Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 14

По дисциплине “Основы алгоритмизации и программирования”

На тему “**Многомерные массивы** ”

Выполнил:

Студент 1 курса 6 группы

Кравченко Сергей Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 8

|  |  |
| --- | --- |
| **8** | 1. Даны две целочисленные матрицы **A(n, m)** и **B(n, m)**. Подсчитать количество тех пар **(ai j, bi j)** , для которых: а) **ai j< bi j**; б) **ai j= bi j**; в) **ai j> bi j**.  2. Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер. Все элементы столбца с таким же номером уменьшить вдвое. |

1.

#include <iostream>

#include <ctime>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

const int ROW = 4; //количество строк массива

const int COL = 4; //количество столбцов массива

int a[ROW][COL];

int b[ROW][COL];

int g=0, h=0, k=0;

srand(time(NULL));

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

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

a[i][j] = rand() % 100; //заполняем массив а рандомными числами

}

}

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

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

b[i][j] = rand() % 100; //заполняем массив b рандомными числами

}

}

cout << "Первая матрица" << endl;

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

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

cout << a[i][j] << " "; //выводим первый массив

}

cout << endl;

}

cout << "Вторая матрица" << endl;

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

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

cout << b[i][j] << " "; //выводим второй массив

}

cout << endl;

}

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

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

if (a[i][j] < b[i][j]) { //проверка условий больше, меньше либо равно

g++;

}

else if (a[i][j] == b[i][j]) {

h++;

}

else if (a[i][j] > b[i][j]) {

k++;

