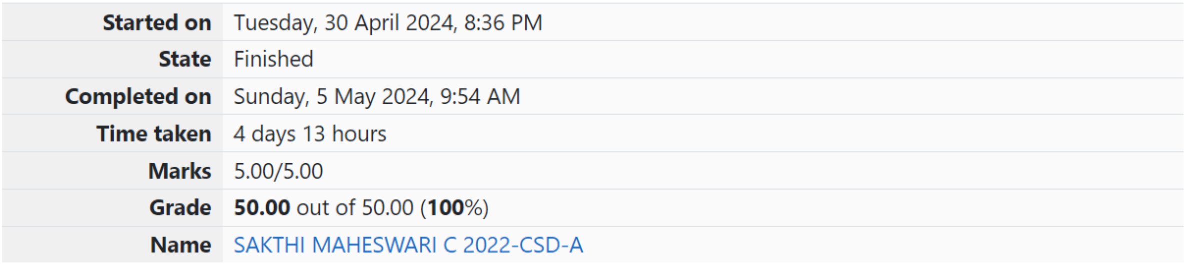
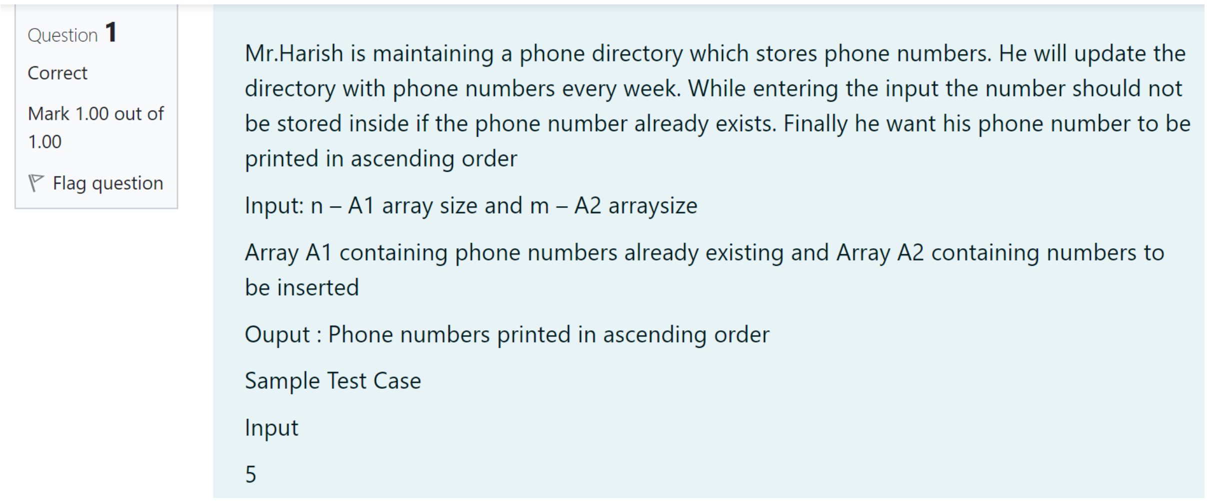
WEEK_09-Set



Started on	Tuesday, 30 April 2024, 8:36 PM
State	Finished
Completed on	Sunday, 5 May 2024, 9:54 AM
Time taken	4 days 13 hours
Marks	5.00/5.00
Grade	**50.00** out of 50.00 (**100**%)
Name	SAKTHI MAHESWARI C 2022-CSD-A

