

Rajalakshmi Engineering College

Name: Mageshwaran M
Email: 240701299@rajalakshmi.edu.in
Roll no: 240701299
Phone: 8148528926
Branch: REC
Department: I CSE AH
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

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

```
def compare_supplier_prices():  
    # Reading first dictionary  
    n1 = int(input())  
    dict1 = {}  
    for _ in range(n1):
```

```

key = int(input())
value = int(input())
dict1[key] = value
n2 = int(input())
dict2 = {}
for _ in range(n2):
    key = int(input())
    value = int(input())
    dict2[key] = value
result = {}
all_keys = list(dict1.keys()) + [key for key in dict2 if key not in dict1]
for key in all_keys:
    if key in dict1 and key in dict2:
        result[key] = abs(dict1[key] - dict2[key])
    elif key in dict1:
        result[key] = dict1[key]
    else:
        result[key] = dict2[key]

print(result)
compare_supplier_prices()

```

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())
s=input().split()
l=len(s)
for i in range(l):
    s[i]=int(s[i])
v=1
ls=[]
while v<l:
    ls.append(abs(s[v]-s[v-1]))
    v+=1
t=tuple(ls)
print(t)
```

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
# Reading the number of clubs
k = int(input())
symmetric_diff = set()
for i in range(k):
    club_members = set(map(int, input().split()))
    if i == 0:
        symmetric_diff = club_members
    else:
        symmetric_diff ^= club_members
```

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

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

```
def text_analysis_tool():  
    # Reading input words and cleaning them  
    w1 = input().strip().lower()  
    w2 = input().strip().lower()  
    set1 = set(w1)  
    set2 = set(w2)  
    common_letters = sorted(list(set1 & set2))  
    unique_letters = sorted(list((set1 - set2) | (set2 - set1)))  
    is_superset = set1.issuperset(set2)  
    no_common = set1.isdisjoint(set2)  
    print(common_letters)  
    print(unique_letters)  
    print(is_superset)  
    print(no_common)  
text_analysis_tool()
```

Status : Correct

Marks : 10/10