}

}

}

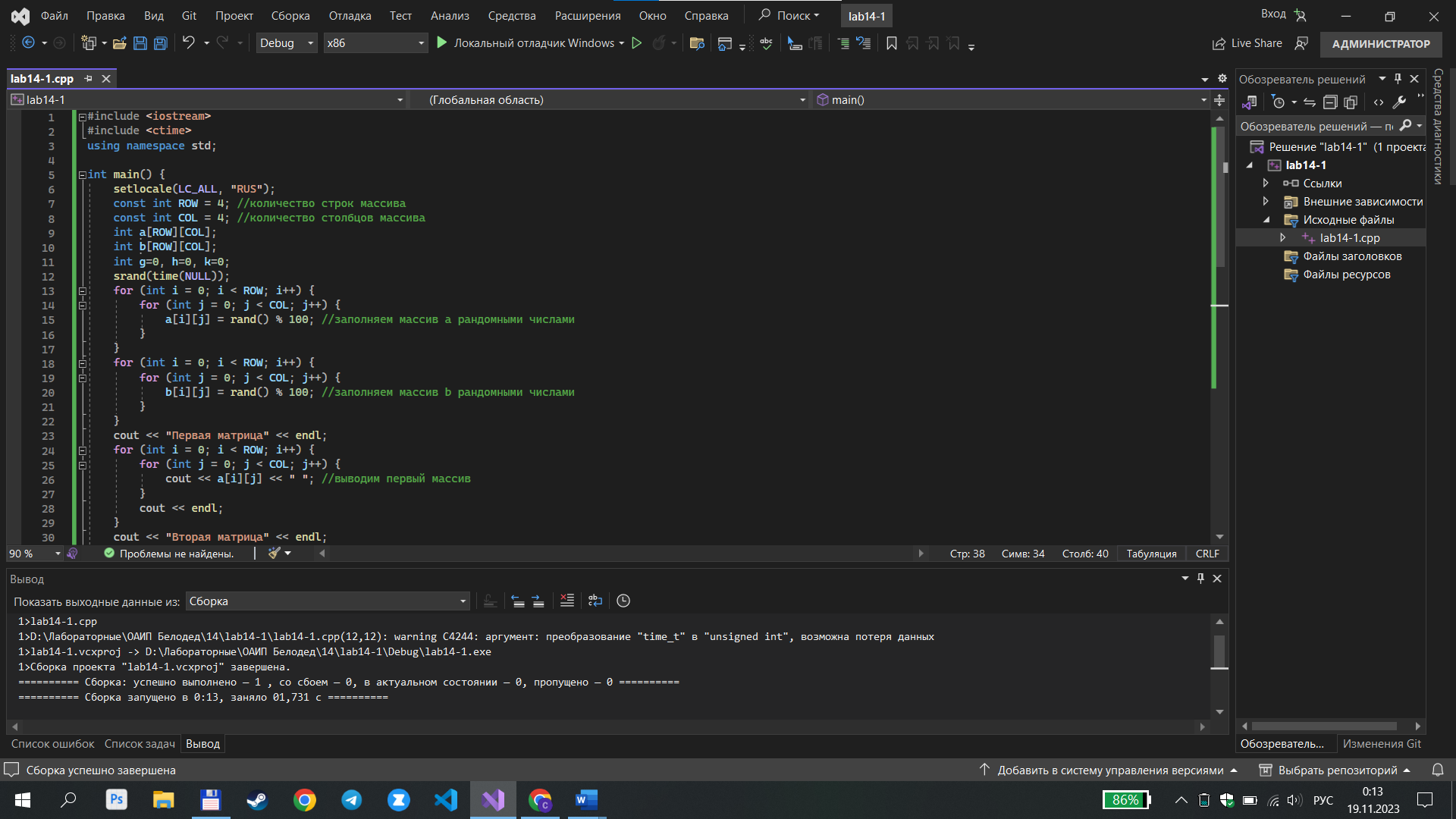
//выводим количество совпадений

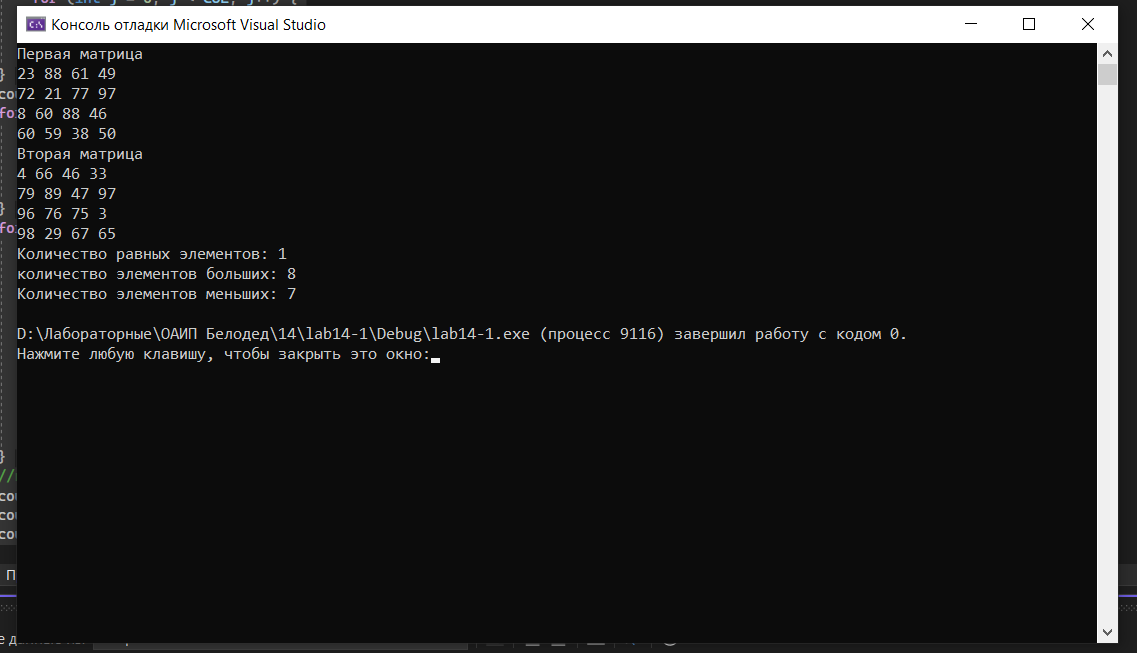
cout << "Количество равных элементов: " << h << endl;

cout << "количество элементов больших: " << k << endl;

cout << "Количество элементов меньших: " << g << endl;

}





2.

#include <iostream>

int main() {

setlocale(LC\_ALL, "RUS");

using namespace std;

int ROWS, COLS, neg\_elem\_line = -1; //переменные

cout << "Введите количество строк\n"; cin >> ROWS;

cout << "Введите количество столбцов\n"; cin >> COLS;

double\*\* a = new double\* [ROWS]; //Создаем двумерный массив a с количеством строк, равным ROWS

for (int i = 0; i < ROWS; ++i) { //Иициализируем каждую строку массива a массивом double длиной COLS

a[i] = new double[COLS];

}

cout << "Введите элементы:\n";

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

for (int j = 0; j < COLS; j++) { //Считываем элементы матрицы из ввода пользователя

cin >> a[i][j];

}

}

cout << "Первая матрица:\n";

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

for (int j = 0; j < COLS; j++) { //выводим первую матрицу

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

}

cout << "\n";

}

for (int i = 0; i < ROWS; i++) { //Ищем первую строку с отрицательным элементом

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

if (a[i][j] < 0) {

neg\_elem\_line = i;

break;

}

}

if (neg\_elem\_line != -1) break;

}

if (neg\_elem\_line != -1) { //Делим каждый элемент в строке с отрицательным элементом на 2, если такая строка найдена

for (int j = neg\_elem\_line, i = 0; i < COLS; i++) {

a[i][j] /= 2;

}

cout << "Результат:\n";

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

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

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

}

cout << "\n";

}

}

else {

cout << "Здесь нет отрицательных элементов\n";

}

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

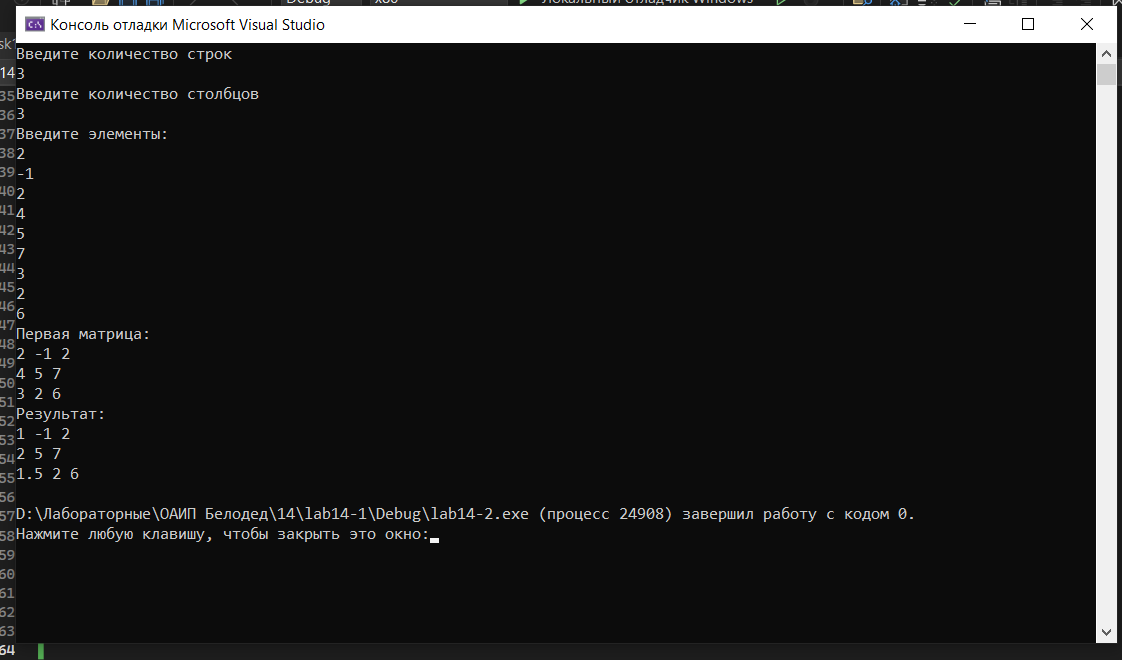
delete[] a[i]; //очищаем буфер

}

delete[] a;

return 0;

}



Доп задания

1. Дана квадратная матрица порядка **2n**, элементы которой фо рмируются случайным образом и находятся в пределах от −10 до 10. Получить новую матрицу, переставляя ее блоки размера **n×n** в соответствии со схемой