WEEK_09-Set



Question **1**

Correct

Mark 1.00 out of 1.00

Flag question

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

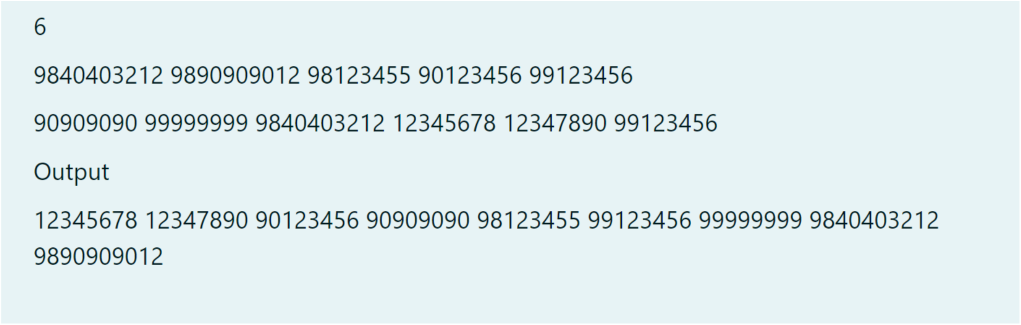
Array A1 containing phone numbers already existing and Array A2 containing numbers to be inserted

Ouput : Phone numbers printed in ascending order

Sample Test Case

Input

5



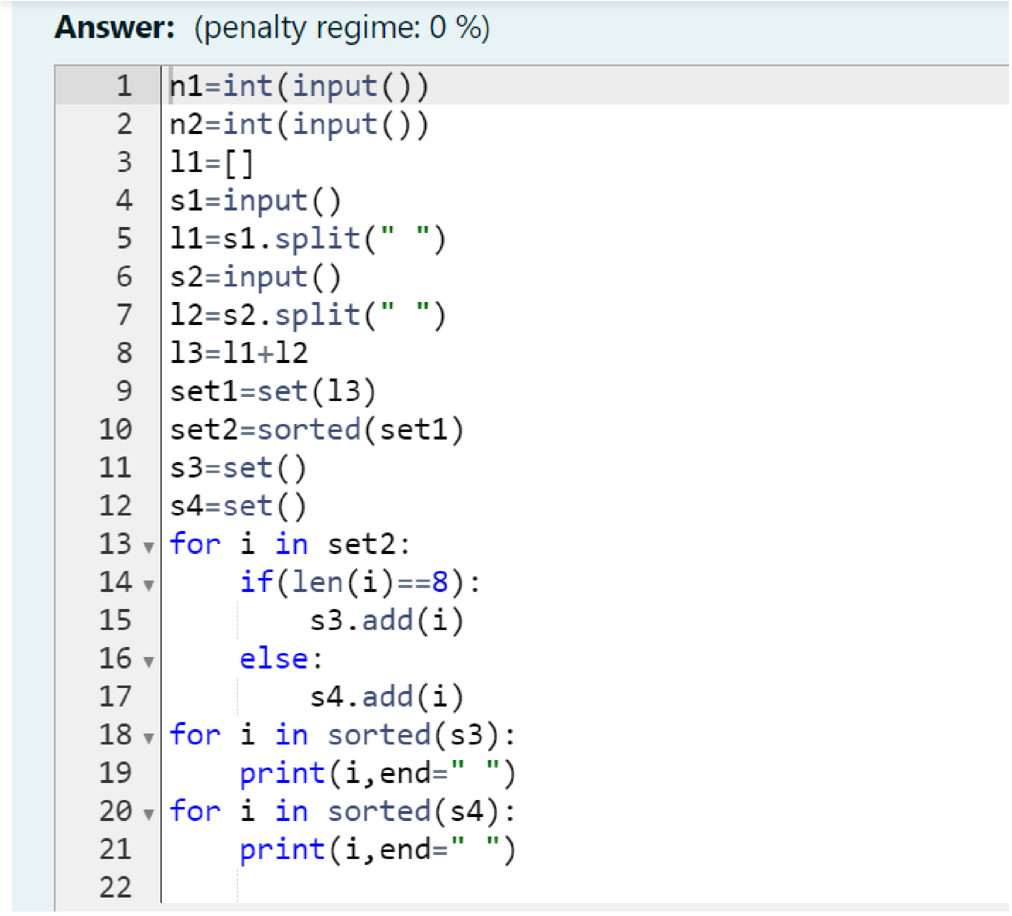
6

9840403212 9890909012 98123455 90123456 99123456

90909090 99999999 9840403212 12345678 12347890 99123456

Output

12345678 12347890 90123456 90909090 98123455 99123456 99999999 9840403212 9890909012

