

Started on	Saturday, 4 May 2024, 5:49 PM
State	Finished
Completed on	Saturday, 4 May 2024, 8:31 PM
Time taken	2 hours 41 mins
Marks	5.00/5.00
Grade	50.00 out of 50.00 (100%)
Name	HARINI V 2022-CSD-A

Question 1

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

- $1 \leq \text{length}(a), \text{length}(b) \leq 50$
- 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 %)

```
1 s1=input().lower()
2 s2=input().lower()
3 if sorted(s1)==sorted(s2):
4     print("Anagrams")
5 else:
6     print("Not Anagrams")
7
```

	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 **2**

Correct

Mark 1.00 out of 1.00

Given a sorted linked list, delete all duplicates such that each element appear only *once*.

Example 1:

Input:

1 1 2

Output:

1 2

Example 2:

Input:

1 1 2 3 3

Output:

1 2 3

Answer: (penalty regime: 0 %)

```
1 List=list(map(int,input().split(" ")))
2 List=list(set(List))
3 List.sort()
4 for i in List:
5     print(i,end=" ")
```

	Test	Input	Expected	Got	
✓	1	1 1 2	1 2	1 2	✓
✓	2	1 1 2 3 3	1 2 3	1 2 3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

You are given an array of N integers, A1, A2, . . . , AN and an integer K. Return the of count of distinct numbers in all windows of size K.

Input :

1 2 1 3 4 3

3

Output :

2

3

3

2

Explanation

All windows of size K are

[1, 2, 1]

[2, 1, 3]

[1, 3, 4]

[3, 4, 3]

Answer: (penalty regime: 0 %)

```

1 n=input()
2 n=n.split(" ")
3 list1=[]
4 for li in n:
5     list1.append(int(li))
6 k=int(input())
7 length=len(list1)
8 for i in range(length-k+1):
9     arr=[]
10    for j in range(0,k):
11        arr.append(list1[i+j])
12    print(len(set(arr)),sep=" ")

```

	Input	Expected	Got	
✓	1 2 1 3 4 3 3	2 3 3 2	2 3 3 2	✓

Question 4

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 %)

```

1 def return_list(str1):
2     str1=str1.replace("{"," ")
3     str1=str1.replace("}","")
4     l=str1.split(",")
5     list1=[]
6     for ele in l:
7         list1.append(int(ele))
8     return list1
9 list1=input()
10 list2=input()
11 list3=input()
12 result=[]
13 list_of_list=[]
14
15 list_of_list.append(return_list(list1))
16 list_of_list.append(return_list(list2))
17
18 list_of_list.append(return_list(list3))
19
20 for j in list_of_list:
21     for i in j:
22         x=list_of_list[0].count(i)
23         x+=list_of_list[1].count(i)
24         x+=list_of_list[2].count(i)
25         if x==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 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 %)

```

1 def check_stability(number):
2     digit_count = {}
3     for digit in str(number):
4         if digit in digit_count:
5             digit_count[digit] += 1
6         else:
7             digit_count[digit] = 1
8
9
10    frequencies = set(digit_count.values())
11    if len(frequencies) == 1:
12        return "Stable Number"
13    else:
14        return "Unstable Number"
15 n=int(input())
16 print(check_stability(n))
17

```

	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.