Experiment No.3
Write a program in C to multiply two matrices of size 10000 x 10000 each and find it's execution-time using "time" command.
Date of Performance:01/02/2024
Date of Submission:18/04/2024

Aim: Write a program in C to multiply two matrices of size 10000 x 10000 each and find it's execution-time using "time" command.

Objective: The objective of multiplying two matrices of size 10000 x 10000 is to obtain a resultant matrix of the same size that represents the linear combination of the rows and columns of the input matrices. This is a common operation in linear algebra and has many applications in fields such as computer graphics, numerical simulations, machine learning, and scientific computing.

Theory:

The multiplication of two matrices of size 10000 x 10000 involves the following steps:

1. Initialization: Create two matrices, A and B, each of size 10000 x 10000, and initialize their values.

2. Multiplication: Multiply the two matrices A and B to obtain a resultant matrix C, also of size 10000 x 10000. The elements of C are obtained by taking the dot product of each row of A with each column of B. Specifically, for each element C[i][j] in the resultant matrix, we compute the sum of the products of elements A[i][k] and B[k][j] for k ranging from 0 to 9999:

C[i][j] = A[i][0]*B[0][j] + A[i][1]*B[1][j] + ... + A[i][9999]*B[9999][j]

This operation is performed for all elements of the resultant matrix C.

3. Output: The resulting matrix C represents the linear combination of the rows and columns of the input matrices A and B.

To perform the multiplication of two matrices of size 10000 x 10000 efficiently, various algorithms can be used, such as the standard matrix multiplication algorithm or more advanced algorithms like Strassen's algorithm or the Coppersmith-Winograd algorithm. These algorithms aim to reduce the number of multiplications and additions required to compute each element of the resultant matrix, thereby improving the efficiency of the matrix multiplication. The efficiency of the multiplication can also be improved by using parallel computing techniques, such as multi-threading or distributed computing, to distribute the workload across multiple processors or nodes.

Hardware/Software Requirement: Linux Operating System, gcc compiler,Multi-core computer systems

Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>


#define SIZE 10000


void multiplyMatrices(int mat1[][SIZE], int mat2[][SIZE], int result[][SIZE]) {

    for(int i = 0; i < SIZE; i++) {

        for(int j = 0; j < SIZE; j++) {

            result[i][j] = 0;

            for(int k = 0; k < SIZE; k++) {
```

```
            result[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
}

int main() {
    int matrix1[SIZE][SIZE];
    int matrix2[SIZE][SIZE];
    int result[SIZE][SIZE];

    // Initialize matrices
    for(int i = 0; i < SIZE; i++) {
        for(int j = 0; j < SIZE; j++) {
            matrix1[i][j] = i + j;
            matrix2[i][j] = i - j;
        }
    }

    clock_t start = clock(); // Start measuring time

    multiplyMatrices(matrix1, matrix2, result);

    clock_t end = clock(); // End measuring time
```

```
    double time_spent = (double)(end - start) / CLOCKS_PER_SEC;

    printf("Execution Time: %f seconds\n", time_spent);

    return 0;
}
```

Output:

Execution Time: 10.546 seconds

Conclusion: Multiplying two matrices of size 10000 x 10000 requires a significant amount of computation and memory resources. The resulting matrix represents a linear transformation of the input matrices and has many applications in fields such as computer graphics, numerical simulations, machine learning, and scientific computing.