

Rajalakshmi Engineering College

Name: Saii Anish R
Email: 241501175@rajalakshmi.edu.in
Roll no: 241501175
Phone: 8438920387
Branch: REC
Department: I AI & ML FB
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

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

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

```
member_sets = []
```

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

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

```
    member_sets.append(members)
```

```
symmetric_diff = member_sets[0]
```

```
for i in range(1, k):
```

```
    symmetric_diff = symmetric_diff.symmetric_difference(member_sets[i])
```

```
print(symmetric_diff)
```

```
print(sum(symmetric_diff))
```

Status : Partially correct

Marks : 9/10

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

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

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

```
diffs = tuple(abs(pixels[i] - pixels[i+1]) for i in range(n - 1))
```

```
print(diffs)
```

Status : Correct

Marks : 10/10

3. 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

```
N = int(input())

max_increase = -1
max_country = ""

for _ in range(N):
    code = input().strip()
    pop_year1 = int(input())
    pop_year2 = int(input())

    increase = pop_year2 - pop_year1
    if increase > max_increase:
        max_increase = increase
        max_country = code

print(f"{{{max_country}}}:{max_increase}")
```

Status : Correct

Marks : 10/10

4. 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 n_1 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 n_2 , representing the number of items in the second dictionary

The next n_2 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

```
def compare_prices(dict1, dict2):  
    comparison_dict = {}
```

```
for key in dict1:
    if key in dict2:
        comparison_dict[key] = abs(dict1[key] - dict2[key])
    else:
        comparison_dict[key] = dict1[key]

for key in dict2:
    if key not in dict1:
        comparison_dict[key] = dict2[key]
return comparison_dict

if __name__ == "__main__":
    n1 = int(input())
    dict1 = {}
    for _ in range(n1):
        item = int(input())
        price = int(input())
        dict1[item] = price
    n2 = int(input())
    dict2 = {}
    for _ in range(n2):
        item = int(input())
        price = int(input())
        dict2[item] = price
    result = compare_prices(dict1, dict2)
    print(result)
```

Status : Correct

Marks : 10/10