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

const int n = 2; // размер матрицы

int a[n][n];

int b[n][n];

// Заполняем матрицу

srand(time(NULL));

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

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

a[i][j] = rand() % 21 - 10; // заполняем матрицу случайными числами от -10 до 10

}

}

cout << "Начальная Матрица" << endl;

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

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

cout << a[i][j] << " "; //выводим массив

}

cout << endl;

}

// Переставляем блоки матрицы

for (int i = 0; i < n; i += 2) {

for (int j = 0; j < n; j += 2) {

// Переставляем блоки в соответствии со схемой

b[i][j] = a[i + 1][j + 1];

b[i + 1][j] = a[i][j + 1];

b[i + 1][j + 1] = a[i][j];

b[i][j + 1] = a[i + 1][j];

}

}

cout << "Матрица после перестановок: " << endl;

// Выводим полученную матрицу

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

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

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

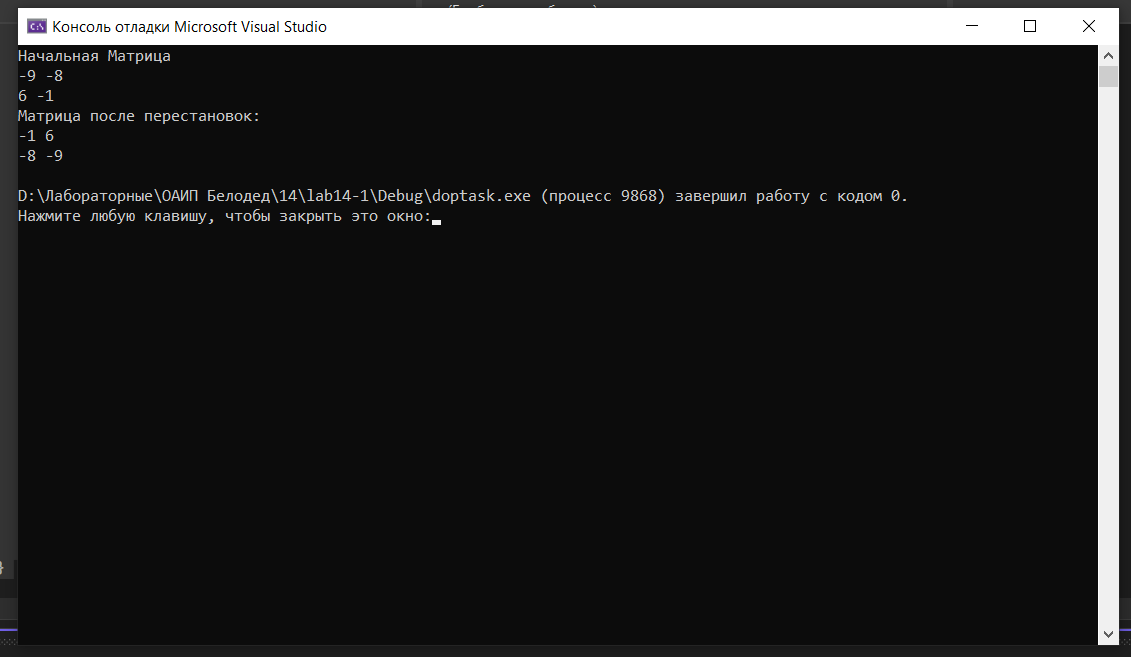
}

cout << endl;

}

return 0;

}



2.

Латинским квадратом порядка **n** называется квадратная таблица размером **nхn**, каждая строка и каждый столбец которой содержат все числа от 1 до **n**. Для заданного **n** в матрице **L(n, n**) построить латинский квадрат порядка **n**.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

int n, m;

int a[100][100]; // создаём двумерный массив

cout << "Введите размер массива (количество строк и столбцов)";

cin >> n; //считываем размер массива

for (int i = 0; i < n; i++) { // начинаем цикл по строкам массива

m = i + 1;

for (int j = 0; j < n; j++) { //Начинается цикл по столбцам массива

a[i][j] = m; //Заполняется элемент массива a[i][j] значением m

m += 1;

if (m > n) { //Если m больше n, то m сбрасывается обратно до 1

m = 1;

}

}

}

for (int i = 0; i < n; i++) { //Начинается цикл для вывода массива на экран.

