Rajalakshmi Engineering College

Name: I Mohammed Hamza
Email: 240701326@rajalakshmi.edu.in
Roll no: 240701326
Phone: 7358328592
Branch: REC
Department: I CSE AH
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 : 37.5

Section 1 : Coding

1. 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

```
def solve():
    k = int(input())
    sets = [set(map(int, input().split())) for _ in range(k)]
    symmetric_diff = sets[0]

    for s in sets[1:]:
        symmetric_diff ^= s

    print(symmetric_diff)
    print(sum(symmetric_diff))

solve()
```

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

2. Problem Statement

Emily is a librarian who keeps track of books borrowed and returned by her patrons. She maintains four sets of book IDs: the first set represents books borrowed, the second set represents books returned, the third set represents books added to the collection, and the fourth set represents books that are now missing. Emily wants to determine which books are still borrowed but not returned, as well as those that were added but are now missing. Finally, she needs to find all unique book IDs from both results.

Help Emily by writing a program that performs the following operations on four sets of integers:

Compute the difference between the borrowed books (first set) and the returned books (second set).Compute the difference between the added books (third set) and the missing books (fourth set).Find the union of the results from the previous two steps, and sort the final result in descending order.

Input Format

The first line of input consists of a list of integers representing borrowed books.

The second line of input consists of a list of integers representing returned books.

The third line of input consists of a list of integers representing added books.

The fourth line of input consists of a list of integers representing missing books.

Output Format

The first line of output displays the difference between sets P and Q, sorted in descending order.

The second line of output displays the difference between sets R and S, sorted in descending order.

The third line of output displays the union of the differences from the previous two steps, sorted in descending order.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3
2 3 4
5 6 7
6 7 8

Output: [1]
[5]
[5, 1]

Answer

```
def solve():
    borrowed = set(map(int, input().split()))
    returned = set(map(int, input().split()))
    added = set(map(int, input().split()))
    missing = set(map(int, input().split()))

    borrowed_not_returned = sorted(borrowed - returned, reverse=True)
    added_not_missing = sorted(added - missing, reverse=True)
    final_result = sorted(borrowed_not_returned + added_not_missing,
reverse=True)

    print(borrowed_not_returned)
    print(added_not_missing)
    print(final_result)

solve()
```

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

3. 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 ax2 +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 cmath

def solve():
    N = int(input())
    a, b, c = map(int, input().split())

    if a == 0:
        if b == 0:
            if c == 0:
                print("Infinite solutions")
            else:
                print("No solution")
        else:
            root = -c / b
            print((round(root, 1),))
        return

    d = b**2 - 4*a*c
    sqrt_d = cmath.sqrt(d)
    root1 = (-b + sqrt_d) / (2 * a)
    root2 = (-b - sqrt_d) / (2 * a)

    if d > 0:
        # Two distinct real roots
        result = (round(root1.real, 1), round(root2.real, 1))
    elif d == 0:
        # One real root (repeated)
        result = (round(root1.real, 1),)
    else:
        # Two complex roots
        result = (
            (round(root1.real, 1), round(root1.imag, 1)),
            (round(root2.real, 1), round(root2.imag, 1))
        )

    print(result)

solve()
```

Status : Partially correct ***Marks : 7.5/10***

4. Problem Statement

Alex is working with grayscale pixel intensities from an old photo that has been scanned in a single row. To detect edges in the image, Alex needs to calculate the differences between each pair of consecutive pixel intensities.

Your task is to write a program that performs this calculation and returns the result as a tuple of differences.

Input Format

The first line of input contains an integer n, representing the number of pixel intensities.

The second line contains n space-separated integers representing the pixel intensities.

Output Format

The output displays a tuple containing the absolute differences between consecutive pixel intensities.

Refer to the sample output for format specifications.

Sample Test Case

Input: 5
200 100 20 80 10
Output: (100, 80, 60, 70)

Answer

```
def solve():
    n = int(input())
    pixels = list(map(int, input().split()))
    differences = tuple(abs(pixels[i] - pixels[i + 1]) for i in range(n - 1))
    print(differences)

solve()
```

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