Week 8- Day 1 : Coding Challenge

(Maximum marks -15)

Q-1) Recursive implementation of atoi() function:(5 marks)

Atoi() function converts a string into an integer.

eg:
st = "1234" is a string.
if we perform,
st + 1
this results in error since "st" is a string and 1 is an integer, and,
st + "1"
this will append 1 into 1234. Giving us 12341.

write a function that converts the above variable 'st' into an integer (so that we can perform mathematical operations on it).
Let's call our function "myAtoiRecursive()", it should,

myAtoiRecursive(st) + 1

should give us 1235 (that is 1234+1).

As shown in image below:

```
>>> st = "1234"
>>> st+1
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
>>> st+"1"
'12341'
>>> var = myAtoiRecursive(st)
>>> var + 1
1235
```

Sample input:
"1234"

Sample output:
1234

atoi() function stands for ASCII to integer conversion. It is a C function, but not present in python. Try to write a recursive code that implements atoi() in python.)

Q-2) Write a function that prints digits of a number from left to right , using recursion:(5 marks)

Sample Input:
1234567
Sample output:
1
2
3
4
5
6
7

Q-3) Reverse a string using recursion:(5 marks)

If we have a string, write a function that prints reverse of that string, using recursion.

Sample Input:
ABCD

Sample Output:
DCBA

Q-4) [Bonus Question] Recursive implementation of binary search: (5 extra marks)

We have seen an iterative approach for binary search algorithm , write a recursive approach for that.

HINT: when we divide the array into two parts, we need to perform a search on only one half. Apply binary search only on that half.

Marks distribution:
Question 1,2 and 3 carry 5 marks each.

Question 4 is a bonus question, that means if you leave that question you dont lose a mark, but if you solve it, you can extra 5 marks.

Remark: maximum marks you can get is 15, bonus question helps only of you are not able to solve another question.