RECURSION

What is Recursion?
The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called as recursive function.

```
def Hello():
    print("Hi")
    Hello()

if __name__ == "__main__":
    Hello()
```

If I want to print ‘hi’ 5 times, we can use the following program:

```
cnt = 0
def Hello():
    global cnt
    if cnt == 3: # this is called Base condition.
        return
    print("Hi")
    cnt += 1
    Hello ()
if __name__ == "__main__":
    Hello()
```

OUTPUT:

hi
hi
hi

time Complexity for this program would be O (n).

What is base condition in recursion?
In the recursive program, the solution to the base case is provided and the solution of the bigger problem is expressed in terms of smaller problems.

Q) Given a number print its factorial. (n!)

n = 5; 5! = 5 * 4 * 3 * 2 * 1

OR

5! = 5 * 4!

4 * 3!

3 * 2!

2 * 1!

1 * 0!

5! = 5 * Fact (4)

4 * Fact (3)

3 * Fact (2)

2 * Fact (1)

return 1

Recursive Formula: $F(n) = n * F(n - 1)$

```python
def Factorial(n):
    """
    Factorial n will find the factorial of a no. n.
    """
    if n == 1:
        return 1
    fact = n * Factorial(n - 1)
    return fact

if __name__ = "__main__":
    print(Factorial (5))
```

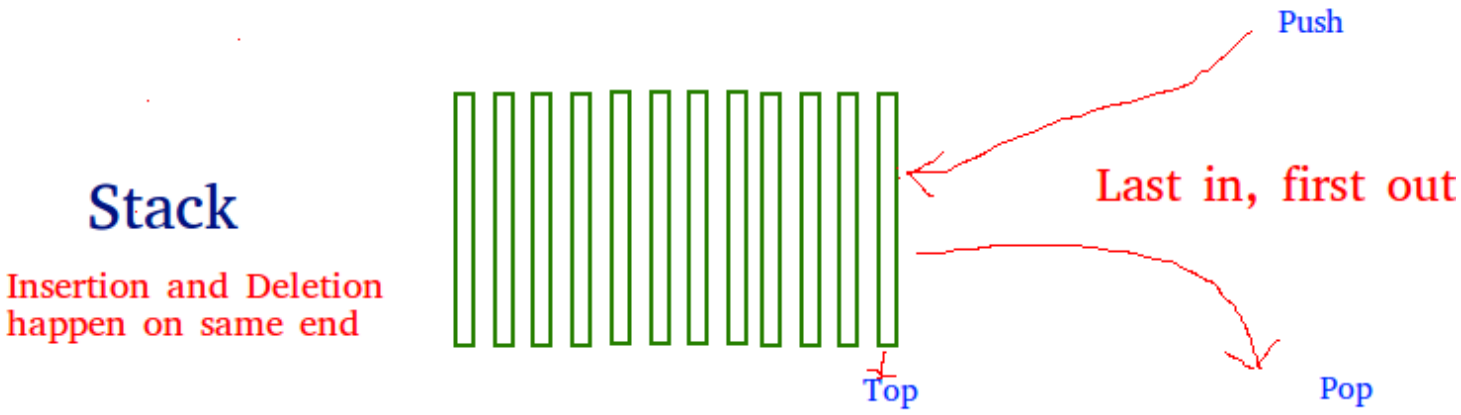
NOTE:

- Code will always go line-by-line.
- Each function variables are stored or preserved for each call.
- In recursion, ‘function calls’ are stored in a type of data structure called stack.

Stack

Stack is a type of Data Structure in which values are added at the top.

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out).



In our program, the first call Fact (5), gets inserted in the **call stack**. Since n == 1 is false, the program will go to line 3. The formula will get initiated. This is paused because, in that, Factorial (n-1) is another function, which gets added to the stack and it will resume after Factorial (n – 1) is resolved.

For n = 4, the first line will not be executed. So, the program will go to 3rd line and the formula is initiated. This is paused because, in that, Factorial (n-1) is another function, which gets added to the stack and it will resume after Factorial (n – 1) is resolved.

This will repeat for n = 3, 2 & 1 and all the functions would be stacked in the **call stack**. Once the Fact (1) is resolved, its value is popped from the stack and will return the value to Fact (2).

The value of Fact (2) returned 2, which will be returned to Fact (3) and the answer would be 2 * 3 = 6.