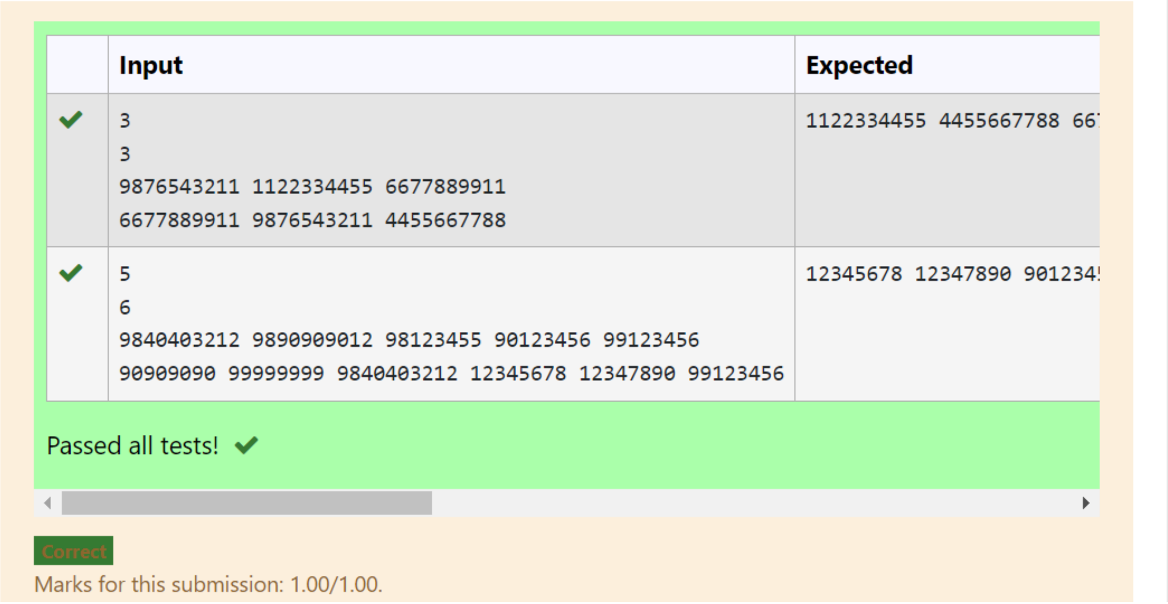


Answer: (penalty regime: 0 %)

```
n1=int(input())
n2=int(input())
l1=[]
s1=input()
l1=s1.split(" ")
s2=input()
l2=s2.split(" ")
l3=l1+l2
set1=set(l3)
set2=sorted(set1)
s3=set()
s4=set()
for i in set2:
    if(len(i)==8):
        s3.add(i)
    else:
        s4.add(i)
for i in sorted(s3):
    print(i,end=" ")
for i in sorted(s4):
    print(i,end=" ")

```

WEEK_09-Set



	Input	Expected

✓	3
	3
	9876543211 1122334455 6677889911
	6677889911 9876543211 4455667788
		1122334455 4455667788 66

