Lecture 16 : Recursion - 1

(64) Factorial

```
def fact(n) :
        if n==0 :
            return 1
        return n*fact(n-1)
```

$fact(4) = \underline{\underline{24}}$

$$\begin{tikzcd} fact(4) \arrow[r] & 4*fact(3) \arrow[r] & 4*3*fact(2) \arrow[d] \\ & 4*3*2*1*\underbrace{fact(0)}_{1} & 4*3*2*fact(1) \arrow[l] \end{tikzcd}$$

(65) Fibonacci

ix :	0	1	2	3	4	5
v :	0	1	1	2	3	5 ...

```
def fib(n):
    if n==0:
        return 0
    if n==1
        return 1
    return fib(n-1) + fib(n-2)
```

fib(3) ⟶ fib(2) + fib(1) = $\underline{\underline{2}}$

1

fib(1) + fib(0)

1 0

(66) Check if a string is PALINDROME or not (using Recursion)

Brute Force :

```
def checkPalindrome(s):
    if len(s) <= 1 :
        return True

    if s[0] == s[-1] and solve(s[1:-1]):

        return True

    return False
```

Optimized

```
def isPalindrome(w, s, e):
    if s >= e:
        return True
    if w[s] == w[e]:
        return isPalindrome(w, s+1, e-1)
    return False
def solve(word):
    return isPalindrome(word, 0, len(word)-1)
```

(67) Print reverse string using Recursion

```
import sys
sys.setrecursionlimit(10**6)

def printrevstring( ip-string, s, e):

    if s>=e:
        return ip-string[s]

    print ( ip-string[e], end="")

    return printrevstring(ip-string, s, e-1)


def main():

    ip = str(input())

    print ( printrevstring(ip, 0, len(ip-1)))
```