Vector

#include <stdio.h>

#include <iostream>

#include <omp.h>

#include <cstdlib>

using namespace std;

#define MAX 10

int main() {

int a[MAX], b[MAX], c[MAX], i;

printf("\nFirst Vector:\t");

#pragma omp parallel for

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

a[i] = rand() % 1000;

}

// Sequential loop, values printed in order

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

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

}

printf("\nSecond Vector:\t");

#pragma omp parallel for

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

b[i] = rand() % 1000;

}

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

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

}

printf("\nParallel-Vector Addition:(a,b,c)\t");

#pragma omp parallel for

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

c[i] = a[i] + b[i];

}

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

printf("\n%d\t%d\t%d", a[i], b[i], c[i]);

}

return 0;

}

Matrix

#include <iostream>

#include <cstdlib>

#include <omp.h>

using namespace std;

int main() {

int a[10][10], b[10][10], mul[10][10], row, column, i, j, k;

cout << "Enter the number of rows: ";

cin >> row;

cout << "Enter the number of columns: ";

cin >> column;

cout << "Enter the first matrix elements:\n";

for (i = 0; i < row; i++) {

for (j = 0; j < column; j++) {

cin >> a[i][j];

}

}

cout << "Enter the second matrix elements:\n";

for (i = 0; i < row; i++) {

for (j = 0; j < column; j++) {

cin >> b[i][j];

}

}

cout << "Multiplication of the matrices:\n";

#pragma omp parallel for private(j, k)

for (i = 0; i < row; i++) {

for (j = 0; j < column; j++) {

mul[i][j] = 0;

for (k = 0; k < column; k++) {

mul[i][j] += a[i][k] \* b[k][j];

}

}

}

// Printing the result

for (i = 0; i < row; i++) {

for (j = 0; j < column; j++) {

cout << mul[i][j] << " ";

}

cout << "\n";

}

return 0;

}