

```
def factorial_recursive(n, depth=0):
    if n < 0:
        print("Factorial not possible for negative numbers")
        return None
    if n == 0 or n == 1:
        print(" " * depth + f"factorial({n}) = 1 (base case)")
        return 1
    else:
        print(" " * depth + f"factorial({n}) = {n} * factorial({n-1})")
        result = n * factorial_recursive(n-1, depth+1)
        return result
num = int(input("Enter a number: "))

print("\n--- Recursive Factorial Trace ---")
answer = factorial_recursive(num)
print("\nFinal Answer:", answer)
```

Enter a number: 8

```
--- Recursive Factorial Trace ---
factorial(8) = 8 * factorial(7)
factorial(7) = 7 * factorial(6)
factorial(6) = 6 * factorial(5)
factorial(5) = 5 * factorial(4)
factorial(4) = 4 * factorial(3)
factorial(3) = 3 * factorial(2)
factorial(2) = 2 * factorial(1)
factorial(1) = 1 (base case)
```

Final Answer: 40320

```
def factorial_tail(n, acc=1):
    if n < 0:
        return None
    if n == 0 or n == 1:
        return acc
    return factorial_tail(n-1, n*acc)
num = int(input("Enter a number: "))
print("\n--- Tail Recursive Factorial ---")
print("Answer:", factorial_tail(num))
```

Enter a number: 8

```
--- Tail Recursive Factorial ---
Answer: 40320
```

```
def factorial_iterative(n):
    if n < 0:
        return None
    fact = 1
    for i in range(2, n+1):
        fact *= i
    return fact
num = int(input("Enter a number: "))
print("\n--- Iterative Factorial ---")
print("Answer:", factorial_iterative(num))
```

Enter a number: 8

```
--- Iterative Factorial ---
Answer: 40320
```

```
num = int(input("Enter a number: "))
print("\nComparison Table:")
print("Recursive:", factorial_recursive(num))
print("Tail Recursion:", factorial_tail(num))
print("Iterative:", factorial_iterative(num))
```

Enter a number: 8

```
Comparison Table:
factorial(8) = 8 * factorial(7)
factorial(7) = 7 * factorial(6)
factorial(6) = 6 * factorial(5)
factorial(5) = 5 * factorial(4)
factorial(4) = 4 * factorial(3)
factorial(3) = 3 * factorial(2)
factorial(2) = 2 * factorial(1)
factorial(1) = 1 (base case)
Recursive: 40320
Tail Recursion: 40320
```

Iterative: 40320

```
# 2)Calculate nCr Using Recursive Factorial
def factorial_recursive(n):
    if n < 0:
        print("Factorial not possible for negative numbers")
        return None
    if n == 0 or n == 1: # base case
        print("factorial(", n, ") = 1")
        return 1

    # recursive case
    print("factorial(", n, ") = ", n, "* factorial(", n-1, ")")
    return n * factorial_recursive(n-1)

def nCr(n, r):
    # validate input
    if n < 0 or r < 0 or n < r:
        print("Invalid values! make sure n ≥ r ≥ 0")
        return None

    # edge cases
    if r == 0 or n == r:
        print("Since r=0 or n=r, nCr = 1 directly")
        return 1

    # compute using formula nCr = n! / (r! * (n-r)!)
    print("\n--- Calculating n! ---")
    fact_n = factorial_recursive(n)
    print("\n--- Calculating r! ---")
    fact_r = factorial_recursive(r)
    print("\n--- Calculating (n-r)! ---")
    fact_n_r = factorial_recursive(n-r)

    result = fact_n // (fact_r * fact_n_r)
    return result

n = int(input("Enter n: "))
r = int(input("Enter r: "))
print("\n--- nCr Calculation ---")
print("C(", n, ", ", r, ") =", nCr(n, r))
```



Enter n: 8
Enter r: 2

--- nCr Calculation ---

--- Calculating n! ---

```
factorial( 8 ) = 8 * factorial( 7 )
factorial( 7 ) = 7 * factorial( 6 )
factorial( 6 ) = 6 * factorial( 5 )
factorial( 5 ) = 5 * factorial( 4 )
factorial( 4 ) = 4 * factorial( 3 )
factorial( 3 ) = 3 * factorial( 2 )
factorial( 2 ) = 2 * factorial( 1 )
factorial( 1 ) = 1
```

--- Calculating r! ---

```
factorial( 2 ) = 2 * factorial( 1 )
factorial( 1 ) = 1
```

--- Calculating (n-r)! ---

```
factorial( 6 ) = 6 * factorial( 5 )
factorial( 5 ) = 5 * factorial( 4 )
factorial( 4 ) = 4 * factorial( 3 )
factorial( 3 ) = 3 * factorial( 2 )
factorial( 2 ) = 2 * factorial( 1 )
factorial( 1 ) = 1
C( 8 , 2 ) = 28
```

