

# Rajalakshmi Engineering College

Name: shiloh .s  
Email: 240701498@rajalakshmi.edu.in  
Roll no: 240701498  
Phone: 9488883273  
Branch: REC  
Department: I CSE FE  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 36.5

### Section 1 : Coding

#### 1. Problem Statement

Riley is analyzing DNA sequences and needs to determine which bases match at the same positions in two given DNA sequences. Each DNA sequence is represented as a tuple of integers, where each integer corresponds to a DNA base.

Your task is to write a program that compares these two sequences and identifies the bases that match at the same positions and print it.

#### ***Input Format***

The first line of input consists of an integer  $n$ , representing the size of the first tuple.

The second line contains  $n$  space-separated integers, representing the elements of the first DNA sequence tuple.

The third line of input consists of an integer m, representing the size of the second tuple.

The fourth line contains m space-separated integers, representing the elements of the second DNA sequence tuple.

### **Output Format**

The output is a space-separated integer of the matching bases at the same positions in both sequences.

Refer to the sample output for format specifications.

### **Sample Test Case**

Input: 4  
5 1 8 4  
4  
4 1 8 2  
Output: 1 8

### **Answer**

```
n=int(input())
a=list(map(int,input().split()))
m=int(input())
b=list(map(int,input().split()))
for i in range(n):
    if a[i]==b[i]:
        print(a[i],end=' ')
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

Samantha is working on a text analysis tool that compares two words to find common and unique letters. She wants a program that reads two words, w1, and w2, and performs the following operations:

Print the letters common to both words, in alphabetical order. Print the letters that are unique to each word, in alphabetical order. Determine if the set of letters in the first word is a superset of the letters in the second word. Check if there are no common letters between the two words and print the result as a Boolean value.

Ensure the program ignores case differences and leading/trailing spaces in the input words.

Your task is to help Samantha in implementing the same.

### ***Input Format***

The first line of input consists of a string representing the first word, w1.

The second line consists of a string representing the second word, w2.

### ***Output Format***

The first line of output should display the sorted letters common to both words, printed as a list.

The second line should display the sorted letters that are unique to each word, printed as a list.

The third line should display a Boolean value indicating if the set of letters in w1 is a superset of the set of letters in w2.

The fourth line should display a Boolean value indicating if there are no common letters between w1 and w2.

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: program

Peace

Output: ['a', 'p']

['c', 'e', 'g', 'm', 'o', 'r']

False

False

**Answer**

```
n=input().lower()
a=set(n)
m=input().lower()
b=set(m)
print(sorted(a&b))
print(sorted(a^b))
if a>=b:
    print("True")
else:
    print("False")
if (a&b)==set():
    print("True")
else:
    print("False")
```

**Status :** Partially correct**Marks :** 7.5/10**3. Problem Statement**

Alex is tasked with managing the membership lists of several exclusive clubs. Each club has its own list of members, and Alex needs to determine the unique members who are part of exactly one club when considering all clubs together.

Your goal is to help Alex by writing a program that calculates the symmetric difference of membership lists from multiple clubs and then finds the total number of unique members.

**Input Format**

The first line of input consists of an integer  $k$ , representing the number of clubs.

The next  $k$  lines each contain a space-separated list of integers, where each integer represents a member's ID.

**Output Format**

The first line of output displays the symmetric difference of the membership lists as a set.

The second line displays the sum of the elements in this symmetric difference.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 3

1 2 3

2 3 4

5 6 7

Output: {1, 4, 5, 6, 7}

23

### **Answer**

```
a=int(input())
c=set()
for i in range(a):
    b=list(map(int,input().split()))
    c=c^set(b)
print(c)
print(sum(c))
```

**Status :** Partially correct

**Marks :** 9/10

## **4. Problem Statement**

James is an engineer working on designing a new rocket propulsion system. He needs to solve a quadratic equation to determine the optimal launch trajectory. The equation is of the form  $ax^2 + bx + c = 0$ .

Your task is to help James find the roots of this quadratic equation. Depending on the discriminant, the roots might be real and distinct, real and equal, or complex. Implement a program to determine and display the roots of the equation based on the given coefficients.

### **Input Format**

The first line of input consists of an integer N, representing the number of coefficients.

The second line contains three space-separated integers a,b, and c representing the coefficients of the quadratic equation.

### **Output Format**

The output displays:

1. If the discriminant is positive, display the two real roots.
2. If the discriminant is zero, display the repeated real root.
3. If the discriminant is negative, display the complex roots as a tuple with real and imaginary parts.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 3

1 5 6

Output: (-2.0, -3.0)

### **Answer**

```
import math
```

```
N = int(input())
```

```
a, b, c = map(int, input().split())
```

```
D = b**2 - 4*a*c
```

```
if D > 0:
```

```
    r1 = (-b + math.sqrt(D)) / (2*a)
```

```
    r2 = (-b - math.sqrt(D)) / (2*a)
```

```
    print((r1, r2))
```

```
elif D == 0:
```

```
    r = -b / (2*a)
```

```
    print((r, r))
```

```
else:
```

```
    real = -b / (2*a)
```

```
    imag = math.sqrt(-D) / (2*a)
```

```
print(((real, imag), (real, -imag)))
```

**Status :** Correct

**Marks :** 10/10