

[Dashboard](#) / [My courses](#) / [CD19411-PPD-2022](#) / [WEEK 09-Set](#) / [WEEK-09 CODING](#)

<b>Started on</b>	Sunday, 5 May 2024, 8:50 PM
<b>State</b>	Finished
<b>Completed on</b>	Sunday, 5 May 2024, 9:54 PM
<b>Time taken</b>	1 hour 4 mins
<b>Marks</b>	5.00/5.00
<b>Grade</b>	<b>50.00</b> out of 50.00 ( <b>100%</b> )
<b>Name</b>	<a href="#">ABHIGNYA P 2022-CSD-A</a>



Question 1

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 n=int(input())
2 l1=list()
3 s1=set()
4 c=0
5 while(n!=0):
6     rem=n%10
7     n=n//10
8     l1.append(rem)
9 for i in l1:
10     if (l1.count(i)==l1.count(i+1)):
11         c=1
12 if(c==1):
13     print("Stable Number")
14 else:
15     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.

Question **2**

Correct

Mark 1.00 out of 1.00

Take a complete sentence as an input and remove duplicate word in it and print (sorted order), then count all the words which have a length greater than 3 and print.

Input

we are good are we good

Output

are good we

Count = 1

**For example:**

Input	Result
welcome to rec rec cse ece	cse ece rec to welcome Count = 1

**Answer:** (penalty regime: 0 %)

```
1 a=set(input().split())
2 print(' '.join(sorted(a)))
3 print("Count = {}".format(sum(1 for word in a if len(word)>3)))
```

	Input	Expected	Got	
✓	we are good are we good	are good we Count = 1	are good we Count = 1	✓
✓	welcome to rec rec cse ece	cse ece rec to welcome Count = 1	cse ece rec to welcome Count = 1	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Check if a set is a subset of another set.

Example:

Sample Input1:

mango apple  
mango orange  
mango

output1:

yes

set3 is subset of set1 and set2

input2:

mango orange

banana orange

grapes

output2:

no

**Answer:** (penalty regime: 0 %)

```
1 a=set(input())
2 b=set(input())
3 c=set(input())
4 if c.issubset(a):
5     print("yes\nset3 is subset of set1 and set2")
6 else:
7     print("No")
```

	Test	Input	Expected	Got	
✓	1	mango apple mango orange mango	yes set3 is subset of set1 and set2	yes set3 is subset of set1 and set2	✓
✓	2	mango orange banana orange grapes	No	No	✓

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**

- $1 \leq \text{length}(a), \text{length}(b) \leq 50$
- 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 | a=set(input().lower())
2 | b=set(input().lower())
3 | if a==b:
4 |     print("Anagrams")
5 | else:
6 |     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

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 def count(arr,k):
2     n=len(arr)
3     for i in range(n-k+1):
4         window=arr[i:i+k]
5         distinct=len(set(window))
6         print(distinct)
7 arr=list(map(int,input().split()))
8 k=int(input())
9 count(arr,k)
```

	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\_MCQ

