OOP LAB 13 TASKS

1. Write a program where an exception is thrown from a function deep down the call stack. Create a chain of function calls (e.g., funcA() -> funcB() -> funcC() -> throwException()). Inside throwException(), throw a custom exception. Ensure that each function in the call chain has appropriate try and catch blocks to catch and handle this exception. Observe how the exception is propagated up the call stack and how stack unwinding occurs.

2. Create a program with multiple levels of function calls. Implement a function **processData()** that calls another function **parseData()**, which in turn may throw exceptions related to data parsing errors (e.g., invalid format). Use exception handling in **processData** to catch these exceptions, log an error message, and rethrow the exception with additional context (if needed). Catch the rethrown exception in **main** and display an appropriate error message.

3. Create a template function **matrixMultiply** that performs matrix multiplication for two-dimensional arrays (matrices) of any numeric type (**int**, **double**, etc.).

4. Create a template function **factorial** that computes the factorial of a given non-negative integer using template metaprogramming (compile-time computation).'

5. Create a class containing a function template capable of returning the sum of all the elements in an array being passed as parameter. Show the results for two arrays, one integer type and the other double type.