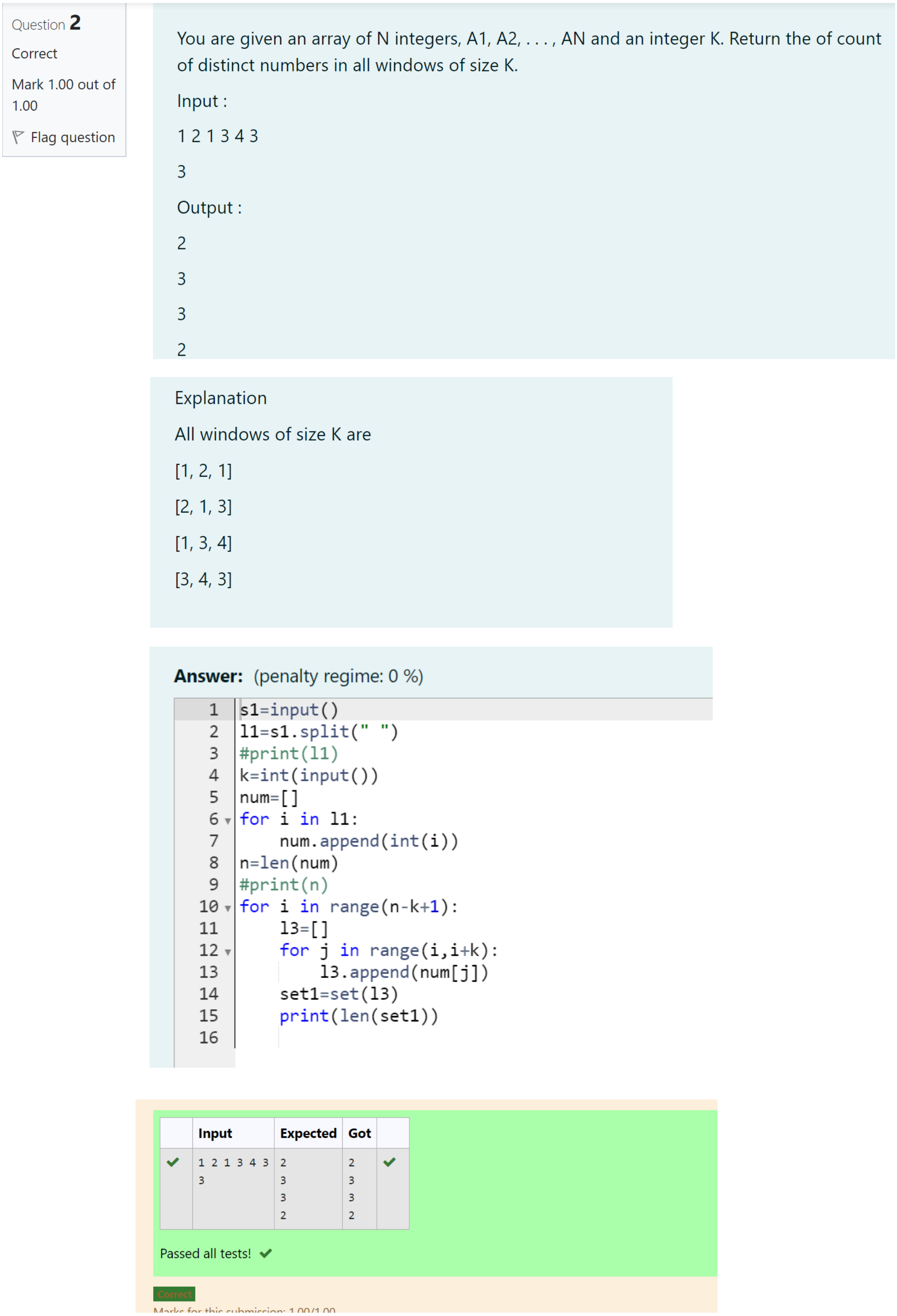
✓	5
	6
	9840403212 9890909012 98123455 90123456 99123456
	90909090 99999999 9840403212 12345678 12347890 99123456
		12345678 12347890 901234

