

# Rajalakshmi Engineering College

Name: Elango G  
Email: 241501055@rajalakshmi.edu.in  
Roll no: 241501055  
Phone: 7010568330  
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 : 25

### 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())
```

```
d=list(map(int,input().split()))
```

```
r=int(input())
```

```
for _ in range(r):
```

```
    start,end=map(int,input().split())
```

```
    sl=d[start-1:end]
```

```
    sl.sort()
```

```
    le=len(sl)
```

```
    if le % 2==1:
```

```
        median=sl[le//2]
```

```
    else:
```

```
        median=sl[(le//2)-1]
```

```
for i in range(len(d)):
```

```
    if d[i]==median:
```

```
        ne=-(len(d)-i)
```

```
        break
```

```
print(f"{median}:{ne}")
```

**Status : Correct**

**Marks : 10/10**

## 2. Problem Statement

Raja needs a program that helps him manage his shopping list efficiently. The program should allow him to perform the following operations:

**Add Items:** Raja should be able to add multiple items to his shopping list at once. He will input a space-separated list of items, each item being a string.

**Remove Item:** Raja should be able to remove a specific item from his shopping list. He will input the item he wants to remove, and if it exists in the list, it will be removed. If the item is not found, the program should notify him.

**Update List:** Raja might realize he forgot to add some items initially. After removing unnecessary items, he should be able to update his list by adding more items. Similar to the initial input, he will provide a space-separated list of new items.

### ***Input Format***

The first line consists of the initial list of integers should be entered as space-separated values.

The second line consists of the element to be removed should be entered as a single integer value.

The third line consists of the new elements to be appended should be entered as space-separated values.

### ***Output Format***

The output displays the current state of Raja's shopping list after each operation. After adding items, removing items, and updating the list, the program prints the updated shopping list in the following format:

"List1: [element1, element2, ... ,element\_n]

List after removal: [element1, element2, ... ,element\_n]

Final list: [element1, element2, ..., element\_n]".

If the item is not found in the removing item process, print the message "Element not found in the list".

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 1 2 3 4 5

3

6 7 8

Output: List1: [1, 2, 3, 4, 5]

List after removal: [1, 2, 4, 5]

Final list: [1, 2, 4, 5, 6, 7, 8]

### **Answer**

# You are using Python

```
n=list(map(int,input().split()))
```

```
print("List1: ",n)
```

```
rem=int(input())
```

```
if rem in n:
```

```
    n.remove(rem)
```

```
    print("List after removal: ",n)
```

```
elif rem!=n:
```

```
    print("Element not found in the list")
```

```
d=list(map(int,input().split()))
```

```
n.extend(d)
```

```
print("Final List:",n)
```

**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***

```
# You are using Python
s=input().strip()
v=set('aeiou')
c=set('abcdefghijklmnopqrstuvwxyz')-v
```

```
if not s or s[0] not in c:
```

```
    print("False")
```

```
else:
```

```
    sc=set()
```

```
    ev=False
```

```
    i=0
```

```
    ip=True
```

```
    while i<len(s):
```

```
ch=s[i]
if ch in v:
    if not ev:
        ip=False
        break
    while i<len(s) and s[i] in v:
        i+=1
    ev=False
elif ch in c:
    if ev or ch in sc:
        ip=False
        break
    sc.add(ch)
    i+=1
    ev=True
else:
    ip=False
    break
if ip and not ev:
    print("True")
else:
    print("False")
```

**Status :** Partially correct

**Marks :** 5/10