in the original list is -7.

For the third range (3 to 10), the sublist is [3, 4, 5, 7, 8, 9, 10, 11]. The median is 7, and its negative index in the original list is -5.

Input Format

The first line of input consists of an integer N, representing the number of elements in the list.

The second line consists of N space-separated integers representing the elements of the list.

The third line consists of an integer R, representing the number of ranges.

The next R lines each consist of two integers separated by space representing the start and end indices (1-based) of the ranges.

Output Format

The output consists of n lines, displaying "X : Y" where X is the median of the sublist and Y is the negative index in the original list.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10

1 2 3 4 5 7 8 9 10 11

3

1 5

2 6

3 10

Output: 3 : -8

4 : -7

7 : -5

Answer

```
a=int(input())
b=list(map(int,input().split()))
m = int(input())
sub = []
for i in range(m):
    start,end = list(map(int,input().split()))
    sub.append((start,end))
for start,end in sub:
    temp = b[start-1:end]
    leng = len(temp)
    temp1 = sorted(temp)
    if(leng % 2 == 0):
        mi = leng//2 - 1
    else:
        mi = leng//2
    mid = temp1[mi]
    ind = b.index(mid)
    ind = a - ind
    print(mid,":",-1*ind)
```

Status : Correct

Marks : 10/10

3. 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):
    vowels = "aeiou"
    consonants_seen = set()

    if s[0] in vowels:
        return False # Must start with consonant

    i = 0
    n = len(s)

    while i < n:
        if s[i] not in vowels: # consonant
            if s[i] in consonants_seen:
                return False # consonant repeated
            consonants_seen.add(s[i])
            i += 1
            if i < n and s[i] not in vowels:
                return False # two consonants in a row
        else:
            # allow consecutive vowels, but check next after run of vowels
            while i < n and s[i] in vowels:
                i += 1
            if i < n and s[i] not in vowels:
```

```
        return False # vowel run immediately followed by consonant

    return True

# Input
s = input()

# Output
print("True" if is_perfect_string(s) else "False")
```

Status : Partially correct

Marks : 5/10