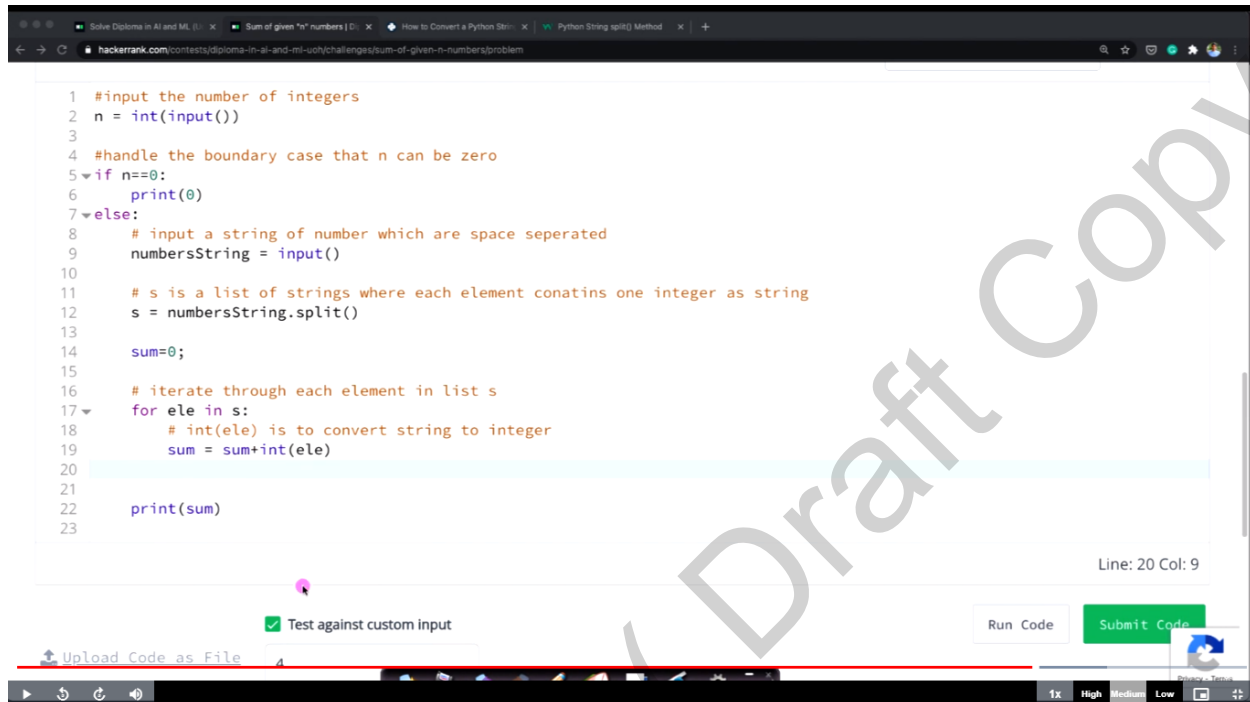


7.2 Problem 1: Sum of 'n' input numbers

The below example is explained in the video



```
1 #input the number of integers
2 n = int(input())
3
4 #handle the boundary case that n can be zero
5 if n==0:
6     print(0)
7 else:
8     # input a string of number which are space seperated
9     numbersString = input()
10
11     # s is a list of strings where each element conatins one integer as string
12     s = numbersString.split()
13
14     sum=0;
15
16     # iterate through each element in list s
17     for ele in s:
18         # int(ele) is to convert string to integer
19         sum = sum+int(ele)
20
21
22     print(sum)
23
```

Line: 20 Col: 9

☒ Test against custom input

[Upload Code as File](#) [Run Code](#) [Submit Code](#)

- In this program, the first input we give is the total number of elements in the list.
- In case, if the entered number of elements is 0, then we have to print 0.
- Otherwise, we accept the sequence of values in the form of a string separated by spaces. We initialize a variable 'sum' equal to 0.
- We split the string into a list of strings and while traversing through the list, we convert each number in the string format into integer format and then add it to the 'sum' variable. Finally we are printing the 'sum' value.

