Rajalakshmi Engineering College

Name: aakash velan
Email: 241501001@rajalakshmi.edu.in
Roll no:
Phone: 7358618861
Branch: REC
Department: I AI & ML FA
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 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())
dict1 = {}
for _ in range(n1):
    key = int(input())
    value = int(input())
    dict1[key] = value

# Read second dictionary
n2 = int(input())
dict2 = {}
for _ in range(n2):
    key = int(input())
    value = int(input())
    dict2[key] = value

# Create the result dictionary
result = {}

# Keys in both dictionaries
for key in dict1:
    if key in dict2:
        result[key] = abs(dict1[key] - dict2[key])
    else:
        result[key] = dict1[key]

# Keys only in dict2
for key in dict2:
    if key not in dict1:
        result[key] = dict2[key]

# Print the result
print(result)
```

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

```
# Read input lines and convert them to sets
P = set(map(int, input().split()))  # Borrowed books
Q = set(map(int, input().split()))  # Returned books
R = set(map(int, input().split()))  # Added books
S = set(map(int, input().split()))  # Missing books

# Compute differences
borrowed_not_returned = sorted(P - Q, reverse=True)
added_not_missing = sorted(R - S, reverse=True)

# Compute union of the differences
final_result = sorted(set(borrowed_not_returned + added_not_missing),
reverse=True)

# Print results
print(borrowed_not_returned)
print(added_not_missing)
print(final_result)
```

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

3. 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

```
# Read number of pixel intensities
n = int(input())

# Read pixel intensities
pixels = list(map(int, input().split()))

# Calculate absolute differences between consecutive pixels
differences = tuple(abs(pixels[i] - pixels[i + 1]) for i in range(n - 1))

# Print the result as a tuple
print(differences)
```

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

4. 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

```
# Read number of clubs
k = int(input())
if k==0:
    print(0)
    exit()
```

```
# Read membership lists for each club and convert them to sets
club_sets = []
for _ in range(k):
    members = set(map(int, input().split()))
    club_sets.append(members)

# Compute the symmetric difference across all club sets
sym_diff = club_sets[0]
for s in club_sets[1:]:
    sym_diff = sym_diff.symmetric_difference(s)

# Print the symmetric difference set and the sum of its elements
print(sym_diff)
print(sum(sym_diff))
```

Status : Partially correct ***Marks : 9/10***