**ASSIGNMENT-1**

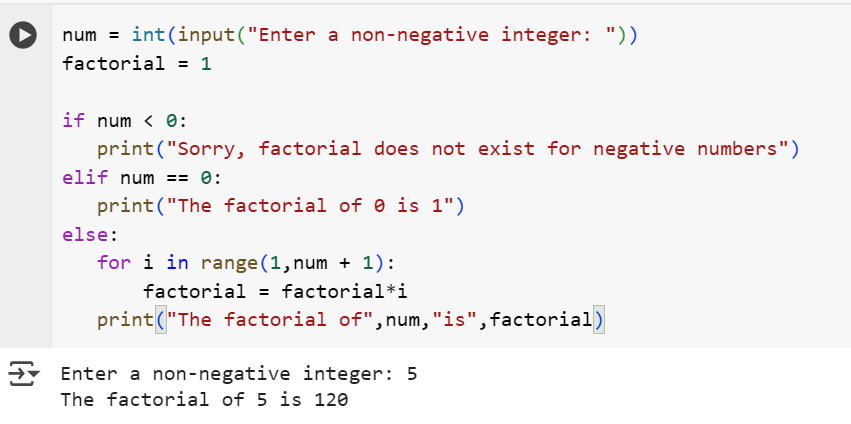
**NAME : G.SAI RAJ**

**HT NO: 2403A52149**

**BATCH:** 24BTCAIAIB06

**TASK-1:** Write a python program that calculates the factorial of a number without defining my functions

**CODE AND OUTPUT:**

****

**EXPLAINATION:**

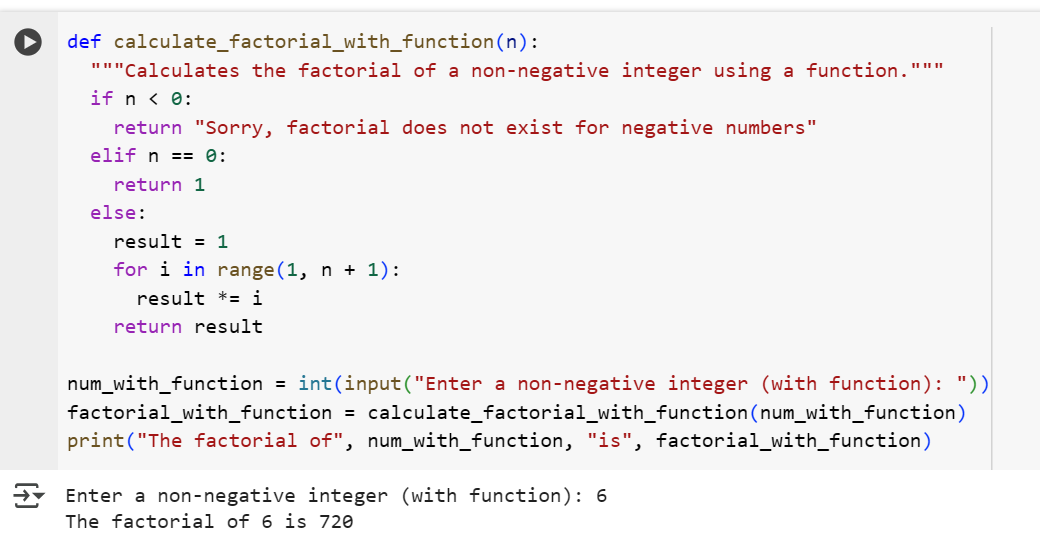
1. **num = int(input("Enter a non-negative integer: "))**: This line prompts the user to enter a non-negative integer using the input() function. The entered value is then converted to an integer using int() and stored in the variable num.
2. **factorial = 1**: This line initializes a variable called factorial to 1. This variable will store the calculated factorial value.
3. **if num < 0:**: This is the start of an if-elif-else block that checks the value of num. This condition checks if the entered number is negative.
4. **print("Sorry, factorial does not exist for negative numbers")**: If num is negative, this message is printed.
5. **elif num == 0:**: This condition checks if the entered number is 0.
6. **print("The factorial of 0 is 1")**: If num is 0, this message is printed because the factorial of 0 is defined as 1.
7. **else:**: This block is executed if num is neither negative nor zero (i.e., it's a positive integer).
8. **for i in range(1,num + 1):**: This line starts a for loop. The range(1, num + 1) function generates a sequence of numbers from 1 up to and including num. The loop iterates through each number in this sequence, assigning it to the variable i.
9. **factorial = factorial\*i**: Inside the loop, this line updates the factorial variable by multiplying its current value by the current value of i. This repeatedly multiplies factorial by 1, 2, 3, and so on, up to num.
10. **print("The factorial of",num,"is",factorial)**: After the loop finishes, this line prints the final calculated factorial value along with the original number entered by the user.

