Rajalakshmi Engineering College

Name: dakshesh p
Email: 241501038@rajalakshmi.edu.in
Roll no:
Phone: 9176316602
Branch: REC
Department: I AI & ML FA
Batch: 2028
Degree: B.E - AI & ML

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

```
N = int(input())
lst = list(map(int,input().split()))
R = int(input())
for _ in range(R):
    start,end = map(int,input().split())
    sublist = lst[start-1:end]
    sublist.sort()
    length = len(sublist)
    if length % 2 == 1:
        median=sublist[length // 2]
    else:
        median = sublist[length // 2 - 1]
    median_index = lst.index(median)
    negitive_index = -(median_index + 1)
    print(f"{median} : {negtive index}")
```

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

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

```
str1=input()
str2=input()
char1=input()
char2=input()

modified_str1 = ""
for ch in str1:
    if ch==char1:
        modified_str1 += char2
    else:
        modified_str1 += ch

modified_str2 = ""
for ch in str2:
    if ch==char1:
        modified_str2 += char2
    else:
        modified_str2 += ch
print(modified_str1,modified_str2)
```

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

3. Problem Statement

A company is creating email accounts for its new employees. They want to

use a naming convention for email addresses that consists of the first letter of the employee's first name, followed by their last name, followed by @company.com.

The company also has a separate email domain for administrative employees.

Write a program that prompts the user for their first name, last name, role, and company and then generates their email address using the appropriate naming convention based on their role. This is demonstrated in the below examples.

Note:

The generated email address should consist of the first letter of the first name, the last name in lowercase, and a suffix based on the role and company, all in lowercase.

Input Format

The first line of input consists of the first name of an employee as a string.

The second line consists of the last name of an employee as a string.

The third line consists of the role of the employee as a string.

The last line consists of the company name as a string.

Output Format

The output consists of a single line containing the generated email address for the employee, following the specified naming convention.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: John
Smith
admin

iamNeo

Output: jsmith@admin.iamneo.com

Answer

```
a=input()
b=input()
c=input()
d=input()
if(c=="admin"):
   print(f"{a[0].lower()}{b.lower()}@{c}.{d}.com")
else:
   print(f"{a[0].lower()}{b.lower()}@{d}.com")
```

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