Rajalakshmi Engineering College

Name: Monica B
Email: 240701330@rajalakshmi.edu.in
Roll no: 240701330
Phone: 6385195950
Branch: REC
Department: I CSE FC
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 : 40

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

```
N = int(input())
max_increase = -1
max_country = ""
for _ in range(N):
    code = input().strip()
    pop1 = int(input())
    pop2 = int(input())
    increase = pop2 - pop1
    if increase > max_increase:
        max_increase = increase
        max_country = code
print(f"{{{max_country}:{max_increase}}}")
```

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

```
n = int(input())
seq1 = list(map(int, input().split()))
m = int(input())
seq2 = list(map(int, input().split()))
min_len = min(n, m)
matches = []
for i in range(min_len):
    if seq1[i] == seq2[i]:
        matches.append(str(seq1[i]))
print(" ".join(matches))
```

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

3. 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 n1, 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

```
n1 = int(input())
d1 = {}
order = []
for _ in range(n1):
    key = int(input())
    value = int(input())
    d1[key] = value
    if key not in order:
        order.append(key)
```

```
n2 = int(input())
d2 = {}
for _ in range(n2):
   key = int(input())
   value = int(input())
   d2[key] = value
   if key not in order:
      order.append(key)
result = {}
for key in order:
   if key in d1 and key in d2:
      result[key] = abs(d1[key] - d2[key])
   elif key in d1:
      result[key] = d1[key]
   else:
      result[key] = d2[key]
print(result)
```

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

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

```
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***