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

WEEK_09-Set



Question **2**

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
s1=input()
l1=s1.split(" ")
#print(l1)
k=int(input())
num=[]
for i in l1:
    num.append(int(i))
n=len(num)
#print(n)
for i in range(n-k+1):
    l3=[]
    for j in range(i,i+k):
        l3.append(num[j])
    set1=set(l3)
    print(len(set1))

```

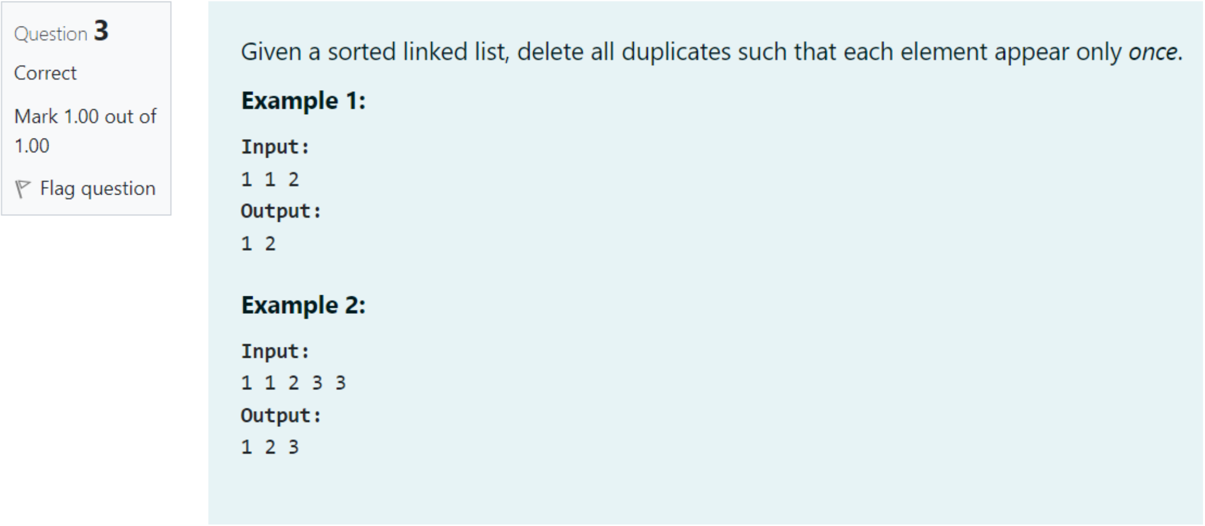
	Input	Expected	Got	
✓	1 2 1 3 4 3 3	2 3 3 2	2 3 3 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

WEEK_09-Set



Question **3**

Correct

Mark 1.00 out of 1.00

Flag question

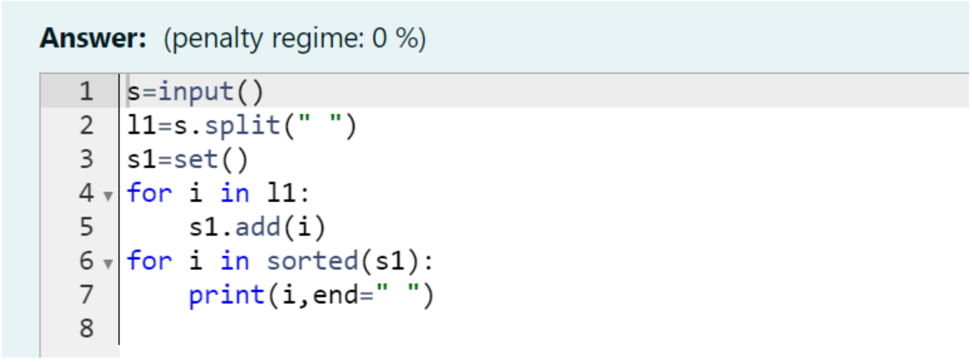
Given a sorted linked list, delete all duplicates such that each element appear only *once*.

Example 1:

```
Input:
1 1 2
Output:
1 2
```

Example 2:

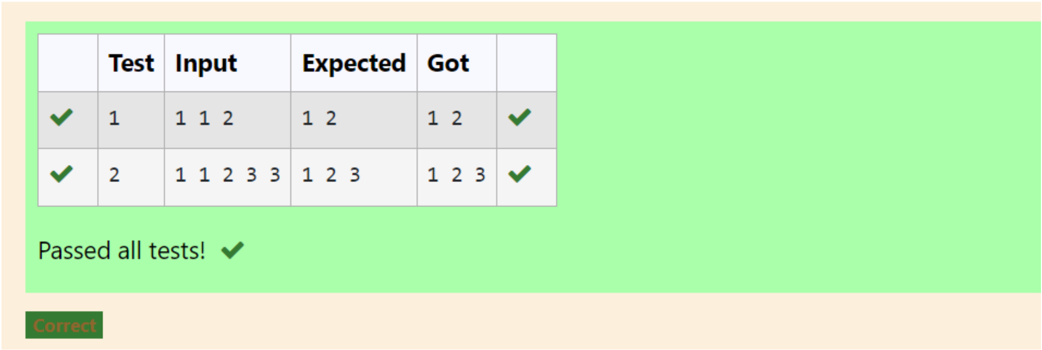
```
Input:
1 1 2 3 3
Output:
1 2 3
```



Answer: (penalty regime: 0 %)

