

# Rajalakshmi Engineering College

Name: bhagawath narayanan n  
Email: 241501034@rajalakshmi.edu.in  
Roll no: 241501034  
Phone: 6374835866  
Branch: REC  
Department: I AIML AD  
Batch: 2028  
Degree: B.E - AI & ML

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

Riya owns a store and keeps track of item prices from two different suppliers using two separate dictionaries. He wants to compare these prices to identify any differences. Your task is to write a program that calculates the absolute difference in prices for items that are present in both dictionaries. For items that are unique to one dictionary (i.e., not present in the other), include them in the output dictionary with their original prices.

Help Riya to implement the above task using a dictionary.

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

The first line of input consists of an integer  $n_1$ , representing the number of items in the first dictionary.

The next n1 lines contain two integers

1. The first line contains the item (key), and
2. The second line contains the price (value).

The following line consists of an integer n2, representing the number of items in the second dictionary

The next n2 lines contain two integers

1. The first line contains the item (key), and
2. The second line contains the price (value).

### **Output Format**

The output should display a dictionary that includes:

1. For items common to both dictionaries, the absolute difference between their prices.
2. For items that are unique to one dictionary, the original price from that dictionary.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 1

4

4

1

8

7

Output: {4: 4, 8: 7}

### **Answer**

# You are using Python

```
def read_dictionary():
```

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

```
    d = {}
```

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

```

    key = int(input())
    value = int(input())
    d[key] = value
    return d
dict1 = read_dictionary()
dict2 = read_dictionary()
result_dict = {}
for key, value1 in dict1.items():
    if key in dict2:
        value2 = dict2[key]
        result_dict[key] = abs(value1 - value2)
    else:
        result_dict[key] = value1
for key, value2 in dict2.items():
    if key not in dict1:
        result_dict[key] = value2

print(result_dict)

```

**Status :** Correct

**Marks :** 10/10

## 2. 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())
```

```
pixel_intensities_str = input().split()
```

```
pixel_intensities = [int(p) for p in pixel_intensities_str]
```

```
differences = []
```

```
for i in range(n - 1):
```

```
    diff = abs(pixel_intensities[i + 1] - pixel_intensities[i])
```

```
    differences.append(diff)
```

```
result_tuple = tuple(differences)
```

```
print(result_tuple)
```

**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())
```

```
first_line = input().split()
```

```
symmetric_diff_result = set(map(int, first_line))
```

```
for _ in range(k - 1):
```

```
    members = set(map(int, input().split()))
```

```
    symmetric_diff_result =
```

```
symmetric_diff_result.symmetric_difference(members)
```

```
print(symmetric_diff_result)
```

```
print(sum(symmetric_diff_result))
```

**Status :** Partially correct

**Marks :** 9/10

#### 4. 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())
dna1_elements = list(map(int, input().split()))
dna1_tuple = tuple(dna1_elements)

m = int(input())
dna2_elements = list(map(int, input().split()))
dna2_tuple = tuple(dna2_elements)

comparison_length = min(n, m)

matching_bases = []

for i in range(comparison_length):
    if dna1_tuple[i] == dna2_tuple[i]:
        matching_bases.append(str(dna1_tuple[i]))

print(" ".join(matching_bases))
```

**Status :** Correct

**Marks :** 10/10