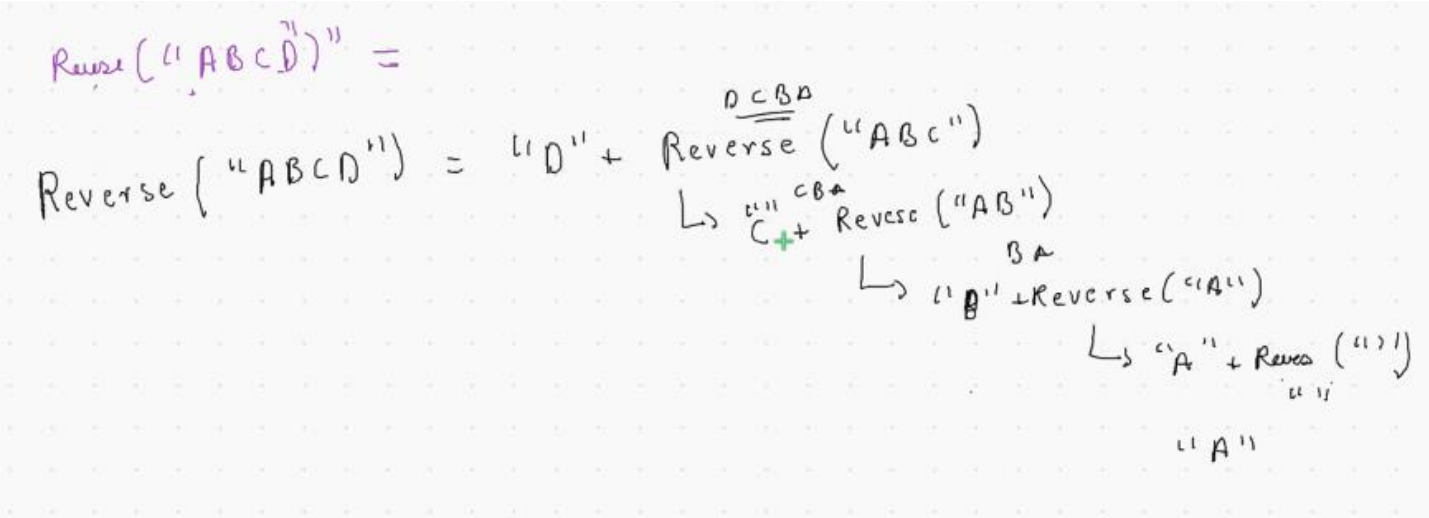
The Fact (3) recursion is done with the value 6 which is popped out from the stack and will be returned to Fact (4). So, Fact (4) = 4 * 6 = 24.

24 is popped out and the Fact (5) = 5 * 24 = **120**.

Time Complexity for this program would be O (n) and space complexity for this would-be O (n) because a stack is created.

Q) Given a string reverse it using recursion.

INPUT: “ABCD” OUTPUT: “DCBA”




```python
def Reverse(str):
    if str == "":
        return ""
    rev = str[-1] + Reverse[:-1]
    return rev

if __name__ == "__main__":
    print(Reverse("ABCD"))
```

OUTPUT: “DCBA”

MCQ's

Q)

```python
def R(l):
    if len(l) == 0:
        return 0
    count = 1 + R(l[:-1])
    return count

if __name__ == "__main__":
    print(R([1,2,3,4,5]))
```

Step 7	cnt = 0	R([ ])	cnt = 1 + 0	Step 6
Step 8	cnt = 1	R([1])	cnt = 1 + R([ ])	Step 5
Step 9	cnt = 2	R([1, 2])	cnt = 1 + R([1])	Step 4
Step 10	cnt = 3	R([1, 2, 3])	cnt = 1 + R([1, 2])	Step 3
Step 11	cnt = 4	R([1, 2, 3, 4])	cnt = 1 + R([1, 2, 3])	Step 2
Step 12	cnt = 5	R([1, 2, 3, 4, 5])	cnt = 1 + R([1, 2, 3, 4])	Step 1

Q)

```python
def R(l):
    if len(l) == 0:
        return 0
    count = 0
    if l[-1] % 2 == 0:
        count =  1 + R(l[:-1])
    else:
        count =  R(l[:-1])
    return count
```

```
if __name__ == "__main__":
    print(R([12, 22, 32, 43, 58,118, 21]))
```

https://www.geeksforgeeks.org/stack-in-python/

https://www.w3resource.com/c-programming-exercises/recursion/index.php

https://www.youtube.com/watch?v=KEEKn7Me-ms

https://www.youtube.com/watch?v=wMNrSM5RFMc