Dashboard / My courses / CD19411-PPD-2022 / WEEK_09-Set / WEEK-09_CODING

Started on	Friday, 17 May 2024, 12:32 PM
State	Finished
Completed on	Monday, 20 May 2024, 10:17 PM
Time taken	3 days 9 hours
Marks	5.00/5.00
Grade	**50.00** out of 50.00 (**100**%)
Name	JUNIDE CHRIS A 2022-CSD-A

Question **1**

Correct

Mark 1.00 out of 1.00

write a program to identify the common item present in three different set but not on the other set and display the items in the sorted order.

input:

10 50 40 60 30

40 30 70 60 30

20 50 10 75 80

output:

20 70 75 80

Answer: (penalty regime: 0 %)

```
def unique_elements(set1, set2, set3):
    # Create a dictionary to count occurrences
    count = {}

    # Count occurrences in set1
    for num in set1:
        if num in count:
            count[num] += 1
        else:
            count[num] = 1

    # Count occurrences in set2
    for num in set2:
        if num in count:
            count[num] += 1
        else:
            count[num] = 1

    # Count occurrences in set3
    for num in set3:
        if num in count:
            count[num] += 1
```

	Test	Input	Expected	Got	

✓	1	{10,50,40,60,30}
		{40,30,70,60,65}
		{20,50,10,75,80}
			{20,65,70,75,80}	{20,65,70,75,80}	✓

✓	2	{10,15,20,40,50}
		{30,20,40,10,25}
		{40,50,10,45,55}
			{15,25,30,45,55}	{15,25,30,45,55}	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Mr.Harish is maintaining a phone directory which stores phone numbers. He will update the directory with phone numbers every week. While entering the input the number should not be stored inside if the phone number already exists. Finally he want his phone number to be printed in ascending order

Input: n – A1 array size and m – A2 arraysize

Array A1 containing phone numbers already existing and Array A2 containing numbers to be inserted

Ouput : Phone numbers printed in ascending order

Sample Test Case

Input

5

6

9840403212 9890909012 98123455 90123456 99123456

90909090 99999999 9840403212 12345678 12347890 99123456

Output

12345678 12347890 90123456 90909090 98123455 99123456 99999999 9840403212 9890909012

Answer: (penalty regime: 0 %)

```python
def print_phone_numbers(A1, A2):
    phone_set = set(A1)
    for number in A2:
        if number not in phone_set:
            phone_set.add(number)
    sorted_numbers = sorted(phone_set, key=lambda x: int(x))  # Sorting the phone numbers
    print(" ".join(sorted_numbers))

# Sample Test Case
A1_size = int(input())
A2_size = int(input())

A1 = input().split()[:A1_size]
A2 = input().split()[:A2_size]

print_phone_numbers(A1, A2)

```

	Input	Expected	Got	
✓	3 3 9876543211 1122334455 6677889911 6677889911 9876543211 4455667788	1122334455 4455667788 6677889911 9876543211	1122334455 4455667788 6677889911 9876543211	✓

	Input	Expected	Got	
✓	5 6 9840403212 9890909012 98123455 90123456 99123456 90909090 99999999 9840403212 12345678 12347890 99123456	12345678 12347890 90123456 90909090 98123455 99123456 99999999 9840403212 9890909012	12345678 12347890 90123456 90909090 98123455 99123456 99999999 9840403212 9890909012	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Given two lists, print all the common element of two lists.

Note: Sort the list before printing.

Examples:

```
Input :
1 2 3 4 5
5 6 7 8 9
Output :
5

Input :
1 2 3 4 5
6 7 8 9
Output :
No common elements

Input :
1 2 3 4 5 6
5 6 7 8 9
Output :
5 6
```

Answer: (penalty regime: 0 %)

```python
def dupli(str):
    list=[]
    str_list=str.split(" ")
    for i in str_list:
        temp=int(i)
        list.append(temp)
    return list
str1=input()
list1=dupli(str1)
str2=input()
list2=dupli(str2)
flag=0
for e in list1:
    if e in list2:
        print(e,end=" ")
        flag=1
if flag==0:
    print("No common elements")
```

	Input	Expected	Got	
✓	1 2 3 4 5 5 6 7 8 9	5	5	✓
✓	1 2 3 4 5 6 7 8 9	No common elements	No common elements	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Two strings, *a* and *b*, are called anagrams if they contain all the same characters in the same frequencies. For example, the anagrams of CAT are CAT, ACT, TAC, TCA, ATC, and CTA.

Complete the function in the editor. If *a* and *b* are case-insensitive anagrams, print "Anagrams"; otherwise, print "Not Anagrams" instead.

Input Format

The first line contains a string denoting *a*.
The second line contains a string denoting *b*.

Constraints

- *1 ≤ length(a), length(b) ≤ 50*
- Strings *a* and *b* consist of English alphabetic characters.
- The comparison should NOT be case sensitive.

Output Format

Print "Anagrams" if *a* and *b* are case-insensitive anagrams of each other; otherwise, print "Not Anagrams" instead.

Sample Input 0

anagram

margana

Sample Output 0

Anagrams

Explanation 0

Character	**Frequency:** anagram	**Frequency:** margana
A or a	3	3
G or g	1	1
N or n	1	1
M or m	1	1
R or r	1	1

The two strings contain all the same letters in the same frequencies, so we print "Anagrams".

Answer: (penalty regime: 0 %)

```
a = set(input().lower())
b = set(input().lower())
if a == b:
     print("Anagrams")
else:
    print("Not Anagrams")
```

	Input	Expected	Got	
✓	madam maDaM	Anagrams	Anagrams	✓
✓	DAD DAD	Anagrams	Anagrams	✓
✓	MAN MAM	Not Anagrams	Not Anagrams	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

A number is stable if each digit occur the same number of times.i.e, the frequency of each digit in the number is the same. For e.g. 2277,4004,11,23,583835,1010 are examples for stable numbers.
Similarly, a number is unstable if the frequency of each digit in the number is NOT same.

Sample Input:

2277

Sample Output:

Stable Number

Sample Input 2:

121

Sample Output 2:

Unstable Number

Answer: (penalty regime: 0 %)

```
num = input()
digit_freq = {}
for digit in num:
    if digit.isdigit():
        digit_freq[digit] = digit_freq.get(digit, 0) + 1
if len(set(digit_freq.values())) == 1:
    print("Stable Number")
else:
    print("Unstable Number")
```

	Input	Expected	Got	
✓	9988	Stable Number	Stable Number	✓
✓	12	Stable Number	Stable Number	✓
✓	455	Unstable Number	Unstable Number	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◄ Week-09_MCQ

Jump to...

WEEK-09-Extra ►