**Assignment No 4**

Name: Poonam Kisan Salbande

Roll No: 20121011

Title: Write a CUDA Program for :

1. Addition of two large vectors

2. Matrix Multiplication using CUDA C

Code:-

#include<stdio.h>

#include<iostream>

#include<cstdlib>/\* CUDA Library \*/

#include<omp.h>

#define MAX 100

int main()

{

int m1[MAX], m2[MAX], m3[MAX], i;

printf("\n First Vector:\t");

#pragma omp parallel for

for(i=0; i<MAX; i++)

{

m1[i]=rand()%1000;

}

for(i=0;i<MAX; i++)

{

printf("%d\t",m1[i]);

}

printf("\n Second Vector:\t");

#pragma omp parallel for

for(i=0; i<MAX; i++)

{

m2[i]=rand()%1000;

}

for(i=0;i<MAX; i++)

{

printf("%d\t",m2[i]);

}

printf("\n Parallel-vector Addition:(m1,m2,m3)\t");

#pragma omp parallel for

for(i=0; i<MAX; i++)

{

m3[i]=m1[i]+m2[i];

}

for(i=0;i<MAX; i++)

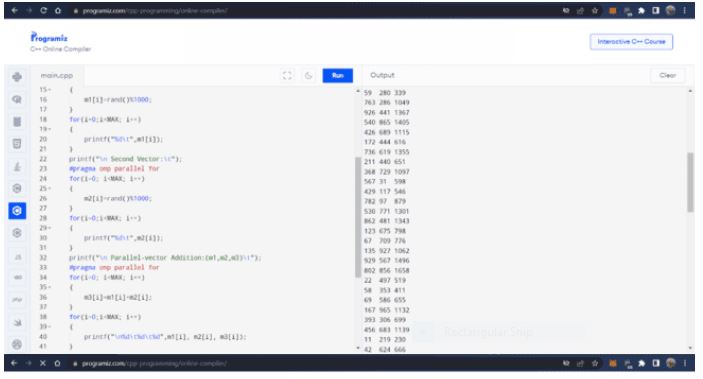
{

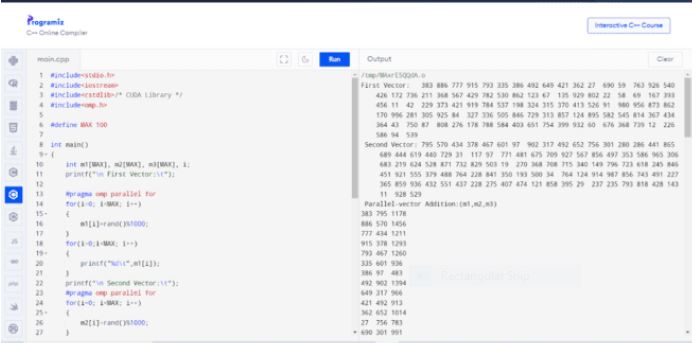
printf("\n%d\t%d\t%d",m1[i], m2[i], m3[i]);

}

}

Output:





**Matrix multiplication using CUDA:-**

#include <stdio.h>

#include <iostream>

#include <cstdlib>

#include <omp.h>

#define MAX 100

using namespace std; // Add this line to use cout and endl

int main()

{

int r = 3, c = 2;

int matrix[r][c], vector[c], out[r];

for (int row = 0; row < r; row++)

{

for (int col = 0; col < c; col++)

{

matrix[row][col] = 1;

}

}

cout << "Input Matrix" << endl; // Use endl instead of end1

for (int row = 0; row < r; row++)

{

for (int col = 0; col < c; col++)

{

cout << "\t" << matrix[row][col];

}

cout << "" << endl; // Use endl instead of end1

}

for (int col = 0; col < c; col++) // Change row to col

{

vector[col] = 2;

}

cout << "Input Col-Vector" << endl; // Use endl instead of end1

for (int col = 0; col < c; col++) // Change row to col

{

cout << vector[col] << endl; // Use endl instead of end1

}

#pragma omp parallel // Move the parallel region outside the for loop

{

#pragma omp for // Remove the inner parallel region

for (int row = 0; row < r; row++)

{

out[row] = 0;

for (int col = 0; col < c; col++) // Remove comma from for loop

{

out[row] += matrix[row][col] \* vector[col];

}

}

}

cout << "Resultant Col-Vector" << endl; // Use endl instead of end1

for (int row = 0; row < r; row++)

{

cout << "\nvector[" << row << "]:" << out[row] << endl; // Use endl instead of end1

}

return 0;

}

Output:

