

# Rajalakshmi Engineering College

Name: Karthik Sah E  
Email: 241501080@rajalakshmi.edu.in  
Roll no: 241501080  
Phone: 8610689556  
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 : 40

#### Section 1 : Coding

##### 1. 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()
```

```
c=set(a)
```

```
d=set(b)
```

```
e=sorted(c&d)
```

```
print(e)
```

```
print(sorted((c^d)))
```

```
print(c.issuperset(d))
```

```
print(c.isdisjoint(d))
```

**Status :** Correct

**Marks :** 10/10

## 2. 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**

```
n1 = int(input())
dict1 = {}
order = []
for _ in range(n1):
    key = int(input())
    value = int(input())
    dict1[key] = value
    order.append(key)

n2 = int(input())
dict2 = {}
for _ in range(n2):
    key = int(input())
    value = int(input())
    dict2[key] = value
    if key not in dict1:
        order.append(key)

result = {}
for key in order:
    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)
```

**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**

```
k = int(input())
sets = [set(map(int, input().split())) for _ in range(k)]

unique_members = set()
all_members = set()

for s in sets:
    for member in s:
        if member in all_members:
            unique_members.discard(member)
        else:
            unique_members.add(member)
            all_members.add(member)

print(unique_members)
print(sum(unique_members))
```

**Status :** Correct

**Marks : 10/10**

#### 4. 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***

# You are using Python

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

```
seq1 = tuple(map(int, input().split()))
```

```
m = int(input())
```

```
seq2 = tuple(map(int, input().split()))
```

```
matches = [str(seq1[i]) for i in range(min(n, m)) if seq1[i] == seq2[i]]
```

```
print(" ".join(matches))
```

**Status :** Correct

**Marks :** 10/10