Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 14

По дисциплине «Основы алгоритмизации и программирования»

На тему «Многомерные массивы»

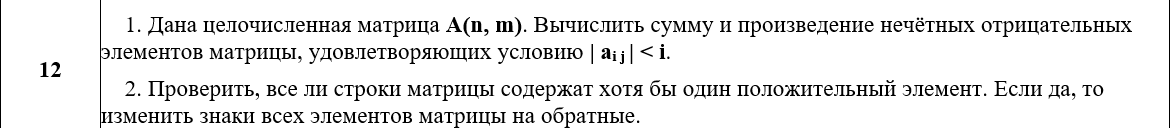
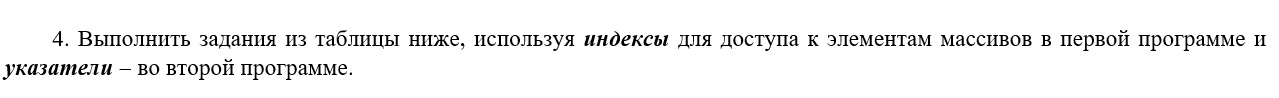
Выполнил:

Студент 1 курса 6 группы

Романов Игорь Вячеславович

Преподаватель: асс. Андронова М.В.

2023, Минск



1.

#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int n = 2, m = 4; //ввод размера массива

int A[n][m]{}; //инициализация массива

int i, j, sum = 0, position = 0, product = 1; //ввод нумерации строки, столбца массива; суммы и произведения элементов; позиции элементов

cout << "Введите элементы массива размера 2x4: " << endl; //ввод элементов массива с клавиатуры

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

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

cin >> A[i][j];

}

}

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

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

if (position % 2 != 0) { //если позиция элемента нечетная

if (abs(A[i][j]) < position && A[i][j] < 0) { //проверить условие (элемент должен быть отрицательным, а его модуль меньше, чем его позиция элемента)

sum += A[i][j];

product \*= A[i][j];

}

}

position++;

}

}

cout << "\nИсходный массив:" << endl; //вывод на экран исходного массива

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

{

cout << "\n";

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

cout << "A[" << i << "," << j << "] = " << A[i][j] << "\t";

}

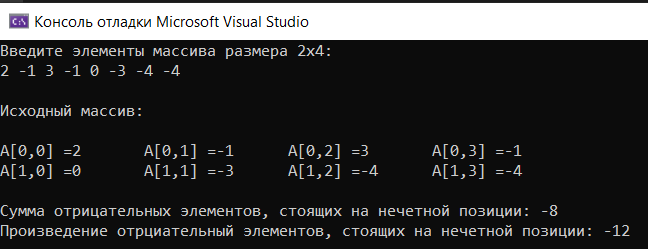
}

cout << endl;

cout << "\nСумма отрицательных элементов, стоящих на нечетной позиции: " << sum << endl; //вывод суммы

cout << "Произведение отрицательный элементов, стоящих на нечетной позиции: " << product << endl; //вывод произведения

}



2.

#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int n = 2, m = 4; //ввод размера массива

int A[n][m]{}; //инициализация массива

int i, j, positive = 0; //ввод нумерации строки, столбца массива; счетчика положительных элементов

cout << "Введите элементы массива размера 2x4: " << endl; //ввод элементов массива с клавиатуры

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

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

cin >> A[i][j];

}

}

cout << "\nИсходный массив:" << endl; //вывод на экран исходного массива

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

cout << "\n";

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

cout << "A[" << i << "," << j << "] = " << A[i][j] << "\t";

}

}

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

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

if (\*(\*(A + i) + j) > 0) { //если находится положительный элемент массива, то увеличить счетчика

positive++;

}

}

}

if (positive) { //если значение счетчика истинно, то умножить элементы массива на -1

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

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

\*(\*(A + i) + j) \*= -1;

}

}

cout << "\nПреобразованный массив:" << endl; //вывод на экран полученного массива

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

{

cout << "\n";

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

cout << "A[" << i << "," << j << "] = " << A[i][j] << "\t";

}

}

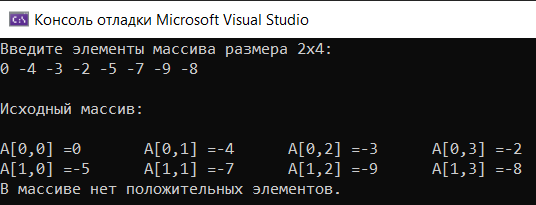
}

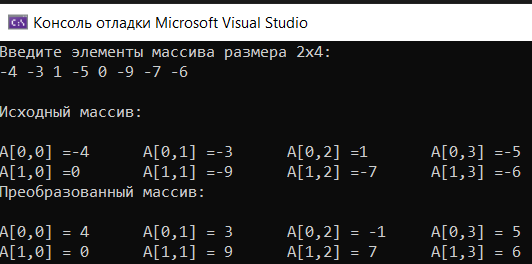
if (!positive) { //если значение счетчика ложно, то вывести соответствующее сообщение

cout << "\nВ массиве нет положительных элементов.";

}

}





Доп. задания

Вариант 1



#include <iostream>

#include <ctime> //для работы с функцией time

void main()