7.3 Problem 2: Second largest element in a list

Current Buffer (saved locally, editable) Python 3

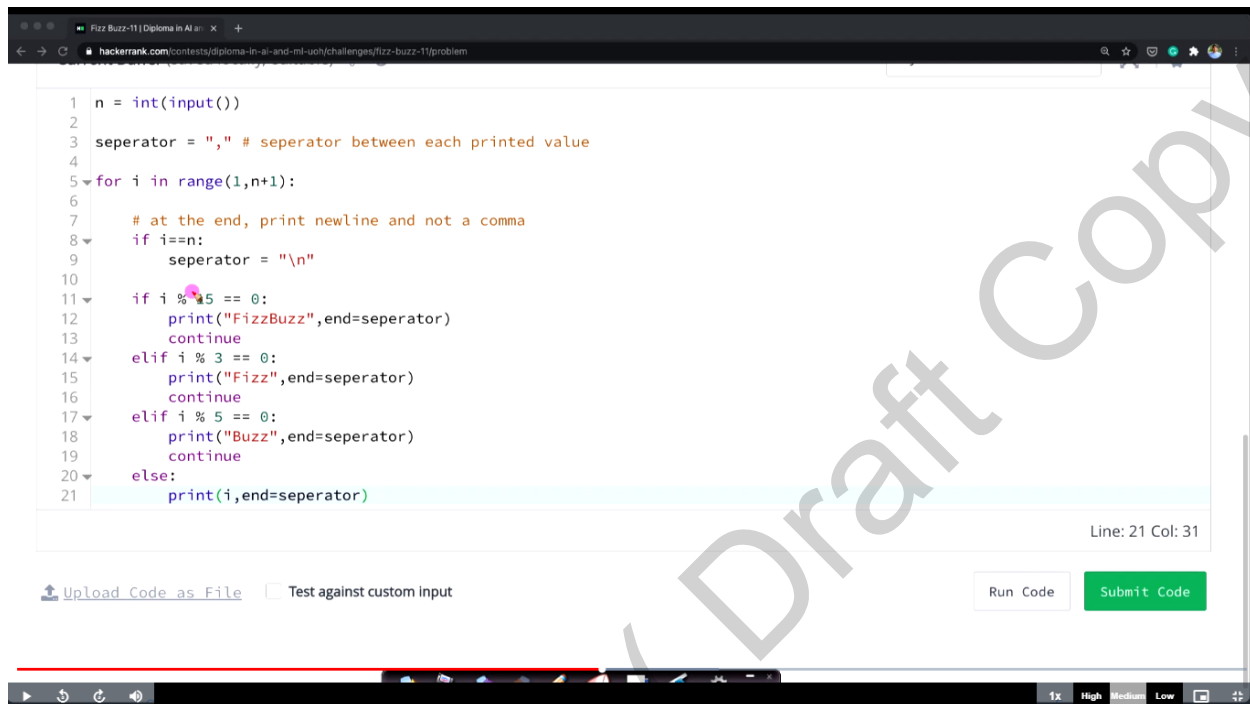
```
1 # Good idea to write your test cases to build the logic
2 # sample input: 1,2,3,4,6,6 =>1
3 # sample input: 3,2,3 => 2
4 # sample input: 1,1,1 => Error
5 # sample input: 1 => Error
6
7 numberString = input()
8
9 # split the numberString and convert it to a list of numbers
10 l = [int(i) for i in numberString.split(',')] # list comprehension
11 n=len(l)
12
13 # boundary case
14 if n<2:
15     print("Error")
16     exit()
17
18 # initialize firstMax and secondMax
19 firstMax=max(l[0],l[1])
20 secondMax=min(l[0],l[1])
21
22
23
24
25 # iterate
26 for i in range(2,n):
27     if l[i]>firstMax: # test case: 3,2,4
28         secondMax=firstMax
29         firstMax=l[i]
30     elif l[i]>secondMax and firstMax != l[i]: # test case: 3,2,3
31         secondMax=l[i]
32
33 # Boundary case: input: 1,1,1 => there is no second largest element
34 if firstMax==secondMax:
35     print("Error")
36 else:
37     print(secondMax)
```

Line: 12 Col: 1

[Upload Code as File](#) [Test against custom input](#) [Run Code](#) [Submit Code](#)

7.4 Problem 3: Fizz Buzz

The below example is explained in the video.



```
1 n = int(input())
2
3 separator = "," # separator between each printed value
4
5 for i in range(1,n+1):
6
7     # at the end, print newline and not a comma
8     if i==n:
9         separator = "\n"
10
11     if i % 5 == 0:
12         print("FizzBuzz",end=separator)
13         continue
14     elif i % 3 == 0:
15         print("Fizz",end=separator)
16         continue
17     elif i % 5 == 0:
18         print("Buzz",end=separator)
19         continue
20     else:
21         print(i,end=separator)
```

Line: 21 Col: 31

[Upload Code as File](#) ☐ Test against custom input

[Run Code](#) [Submit Code](#)

In this example, we are first taking the input number 'n' from the user at the runtime and are initializing the 'separator' variable to ','.

The condition here is for all the numbers in the range 1 to n(inclusive), if a number is divisible by 3, then we have to print the word '**Fizz**'. If it is divisible by 5, then we have to print the word '**Buzz**'. If it is divisible by both 3 and 5, then we have to print the string '**Fizz Buzz**'. If the number is neither divisible by 3 nor divisible by 5, then we have to print the number as it is.

Each value we print should be separated by a comma(,) and hence we have initialized the separator as ','. While looping through the sequence, we have to change the 'separator' value to '\n', once if it becomes equal to '\n'.