```
s=input()
l1=s.split(" ")
s1=set()
for i in l1:
    s1.add(i)
for i in sorted(s1):
    print(i,end=" ")

```

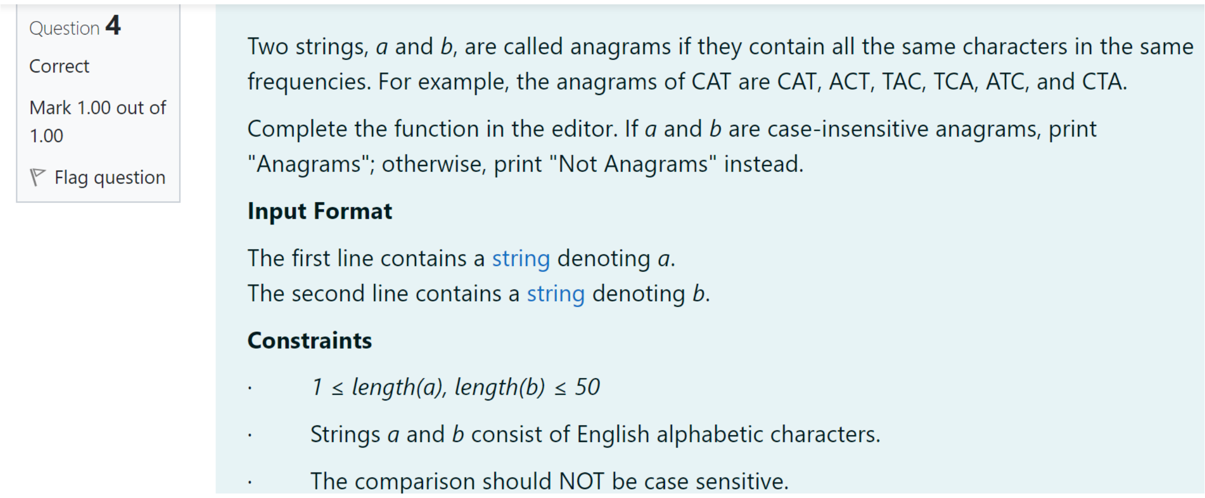


	Test	Input	Expected	Got	
✓	1	1 1 2	1 2	1 2	✓
✓	2	1 1 2 3 3	1 2 3	1 2 3	✓

Passed all tests! ✓

Correct

WEEK_09-Set



Question **4**

Correct

Mark 1.00 out of 1.00

Flag question

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.