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

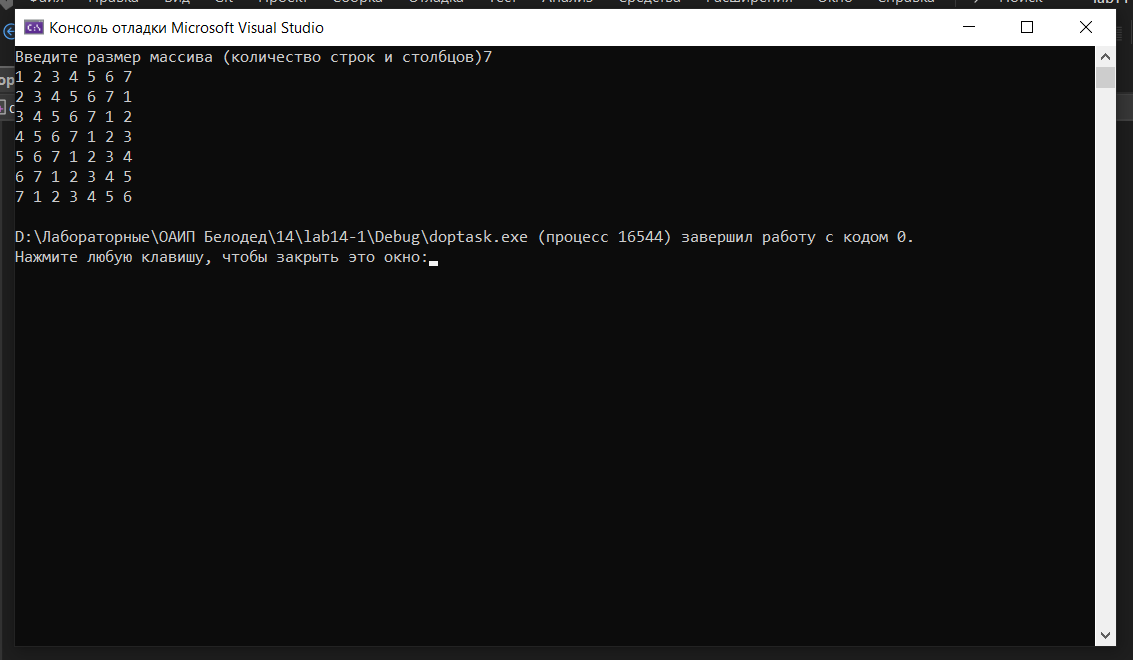
cout << a[i][j] << " "; //Выводится значение элемента массива a[i][j].

}

cout << endl;

}

}



3. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине − в позиции (2, 2), следующий − в позиции (3, 3) и т. д., заполнив таким образом всю главную диагональ

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

int n;

double buf; //переменная

double a[200][200]; //создаем массив

double b[400] = { 0 };

int k = 0;

cout << "Введите размер массива (количество строк и столбцов)";

cin >> n; //считываем размер массива

for (int i = 0; i < n; i++) { //Начинается цикл по строкам массива

for (int j = 0; j < n; j++) { //Начинается цикл по столбцам массива

cin >> a[i][j];

b[k] = a[i][j]; //Записывается элемент матрицы в массив b и увеличивается счетчик k

k++;

}

}

for (int i = 0; i < k; i++) { //Начинается цикл для сортировки массива

for (int j = (k - 1); j >= (i + 1); j--) { // Начинается цикл для сортировки массива в обратном порядке.

if (b[j] < b[j - 1]) { //Если текущий элемент меньше предыдущего, то выполняется блок кода

buf = b[j];

b[j] = b[j - 1];

b[j - 1] = buf; //Элементы меняются местами

}

}

}

k--;

for (int i = 0; i < n; i++) { //Начинается цикл для замены элементов главной диагонали

a[i][i] = b[k]; //Элемент главной диагонали заменяется на соответствующий элемент отсортированного массива и уменьшается счетчик k

k--;

}

for (int i = 0; i < n; i++) { //Начинается цикл для вывода матрицы на экран

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

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

}

cout << endl;

}

}

