**Русенски Университет**

**„Ангел Кънчев“**

**Факултет**

**„ПРИРОДНИ НАУКИ И ОБРАЗУВАНИЕ“**

**Катедра**

**Информатика и Информационни Технологии**

**КУРСОВА ЗАДАЧА**

**по**

**дисциплината**

**„Паралелно програмиране“**

**изготвил: Иво Тодоров**

**специалност: СИ**

**фак.№: 206266**

**Дата: 21.04.2024 проверил: …………………………**

/гл. ас. Сергей Антонов/

**Задание 10**

Да се състави паралелна програма с OpenMP, която инициализира двумерен масив със случайни числа от тип double и намира най-големия елемент на всеки стълб и номера на реда, в който се намира.

**Описание**

Използва се функцията srand(time(NULL)); за генериране на случайни числа. След това се извършва инициализация на двумерния масив (в този случай матрица с 10 реда и 5 колни) matrix със случайни числа между 0 и 10. Чрез двойни for цикли обхождаме матрицата ставнявайки всеки елемент и запазвайки най-големия елемент в max\_values[j] и номерът на реда му в max\_row\_num[j]. Накрая се извеждат най-големите елементи за всяка колона на матрицата, както и техните редове, заедно със самата матрица.

**Текст на програмата**

#include <iostream>

#include <omp.h>

#include <cstdlib>

#include <ctime>

#define ROWS 10

#define COLS 5

using namespace std;

int main() {

srand(time(NULL));

double matrix[ROWS][COLS];

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

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

matrix[i][j] = rand() % 1000 / 100.0;

}

}

double max\_values[COLS];

int max\_row\_num[COLS];

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

max\_values[j] = matrix[0][j];

max\_row\_num[j] = 0;

}

#pragma omp parallel for

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

double max\_val = matrix[0][j];

int max\_row\_index = 0;

for (int i = 1; i < ROWS; ++i) {

if (matrix[i][j] > max\_val) {

max\_val = matrix[i][j];

max\_row\_index = i;

}

}

#pragma omp critical

{

max\_values[j] = max\_val;

max\_row\_num[j] = max\_row\_index;

}

}

cout << „Biggest elements in each column:“ << endl;

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

cout << „Column „ << j+1 << „: Biggest element – „ << max\_values[j] << „, Row – „ << max\_row\_num[j]+1 << endl;

}

cout << „\nMatrix:“ << endl;

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

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

cout << matrix[i][j] << „ „;

}

cout << endl;

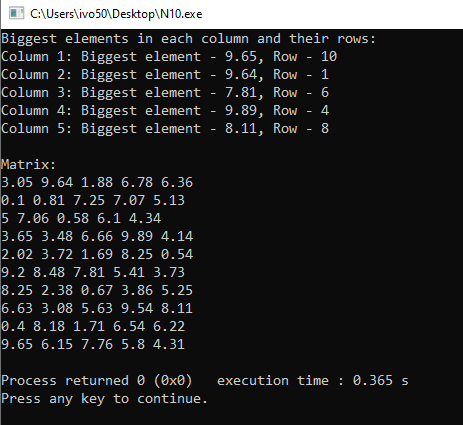
}

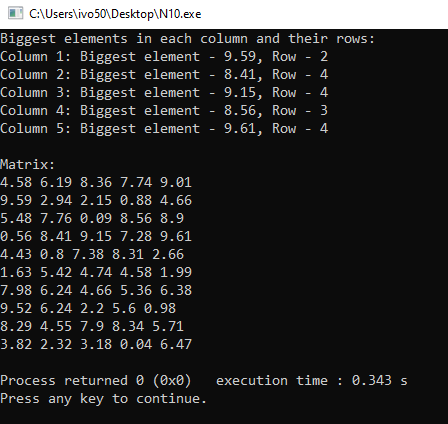
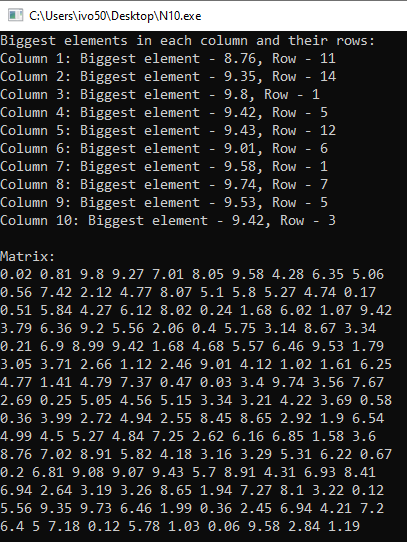
return 0;

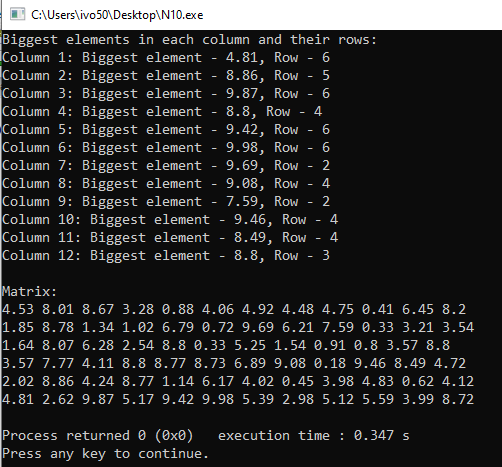
}

**Тестване**

matrix[10][5] matrix[10][5]

****

****

matrix[15][10] matrix[6][12]