

# Rajalakshmi Engineering College

Name: Alvin .B

Email: 240701034@rajalakshmi.edu.in

Roll no: 240701034

Phone: 9677000577

Branch: REC

Department: I CSE FA

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 : 39

### Section 1 : Coding

#### 1. Problem Statement

Noah, a global analyst at a demographic research firm, has been tasked with identifying which country experienced the largest population growth over a two-year period. He has a dataset where each entry consists of a country code and its population figures for two consecutive years. Noah needs to determine which country had the highest increase in population and present the result in a specific format.

Help Noah by writing a program that outputs the country code with the largest population increase, along with the increase itself.

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

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

Each of the following N blocks contains three lines:

1. The first line is a country code.
2. The second line is an integer representing the population of the country in the first year.
3. The third line is an integer representing the population of the country in the second year.

### **Output Format**

The output displays the country code and the population increase in the format {code: difference}, where code is the country code and difference is the increase in population.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 3

01

1000

1500

02

2000

2430

03

1500

3000

Output: {03:1500}

### **Answer**

# You are using Python

```
def find_largest_population_growth(n, data):
```

```
    max_growth = 0
```

```
    country_code = ""
```

```
    for i in range(n):
```

```
        code = data[i][0]
```

```
        population_year_1 = data[i][1]
```

```
        population_year_2 = data[i][2]
```

```

        growth = population_year_2 - population_year_1
        if growth > max_growth:
            max_growth = growth
            country_code = code

    return country_code, max_growth

def main():
    n = int(input())
    data = []
    for _ in range(n):
        code = input().strip()
        population_year_1 = int(input().strip())
        population_year_2 = int(input().strip())
        data.append((code, population_year_1, population_year_2))
    country_code, max_growth = find_largest_population_growth(n, data)
    print(f"{{{country_code}:{max_growth}}}")
if __name__ == "__main__":
    main()

```

**Status :** Correct

**Marks :** 10/10

## 2. 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**

# You are using Python

```
n = int(input())
```

```
sequence1 = list(map(int, input().split()))
```

```
m = int(input())
```

```
sequence2 = list(map(int, input().split()))
```

```
min_length = min(n, m)
```

```
matching_bases = []
```

```
for i in range(min_length):
```

```
    if sequence1[i] == sequence2[i]:
```

```
        matching_bases.append(sequence1[i])
```

```
print(' '.join(map(str, matching_bases)))
```

**Status : Correct**

**Marks : 10/10**

### **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***

# You are using Python

```
k = int(input())
```

```
symmetric_diff = set()
```

```
for _ in range(k):
    club_members = set(map(int, input().split()))
    symmetric_diff = symmetric_diff.symmetric_difference(club_members)
    print(symmetric_diff)
    print(sum(symmetric_diff))
```

**Status :** Partially correct

**Marks :** 9/10

#### 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***

# You are using Python

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)

**Status :** Correct

**Marks :** 10/10