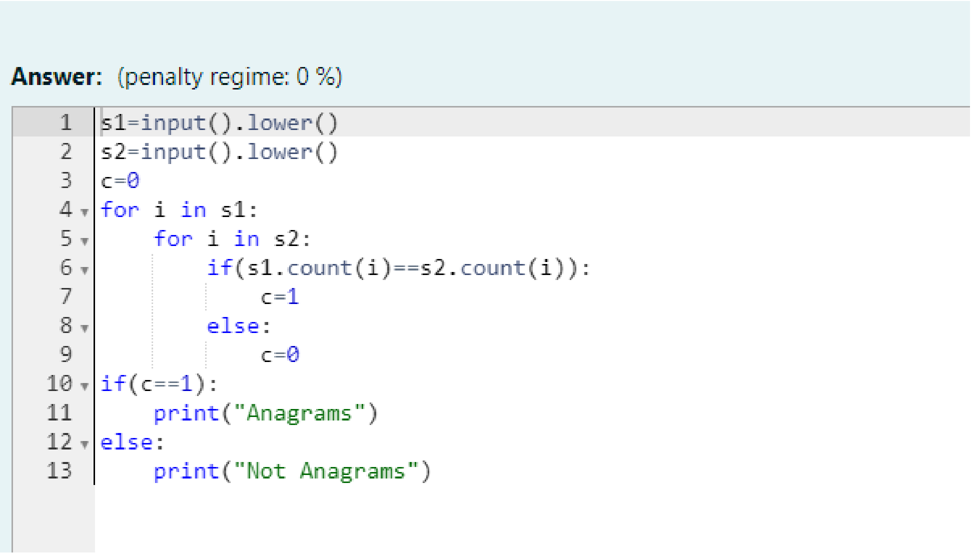
Complete the function in the editor. If *a* and *b* are case-insensitive anagrams, print "Anagrams"; otherwise, print "Not Anagrams" instead.

Input Format

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

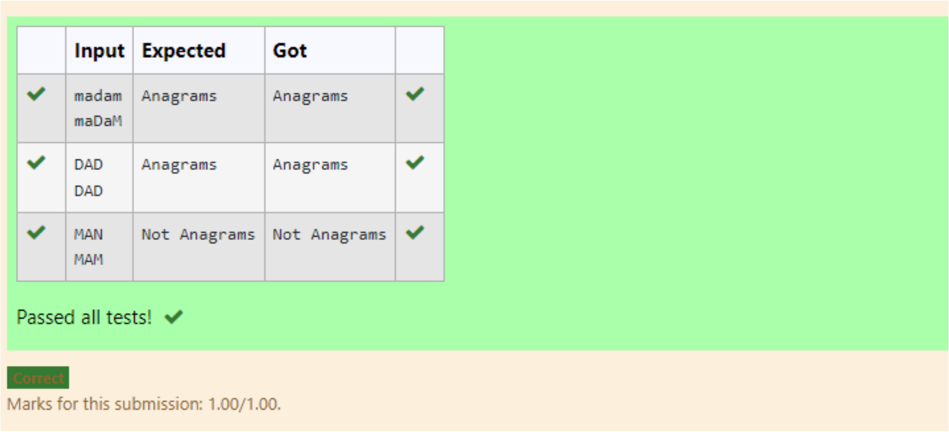
Constraints

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



Answer: (penalty regime: 0 %)

