

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

<b>Started on</b>	Tuesday, 14 May 2024, 8:28 PM
<b>State</b>	Finished
<b>Completed on</b>	Tuesday, 14 May 2024, 8:50 PM
<b>Time taken</b>	21 mins 34 secs
<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="#">LIVIA MARY SEBASTIAN 2022-CSD-A</a>

## Question 1

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 class ListNode:
2     def __init__(self, val=0, next=None):
3         self.val = val
4         self.next = next
5
6 def deleteDuplicates(head: ListNode) -> ListNode:
7     current = head
8     while current and current.next:
9         if current.val == current.next.val:
10            current.next = current.next.next
11        else:
12            current = current.next
13    return head
14
15 def createLinkedListFromInput() -> ListNode:
16     nums = list(map(int, input().split()))
17     dummy = ListNode()
18     current = dummy
19     for num in nums:
20         current.next = ListNode(num)
21         current = current.next
22    return dummy.next

```

	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 **2**

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 %)

```
1 def print_common_elements_from_input():
2     list1 = list(map(int, input().strip().split()))
3     list2 = list(map(int, input().strip().split()))
4
5     set1 = set(list1)
6     set2 = set(list2)
7     common_elements = list(set1 & set2)
8
9     common_elements.sort()
10
11     if common_elements:
12         print(" ".join(map(str, common_elements)))
13     else:
14         print("No common elements")
15
16
17 print_common_elements_from_input()
18
```

	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 3

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")
16

```

	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 **4**

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")
8
```

	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 5

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

Output : 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 %)

```

1 n1=int(input())
2 n2=int(input())
3 l1=[]
4 s1=input()
5 l1=s1.split(" ")
6 s2=input()
7 l2=s2.split(" ")
8 l3=l1+l2
9 set1=set(l3)
10 set2=sorted(set1)
11 s3=set()
12 s4=set()
13 for i in set2:
14     if(len(i)==8):
15         s3.add(i)
16     else:
17         s4.add(i)
18 for i in sorted(s3):
19     print(i,end=" ")
20 for i in sorted(s4):
21     print(i,end=" ")
22

```

	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.

[◀ Week-09\\_MCQ](#)

Jump to...

[WEEK-09-Extra ▶](#)