```
#include <iostream> // --> Includes input-output stream for console I/O

#include <omp.h> // --> Includes OpenMP header for parallel programming

using namespace std; // --> Allows usage of standard namespace to avoid std:: prefix


int main() // --> Entry point of the program

{

  int n; // --> Declares variable to store size of square matrices

  cout << "\nName: Shriharsh Deshmukh\nRoll No.62 \t Div.A\n"; // --> Displays name, roll
number, and division

  cout << "\nEnter the size of the square matrices (e.g. 3 for 3x3): "; // --> Prompts user to enter
size of matrices

  cin >> n; // --> Takes input from user for matrix size


  float A[n][n], B[n][n], C[n][n]; // --> Declares 2D arrays for matrices A, B, and result matrix C


  cout << "\nEnter elements of Matrix A:\n"; // --> Prompts user to enter elements of matrix A

  for (int i = 0; i < n; i++) // --> Loops through rows of matrix A

    for (int j = 0; j < n; j++) // --> Loops through columns of matrix A

      cin >> A[i][j]; // --> Takes input for element A[i][j]


  cout << "\nEnter elements of Matrix B:\n"; // --> Prompts user to enter elements of matrix B

  for (int i = 0; i < n; i++) // --> Loops through rows of matrix B

    for (int j = 0; j < n; j++) // --> Loops through columns of matrix B

      cin >> B[i][j]; // --> Takes input for element B[i][j]


#pragma omp parallel for collapse(2) // --> Parallelizes nested loops to initialize C with 0

  for (int i = 0; i < n; i++) // --> Loops through rows of matrix C

    for (int j = 0; j < n; j++) // --> Loops through columns of matrix C

      C[i][j] = 0; // --> Initializes C[i][j] to 0


  double start = omp_get_wtime(); // --> Records start time using OpenMP timer
#pragma omp parallel for collapse(2) // --> Parallelizes nested loops for matrix multiplication
  for (int i = 0; i < n; i++) // --> Loops through rows of matrix A
    for (int j = 0; j < n; j++) // --> Loops through columns of matrix B
      for (int k = 0; k < n; k++) // --> Loops through columns of matrix A / rows of matrix B
        C[i][j] += A[i][k] * B[k][j]; // --> Performs multiplication and accumulation for C[i][j]

  double end = omp_get_wtime(); // --> Records end time using OpenMP timer

  cout << "\nResultant Matrix C = A x B:\n"; // --> Displays header for resulting matrix
  for (int i = 0; i < n; i++) // --> Loops through rows of matrix C
  {
    for (int j = 0; j < n; j++) // --> Loops through columns of matrix C
      cout << C[i][j] << "\t"; // --> Prints element C[i][j] followed by a tab
    cout << endl; // --> Moves to next line after each row
  }

  cout << "\n Matrix multiplication done using OpenMP."; // --> Prints message indicating
completion
  cout << "\n Time taken: " << end - start << " seconds\n"; // --> Displays time taken for
multiplication

  return 0; // --> Returns 0 indicating successful program termination
}
```