Title: Matrix Multiplication Using OpenMP

Objective:
To multiply two square matrices using **parallel programming** with **OpenMP** in C++, which makes the program run faster by using multiple CPU cores.

What is Matrix Multiplication?
Matrix multiplication is a way to combine two matrices (A and B) to get a new matrix (C).
Each number in the result matrix is calculated by multiplying rows of A with columns of B and adding the results.

What is OpenMP?
OpenMP is a tool in C++ that helps us run parts of the program at the same time (in parallel) using multiple processor cores.
This makes the program finish faster, especially when working with large data like big matrices.

Steps in the Program:

1. **Input Matrix Size:**
   The user is asked to enter the size (n) of the square matrices (like 2 for 2x2, 3 for 3x3, etc.).
2. **Input Matrix Elements:**
   The user enters the values for Matrix A and Matrix B.
3. **Initialize Result Matrix (C):**
   A new matrix C is created with all values set to zero.
4. **Parallel Matrix Multiplication:**
   - The program uses OpenMP to split the multiplication work among multiple CPU cores.
   - This helps in performing the task faster.
5. **Time Measurement:**
   The program checks how long the multiplication takes using a timer.
6. **Display Result:**
   The final result matrix (C) is shown to the user.
   The time taken is also printed.

Why Use OpenMP?

- To make the program **faster** by running multiple parts at the same time.
- It is **easy to use** – just by adding special lines before loops.
- Helps in **handling big calculations** more efficiently.

Conclusion:
This program shows how we can use OpenMP to speed up matrix multiplication.
It is useful in fields like **Artificial Intelligence**, **Machine Learning**, and **Scientific Computing**, where such operations are done many times on large amounts of data.