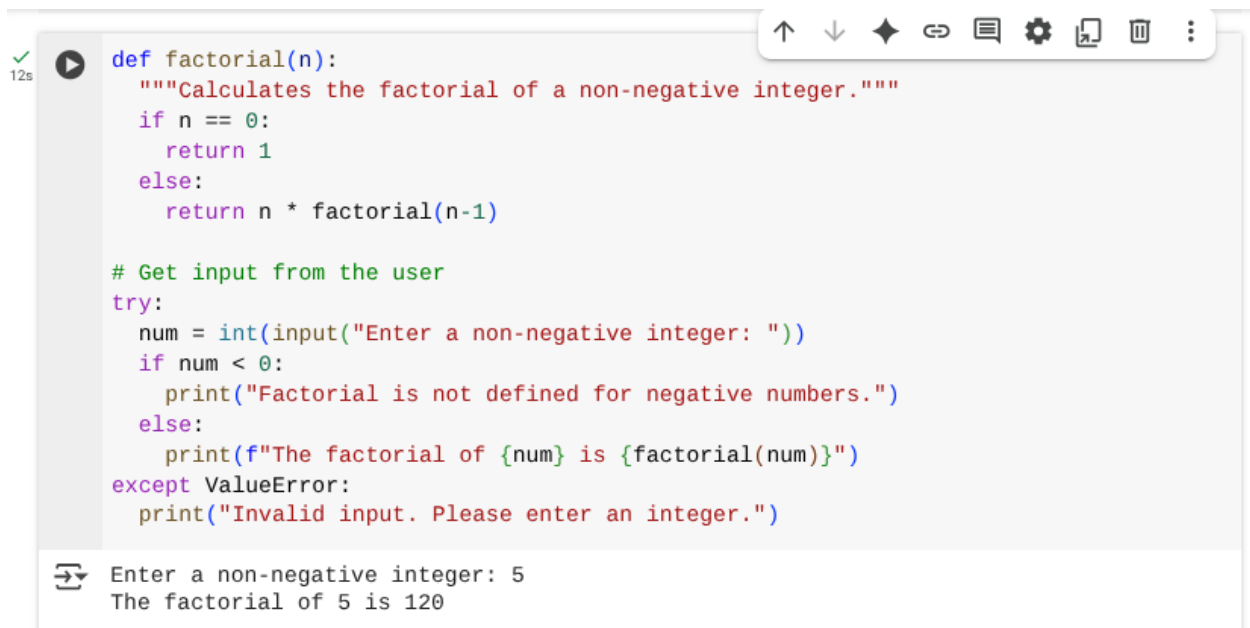


ASSIGNMENT-1

Task-1:

Write a python program that calculates the factorial of a number without defining my functions

Code and Output:



```
def factorial(n):
    """Calculates the factorial of a non-negative integer."""
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

# Get input from the user
try:
    num = int(input("Enter a non-negative integer: "))
    if num < 0:
        print("Factorial is not defined for negative numbers.")
    else:
        print(f"The factorial of {num} is {factorial(num)}")
except ValueError:
    print("Invalid input. Please enter an integer.")
```

Enter a non-negative integer: 5
The factorial of 5 is 120

Explanation:

1. `def factorial(n):` : Defines a function named `factorial` that takes one argument `n`.
 2. `if n == 0:` : Checks if the input number `n` is 0.
 3. `return 1` : If `n` is 0, it returns 1 (factorial of 0 is 1).
 4. `else:` : If `n` is not 0, it executes the following line.
 5. `return n * factorial(n-1)` : Recursively calls itself with `n-1` and multiplies the result by `n`.
 6. `try:` : Starts a block to handle potential errors during input.
 7. `num = int(input("Enter a non-negative integer: "))` : Prompts the user to enter an integer and converts the input to an integer.
 8. `if num < 0:` : Checks if the entered number is negative.
 9. `print("Factorial is not defined...")` : If negative, prints a message.
 10. `print(f"The factorial of {num} is {factorial(num)}")` : If non-negative, calculates and prints the factorial.
 11. `except ValueError:` : Handles the case where the input is not a valid integer.
 12. `print("Invalid input...")` : Prints an error message for invalid input.
-

Task-2:

Write a python program that calculates the factorial of a number with function and without function.

