Lab Manual

Practical and Skills Development

CERTIFICATE

PERFORMED BY

THE ASSIGNMENT ENTERED IN THIS REPORT HAVE BEEN SATISFACTORILY

Registration No	: 25BSA10126
Name of Student	: Deepak singh
Course Name	: Introduction to Problem Solving and Programming
Course Code	: CSE1021
School Name	: SCAI
Slot	: B11+B12+B13
Class ID	: BL2025260100796
Semester	: FALL 2025/26

Course Faculty Name : Dr. Hemraj S. Lamkuche

Signature:

QUESTION 1 :Factorial Calculation and Divisor Count Analysis

AIM/OBJECTIVE(s):

1. To calculate the **factorial** of a user-input number (n).
2. To measure the **execution time** required for the factorial calculation.
3. To count the total number of **divisors** for the input number (n).
4. To estimate the approximate **memory used** based on the number of divisors.

METHODOLOGY & TOOL USED:

. Methodology: Iterative Calculation: A for loop is used to calculate the factorial by repeatedly multiplying a running total (t). A separate for loop is used to check for and count the divisors of the input number (n) **.**

Tool Used: Python programming language, utilizing the built-in time module for performance measurement.

BRIEF DESCRIPTION:

The program first prompts the user to **enter a number (n)**. It then immediately starts a timer. It calculates the factorial of n using a loop that runs from 1 to n, accumulating the product in the variable t. After the loop completes, it stops the timer and prints the **factorial** and the **execution time**. Finally, a second loop iterates from 1 to n to count how many times n is perfectly divisible, storing the count in divisors. This count is then used to give a rough **approximation of memory used** (by multiplying the divisor count by 28).

RESULTS ACHIEVED:

The program successfully achieves the following for a given input number n:

1. **Factorial (n!)**: The computed product of all positive integers less than or equal to n.

2. **Execution Time**: The time taken (in seconds) for the factorial calculation loop to complete.
3. **Divisors Count**: The total number of positive integers that divide n without a remainder.
4. **Memory Used (Approx.)**: An estimated memory usage value based on the calculated divisor count.

DIFFICULTY FACED BY STUDENT:

Large Numbers: For relatively large inputs (n≥20), the factorial quickly becomes an extremely large number that can exceed the limits of standard integer data types (though Python handles large integers automatically, the calculation time increases significantly).

Approximation: The memory calculation (divisors * 28) is a simple, non-standard **approximation** that doesn't reflect the actual memory allocation of the program, which might confuse a student trying to understand real memory usage.

Performance: Understanding the limitations of measuring such short execution times with the time.time() function, which can be affected by system load and clock resolution.

SKILLS ACHIEVED:

Input/Output Handling: Taking user input (input()) and displaying results (print()).

Looping Constructs: Effective use of the for loop for iteration.

Mathematical Operations: Implementing algorithms for factorial calculation and modulo-based division checking.

Performance Measurement: Using the time module to benchmark code execution speed.

Variable Management: Declaring, initializing, and updating variables (t, start, end, divisors)