Rajalakshmi Engineering College

Name: madhavan R
Email: 240701294@rajalakshmi.edu.in
Roll no: 240701294
Phone: 6381021871
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 : 20

Section 1 : Coding

1. 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

```
# You are using Python
N=int(input())
lst=list(map(int, input().split()))
R=int(input())
for _ in range(0, R):
    start,end=map(int, input().split())
    sublist=lst[start-1:end]
    sublist_sorted=sorted(sublist)
    length=len(sublist_sorted)
    if length%2==1:
        median=sublist_sorted[length//2]
    else:
        median=sublist_sorted[(length//2)-1]
    X=median
    Y=lst.index(median)-(len(lst))
    print(X, ":", Y)
```

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

```
# You are using Python
def is_palindrome (num):
```

```
return str(num) == str(num)[::-1]
def process_message(message):
    digit_sum = sum(int(char) for char in message if char.isdigit())

    if digit_sum > 9:
        if is_palindrome(digit_sum):
            print("palindrome")
        else:
            print("Not a palindrome")
    else:
        print("Sum is 7")
```

Status : Wrong ***Marks : 0/10***

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

# Replace the placeholder character with the new character
modified1 = string1.replace(char1, char2)
```

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

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

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