In summary, the code handles negative input, the special case of 0, and iteratively calculates the factorial for positive integers using a loop.

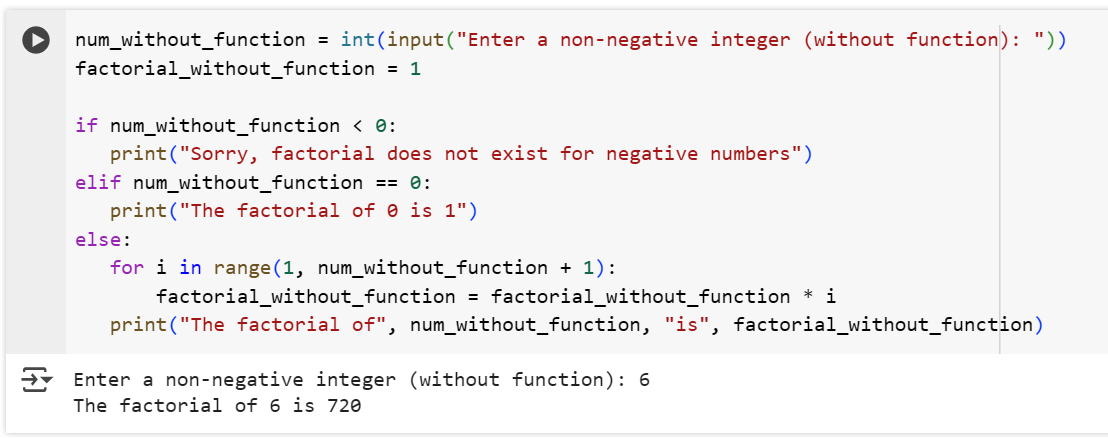
**TASK-2:** Write a python program that calculates the factorial of a number with function and without function.

**CODE AND OUTPUT:**

* WITH FUNCTION:



* WITHOUT FUNCTION:



