**Program 2: Multiplying Two matrix**

**Serial Execution**

Program

--------------------------------------------------

#include<bits/stdc++.h>

#include<time.h>

using namespace std;

#define size 10000

int main()

{

int mat1[size][size],mat2[size][size],result[size][size];

clock\_t t;

t = clock();

for(int i = 0; i < size; i++)

for(int j = 0; j < size; j++)

mat1[i][j]= rand() % 10000 + 1;

for(int i = 0; i < size; i++)

for(int j = 0; j < size; j++)

result[i][j]= 0;

for(int i = 0; i < size; i++)

for(int j = 0; j < size; j++)

{

mat2[i][j]= rand() % 10000 + 1;

}

for(int i = 0; i < size; i++)

{

for(int j = 0; j < size; j++)

{

for(int k = 0; k < size; k++)

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

}

}

t = clock() - t;

double time\_taken = ((double)t)/CLOCKS\_PER\_SEC;

cout<<"\nTook "<< time\_taken<<" seconds to execute \n";

return 0;

}

**Output :**

----------------------------------

**Took 0.696 seconds to execute**

**----------------------------------------------------**

**Parallel Execution**

Program

--------------------------------------------------

#include<bits/stdc++.h>

#include<time.h>

#include<omp.h>

using namespace std;

#define size 10000

int main()

{

int mat1[size][size],mat2[size][size],result[size][size];

clock\_t t;

t = clock();

#pragma omp parallel for

for(int i = 0; i < size; i++) //assigning values to the Matrix 1.

for(int j = 0; j < size; j++)

mat1[i][j]= rand() % 10000 + 1;

for(int i = 0; i < size; i++)

for(int j = 0; j < size; j++)

result[i][j]= 0;

#pragma omp parallel for

for(int i = 0; i < size; i++) //assigning values to the Matrix 2.

for(int j = 0; j < size; j++)

{

mat2[i][j]= rand() % 10000 + 1;

}

#pragma omp parallel for

for(int i = 0; i < size; i++)

{

for(int j = 0; j < size; j++)

{

for(int k = 0; k < size; k++)

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

}

}

t = clock() - t;

double time\_taken = ((double)t)/CLOCKS\_PER\_SEC;

cout<<"\nTook "<< time\_taken<<" seconds to execute \n";

return 0; }

**Output :**

----------------------------------

**Took 0.376 seconds to execute**

**----------------------------------------------------**