|  |  |
| --- | --- |
| EX:NO:6a | Implementing programs using Functions. (Factorial) |
| Date: |

**Aim:**

To Implement programs using Functions. (Factorial)

**Algorithm:**

**Step1:** Start the program

**Step2:** This code defines a function factorial that calculates the factorial of a given

number n.

**Step3:** The base case checks if n is 0 or 1, in which case the factorial is 1.

Otherwise, the function recursively calls itself with n-1 until it reaches the

base case.

**Step4**: The example usage shows how to use the function with a specific

number (4 in this case) and prints the result.

**Program: 1**

# Factorial of a number using recursion

def recursion\_factorial(n):

if n == 1:

return n

else:

return n\*recursion\_factorial(n-1)

num = 4

# check if the number is negative

if num < 0:

print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

print("The factorial of", num, "is", recursion\_factorial(num))

**Program:2**

def factorial(n):

if n == 0 or n == 1:

return 1

else:

return n \* factorial(n-1)

# Example usage:

number = 4

result = factorial(number)

print(f"The factorial of {number} is: {result}")

**Explanation:**

Factorial is a non-negative integer. It is the product of all positive integers less than or equal to that number you ask for factorial. It is denoted by an exclamation sign (!).

**Example:**

n! = n\* (n-1) \* (n-2) \*........1

4! = 4x3x2x1 = 24

The factorial value of 4 is 24.

**Result:**