**EXPLAINATION:**

* WITH FUNCTION EXPLANATION

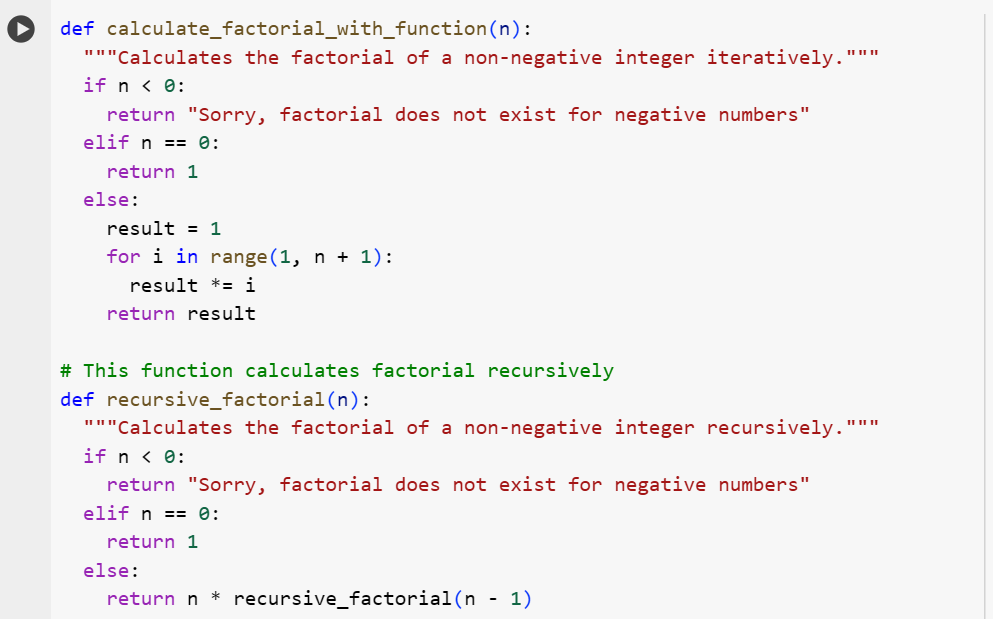
1. **num\_without\_function = int(input("Enter a non-negative integer (without function): "))**: This line prompts the user to enter a non-negative integer using the input() function. The text inside the parentheses is displayed to the user. The input is then converted to an integer using int() and stored in the variable num\_without\_function.
2. **factorial\_without\_function = 1**: This initializes a variable factorial\_without\_function to 1. This variable will hold the calculated factorial.
3. **if num\_without\_function < 0:**: This is the start of an if-elif-else block. This specific condition checks if the entered number is negative.
4. **print("Sorry, factorial does not exist for negative numbers")**: If the number is negative, this message is printed, as factorials are not defined for negative numbers.
5. **elif num\_without\_function == 0:**: This condition checks if the entered number is 0.
6. **print("The factorial of 0 is 1")**: If the number is 0, this message is printed because the factorial of 0 is 1.
7. **else:**: This block is executed if the number is a positive integer.
8. **for i in range(1, num\_without\_function + 1):**: This starts a for loop that iterates from 1 up to and including the value of num\_without\_function. In each iteration, the current number is assigned to the variable i.
9. **factorial\_without\_function = factorial\_without\_function \* i**: Inside the loop, the factorial\_without\_function is updated by multiplying its current value by the current value of i. This process continues for each number from 1 to num\_without\_function, effectively calculating the product of these numbers.
10. **print("The factorial of", num\_without\_function, "is", factorial\_without\_function)**: After the loop finishes, this line prints the final calculated factorial\_without\_function along with the original input number.