```
s1=input().lower()
s2=input().lower()
c=0
for i in s1:
    for i in s2:
        if(s1.count(i)==s2.count(i)):
            c=1
        else:
            c=0
if(c==1):
    print("Anagrams")
else:
    print("Not Anagrams")
```



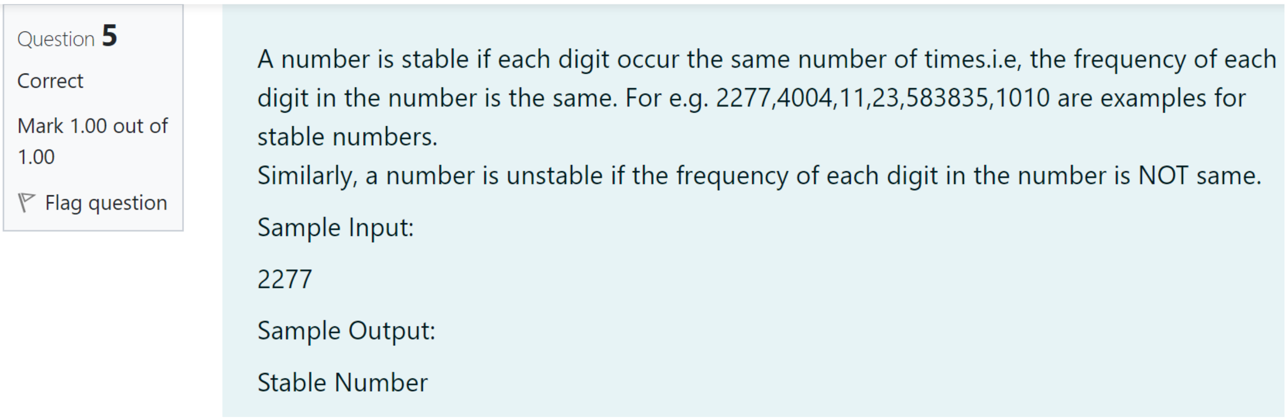
	Input	Expected	Got	
✓	madam maDaM	Anagrams	Anagrams	✓
✓	DAD DAD	Anagrams	Anagrams	✓
✓	MAN MAM	Not Anagrams	Not Anagrams	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

WEEK_09-Set



Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

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.

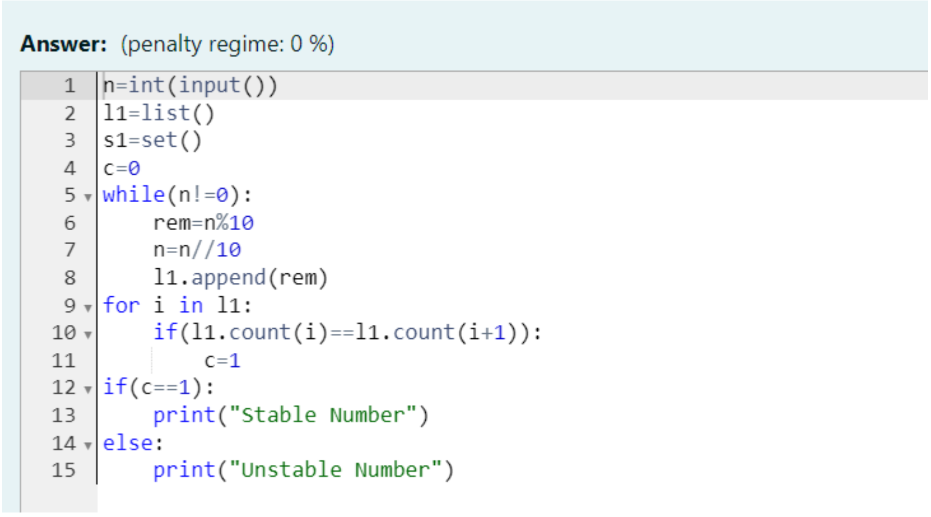
Similarly, a number is unstable if the frequency of each digit in the number is NOT same.

Sample Input:

2277

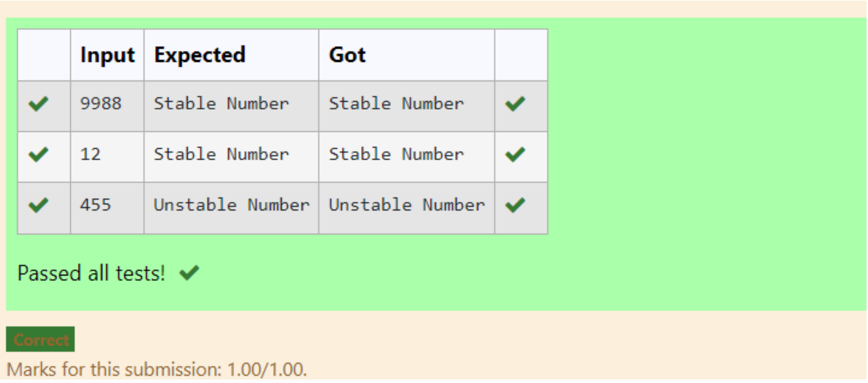
Sample Output:

Stable Number



Answer: (penalty regime: 0 %)

```
n=int(input())
l1=list()
s1=set()
c=0
while(n!=0):
    rem=n%10
    n=n//10
    l1.append(rem)
for i in l1:
    if(l1.count(i)==l1.count(i+1)):
        c=1
if(c==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.