{

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

using namespace std;

int A[100][100]{}; //инициализация массива

int i, j, n, m, temp; //ввод нумерации строки, столбца массива; временной переменной для перестановки элементов массива

cout << "Введите размер квадратной матрицы: "; cin >> n;

m = 2 \* n;

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

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

A[i][j] = -10 + rand() % 21;

}

}

cout << "\nИсходный массив:" << endl; //вывод на экран исходного массива

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

cout << "\n";

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

cout << "A[" << i << "," << j << "] = " << A[i][j] << "\t";

}

}

for (i = 0; i < n; i++) { //замена элементов левого верхнего блока с правым нижним

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

temp = A[i][j];

A[i][j] = A[i + n][j + n];

A[i + n][j + n] = temp;

}

}

for (i = n; i < m ; i++) { //замена элементов правого верхнего блока с левым нижним

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

temp = A[i][j];

A[i][j] = A[i - n][j + n];

A[i - n][j + n] = temp;

}

}

cout << "\nПреобразованный массив:" << endl; //вывод на экран преобразованного массива

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

cout << "\n";

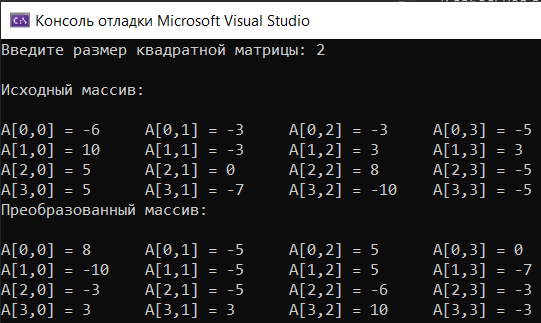
for (j = 0; j < m; j++) {

cout << "A[" << i << "," << j << "] = " << A[i][j] << "\t";

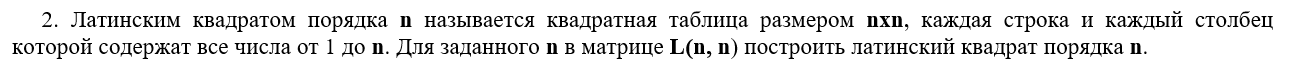
}

}

}



Вариант 2



#include <iostream>

#include <ctime> //для работы с функцией time

void main()

{

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

using namespace std;

int L[100][100]{}; //инициализация массива

int i, j, n, temp = 1; //ввод нумерации строки, столбца массива; временной переменной для расстановки элементов массива

cout << "Введите размер квадратной матрицы: "; cin >> n;

for (i = 0; i < n; i++) { //заполнение латинской матрицы

temp = i + 1;

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

L[i][j] = temp;

temp++;

if (temp > n ) { //если временная переменная выходит за границы размера матрицы, то начать отсчет сначала

temp = 1;

}

}

}

cout << "\nПреобразованный массив:" << endl; //вывод на экран преобразованного массива

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

cout << "\n";

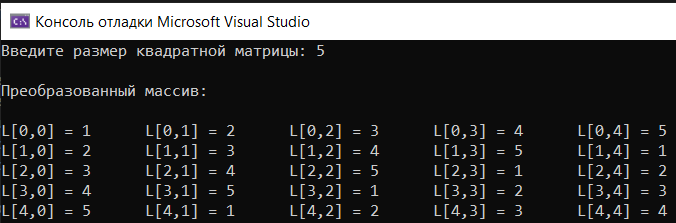
for (j = 0; j < n; j++) {

cout << "L[" << i << "," << j << "] = " << L[i][j] << "\t";

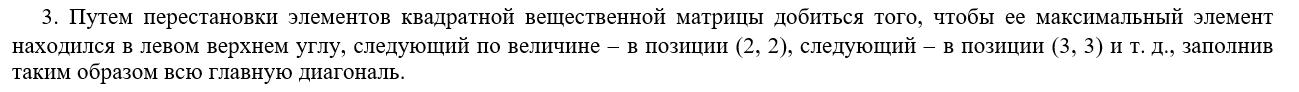
}

}

}



Вариант 3



#include <iostream>

#include <ctime> //для работы с функцией time

#include <algorithm> //для сортировки массивов

void main()

{

setlocale(LC\_CTYPE, "Russian");

srand(time(NULL)); //устанавливает новую точку отсчета для функции rand()

using namespace std;

float A[100][100]{}, B[100]{}; //инициализация массива

int i, j, n, temp = 0; //ввод нумерации строки, столбца массива; размера массива

cout << "Введите размер строки: "; cin >> n;

for (i = 0; i < n; i++) { //заполнение массива

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

A[i][j] = -10 + rand() % 21;

}

}

cout << "\nИсходный массив:" << endl; //вывод на экран исходного массива

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

cout << "\n";

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

cout << "A[" << i << "," << j << "] = " << A[i][j] << "\t";

}

}

int k = 0;

for (i = 0; i < n; i++) { //заполняем буферный массив элементами основного массива

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

B[k] = A[i][j];

k++;

}

}

sort(B, B + k, greater<>()); //сортируем буферный массив по возрастанию

k = 0;

j = 0;

for (i = 0; i < n; i++) { //заполняем буферный массив элементами основного массива

A[i][j] = B[k];

k++;

j++;

}

cout << "\nПреобразованный массив:" << endl; //вывод на экран преобразованного массива

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

cout << "\n";

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

cout << "A[" << i << "," << j << "] = " << A[i][j] << "\t";

}

}

}