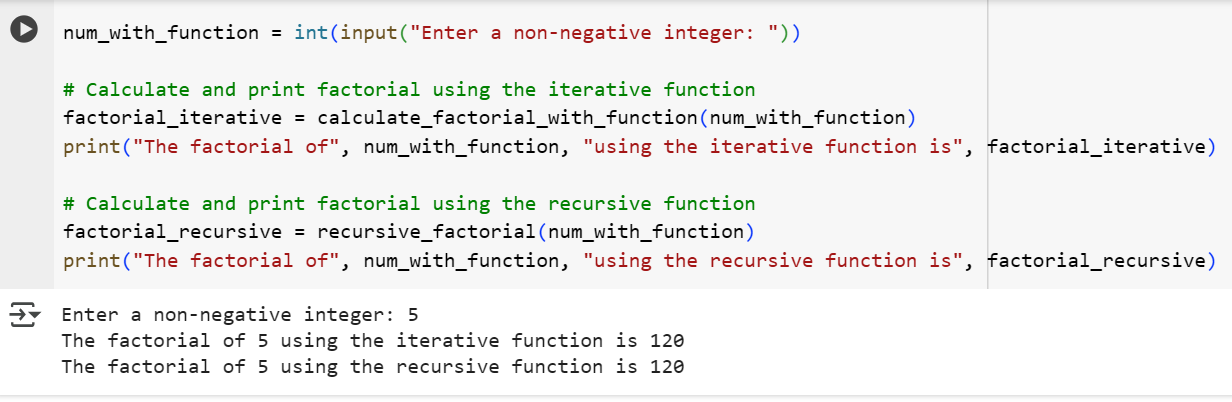
* WITHOUT FUNCTION EXPLANATION

1. **def calculate\_factorial\_with\_function(n):**: This line defines a function named calculate\_factorial\_with\_function that takes one argument, n. The code indented below this line is part of the function's body.
2. **"""Calculates the factorial of a non-negative integer using a function."""**: This is a docstring, which explains what the function does. It's a good practice for documenting your code.
3. **if n < 0:**: Inside the function, this checks if the input number n is negative.
4. **return "Sorry, factorial does not exist for negative numbers"**: If n is negative, the function immediately returns this string message.
5. **elif n == 0:**: This checks if n is 0.
6. **return 1**: If n is 0, the function returns 1, as the factorial of 0 is 1.
7. **else:**: This block is executed if n is a positive integer.
8. **result = 1**: A variable result is initialized to 1. This will store the calculated factorial within the function.
9. **for i in range(1, n + 1):**: A for loop iterates from 1 up to and including n.
10. **result \*= i**: In each iteration, result is multiplied by the current value of i. This is a shorthand for result = result \* i.
11. **return result**: After the loop completes, the function returns the final calculated result.
12. **num\_with\_function = int(input("Enter a non-negative integer (with function): "))**: This line is outside the function definition. It prompts the user for input, converts it to an integer, and stores it in num\_with\_function.
13. **factorial\_with\_function = calculate\_factorial\_with\_function(num\_with\_function)**: This line *calls* the calculate\_factorial\_with\_function function, passing the user's input num\_with\_function as the argument n. The value returned by the function is then stored in the variable factorial\_with\_function.
14. **print("The factorial of", num\_with\_function, "is", factorial\_with\_function)**: Finally, this line prints the original number and the calculated factorial returned by the function.

**TASK-3:** Write a python program that calculates the factorial using both iterative and recursive functions.

**CODE AND OUTPUT:**





**EXPLANATION:**

**1. Iterative Factorial Function (calculate\_factorial\_with\_function)**

* **def calculate\_factorial\_with\_function(n):**: This defines a function named calculate\_factorial\_with\_function that takes an integer n as input.
* **"""Calculates the factorial of a non-negative integer iteratively."""**: This is a docstring explaining the function's purpose.
* **if n < 0:**: Checks if the input n is negative. If it is, it returns a message indicating that factorial doesn't exist for negative numbers.
* **elif n == 0:**: Checks if n is 0. If it is, it returns 1, as the factorial of 0 is 1.
* **else:**: If n is a positive integer, this block is executed.
* **result = 1**: Initializes a variable result to 1.
* **for i in range(1, n + 1):**: A for loop iterates from 1 up to and including n.
* **result \*= i**: In each iteration, result is multiplied by the current loop variable i. This accumulates the product of numbers from 1 to n.
* **return result**: After the loop, the function returns the calculated result.

This function calculates the factorial by repeatedly multiplying numbers in a loop, building up the result step by step.

**2. Recursive Factorial Function (recursive\_factorial)**

* **def recursive\_factorial(n):**: This defines a function named recursive\_factorial that also takes an integer n as input.
* **"""Calculates the factorial of a non-negative integer recursively."""**: This is the docstring for the recursive function.
* **if n < 0:**: Similar to the iterative function, this checks for negative input and returns an appropriate message.
* **elif n == 0:**: Checks if n is 0 and returns 1. This is the **base case** of the recursion, which stops the recursive calls.
* **else:**: If n is a positive integer, this block is executed.
* **return n \* recursive\_factorial(n - 1)**: This is the **recursive step**. The function returns the product of n and the result of calling itself with n - 1. This continues until the base case (n == 0) is reached.

This function calculates the factorial by breaking the problem down into smaller, self-similar subproblems until the base case is solved, and then combines the results back up.

**3. Getting Input and Calling Functions**

* **num\_with\_function = int(input("Enter a non-negative integer: "))**: This line prompts the user to enter a non-negative integer and stores it in the num\_with\_function variable.
* **factorial\_iterative = calculate\_factorial\_with\_function(num\_with\_function)**: This line calls the calculate\_factorial\_with\_function (iterative) with the user's input and stores the returned factorial in factorial\_iterative.
* **print("The factorial of", num\_with\_function, "using the iterative function is", factorial\_iterative)**: This prints the result from the iterative calculation, clearly labeling it.
* **factorial\_recursive = recursive\_factorial(num\_with\_function)**: This line calls the recursive\_factorial function with the user's input and stores the returned factorial in factorial\_recursive.
* **print("The factorial of", num\_with\_function, "using the recursive function is", factorial\_recursive)**: This prints the result from the recursive calculation, clearly labeling it.