Code and Output:

- ♦ With function:

```
# Calculate factorial using a function

def factorial_with_function(n):
    """Calculates the factorial of a non-negative integer using a function."""
    if n == 0:
        return 1
    else:
        return n * factorial_with_function(n-1)

# Get input from the user
try:
    num_func = int(input("Enter a non-negative integer for function calculation: "))
    if num_func < 0:
        print("Factorial is not defined for negative numbers.")
    else:
        print(f"The factorial of {num_func} using a function is {factorial_with_function(num_func)}")
except ValueError:
    print("Invalid input. Please enter an integer.")
```

Enter a non-negative integer for function calculation: 6
The factorial of 6 using a function is 720

◆ Without function:

```
[3] # Calculate factorial without a function

try:
    num_no_func = int(input("Enter a non-negative integer for calculation without a function: "))
    if num_no_func < 0:
        print("Factorial is not defined for negative numbers.")
    else:
        factorial_result = 1
        for i in range(1, num_no_func + 1):
            factorial_result *= i
        print(f"The factorial of {num_no_func} without a function is {factorial_result}")
except ValueError:
    print("Invalid input. Please enter an integer.")
```

Enter a non-negative integer for calculation without a function: 6
The factorial of 6 without a function is 720

Explanation:

1. `def factorial_with_function(n):` : Defines a recursive function `factorial_with_function` to calculate factorial.
2. `if n == 0:` : Base case for the recursion: if `n` is 0, return 1.
3. `return n * factorial_with_function(n-1)` : Recursive step: multiply `n` by the factorial of `n-1`.
4. `try:` : Starts a block to handle potential errors during input for the function method.
5. `num_func = int(input(...))` : Prompts user for input and converts it to an integer.
6. `if num_func < 0:` : Checks for negative input.
7. `print(...)` : Prints the result of the factorial calculated by the function or an error message.
8. `try:` : Starts a block to handle potential errors during input for the non-function method.
9. `num_no_func = int(input(...))` : Prompts user for input and converts it to an integer.
10. `factorial_result = 1` : Initializes a variable to store the factorial result for the non-function method.
11. `for i in range(1, num_no_func + 1):` : Loops from 1 up to the input number.
12. `factorial_result *= i` : Multiplies the `factorial_result` by the current number in the loop.
13. `print(...)` : Prints the result of the factorial calculated without a function or an error message.
14. `except ValueError:` : Handles invalid input for both methods.
15. `print("Invalid input...")` : Prints an error message for invalid input.

Task-3:

Write a python program that calculates the factorial using both iterative and recursive functions.

Code and Output:

```
✓ [1] def factorial_iterative(n):  
      """Calculates the factorial of a number iteratively."""  
      if n < 0:  
          return "Factorial is not defined for negative numbers"  
      result = 1  
      for i in range(1, n + 1):  
          result *= i  
      return result  
  
def factorial_recursive(n):  
    """Calculates the factorial of a number recursively."""  
    if n < 0:  
        return "Factorial is not defined for negative numbers"  
    if n == 0:  
        return 1  
    else:  
        return n * factorial_recursive(n - 1)  
  
# Example usage:  
num = 5  
print(f"Factorial of {num} (iterative): {factorial_iterative(num)}")  
print(f"Factorial of {num} (recursive): {factorial_recursive(num)}")  
  
⇒ Factorial of 5 (iterative): 120  
   Factorial of 5 (recursive): 120
```

Explanation:

1. Iterative Function: `factorial_iterative(n)`

- Checks for negative input, returning an error message.
- Initializes `result` to 1.
- Uses a `for` loop from 1 to `n` to multiply `result` by each number in the range.
- Returns the final calculated factorial.

2. Recursive Function: `factorial_recursive(n)`

- Checks for negative input, returning an error message.
- Defines the base case: factorial of 0 is 1.
- For `n > 0`, it calls itself with `n-1` and multiplies the result by `n`.

Both functions are demonstrated with an example input `num = 5`. The output shows that both methods correctly calculate the factorial of 5 as 120.