Rajalakshmi Engineering College

Name: RAKSHITHA R
Email: 240701418@rajalakshmi.edu.in
Roll no: 240701418
Phone: 7305274265
Branch: REC
Department: I CSE FD
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

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

```
# You are using Python
# Read input lists
```

```
borrowed_books = list(map(int, input().split()))
returned_books = list(map(int, input().split()))
added_books = list(map(int, input().split()))
missing_books = list(map(int, input().split()))
borrowed_set = set(borrowed_books)
returned_set = set(returned_books)
added_set = set(added_books)
missing_set = set(missing_books)
borrowed_not_returned = sorted(borrowed_set - returned_set, reverse=True)
added_but_missing = sorted(added_set - missing_set, reverse=True)
unique_books = sorted(set(borrowed_not_returned) | set(added_but_missing),
reverse=True)
print(borrowed_not_returned)
print(added_but_missing)
print(unique_books)
```

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

2. 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
n=int(input())
high=0
for i in range(n):
    code=input()
    ini=int(input())
    final=int(input())
    res=final-ini
    if res>high:
        d={}
        high=res
        d[code]=res
print("{"+",".join(f'{i}:{j}'for i,j in d.items())+'}')
```

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

```
# You are using Python
n1=int(input())
di1={}
for i in range(n1):
    key=int(input())
    value=int(input())
    di1[key]=value
n2=int(input())
di2={}
for a in range(n2):
    key=int(input())
    value=int(input())
    di2[key]=value
result={}
for key in di1:
    if key in di2:
        result[key]=abs(di1[key]-di2[key])
    else:
        result[key]=di1[key]
for key in di2:
    if key not in di1:
        result[key]=di2[key]
print(result)
```

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

4. Problem Statement

Samantha is working on a text analysis tool that compares two words to find common and unique letters. She wants a program that reads two

words, w1, and w2, and performs the following operations:

Print the letters common to both words, in alphabetical order.Print the letters that are unique to each word, in alphabetical order.Determine if the set of letters in the first word is a superset of the letters in the second word.Check if there are no common letters between the two words and print the result as a Boolean value.

Ensure the program ignores case differences and leading/trailing spaces in the input words.

Your task is to help Samantha in implementing the same.

Input Format

The first line of input consists of a string representing the first word, w1.

The second line consists of a string representing the second word, w2.

Output Format

The first line of output should display the sorted letters common to both words, printed as a list.

The second line should display the sorted letters that are unique to each word, printed as a list.

The third line should display a Boolean value indicating if the set of letters in w1 is a superset of the set of letters in w2.

The fourth line should display a Boolean value indicating if there are no common letters between w1 and w2.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: program
Peace
Output: ['a', 'p']
['c', 'e', 'g', 'm', 'o', 'r']

False
False

Answer

```
# You are using Python
a=input().strip().lower()
b=input().strip().lower()
set1=set(a)
set2=set(b)
common=sorted(set1 & set2)
print(common)
unique=sorted(set1 ^ set2)
print(unique)
print(set2.issubset(set1))
print(set1.isdisjoint(set2))
```

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