```
"""
Recursion:
The Process of function calling itself is called recursion.

Recursive Function:
The Function which is calling itself is called recursive function.


Types of Recursion:
1. Based on the Function Call:
    1. Direct Recursion
       The Recursive call lies in the same function.
       Ex:
       def foo():
           foo()

    2. Indirect Recursion
       The Recursive call lies in a another function which is called in this function.
       Ex:
       def foo():
           bar()

       def bar()
          foo()


2. Based on the position of the recursive Call:
    1. Head Recursion
    2. Tail Recursion.


Recursive Function contains:
1. Base Condition:
   The Condition at which to stop the recursion.
   The Recursive Function without the base condition
   will lead to Maximum Recursion Depth Exceeded.
2. Recursive Call:
   Recursive call with updated parameters.
"""


"""
Find the factorial for a number
Ex:
Input: 5
Output: 120 (5! = 5 * 4 * 3 * 2 * 1)
Input: 6
Output: 720 (6! = 6 * 5 * 4 * 3 * 2 * 1)
"""

# Without Recursion
"""
def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact
"""
```

```
# With Recursion
def factorial(num):
    if num == 0:
        return 1
    else:
        return num * factorial(num - 1)


num = int(input("Enter Number to find the factorial: "))
result = factorial(num)
print(result)


# Head Recursion
def head_recursion_factorial(num):
    if num == 0:
        return 1
    else:
        return factorial(num - 1) * num


# Tail Recursion
def tail_recursion_factorial(num):
    if num == 0:
        return 1
    else:
        return num * factorial(num - 1)
```