Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 17

По дисциплине “Основы алгоритмизации и программирования”

На тему “ **Указатели и ссылки при работе с функциями** ”

Выполнил:

Студент 1 курса 6 группы

Кравченко Сергей Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 8

|  |  |
| --- | --- |
| 8 | 1. В массиве вещественных чисел заменить отрицательные элементы их квадратами. Найти в полученном массиве индекс элемента, для которого сумма элементов, стоящих до него, наименее отличается от суммы элементов, стоящих после него.  2. Проверить, есть ли в матрице хотя бы один столбец, содержащий отрицательный элемент, и найти его номер. Уменьшить элементы найденного столбца вдвое. |

1.

#include <iostream>

using namespace std;

// Функция для замены отрицательных элементов массива и поиска индекса элемента с минимальной разницей сумм до и после него

float\* processArray(float\* arr, int size) {

// Замена отрицательных элементов их квадратами

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

if (arr[i] < 0) {

arr[i] = arr[i] \* arr[i];

}

}

setlocale(LC\_ALL, "RUS");

// Поиск индекса элемента с минимальной разницей сумм до и после него

float minDiff = FLT\_MAX;

int minDiffIndex = -1;

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

float sumBefore = 0;

float sumAfter = 0;

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

sumBefore += arr[j];

}

for (int j = i + 1; j < size; j++) {

sumAfter += arr[j];

}

float diff = abs(sumBefore - sumAfter);

if (diff < minDiff) {

minDiff = diff;

minDiffIndex = i;

}

}

// Вывод результата

cout << "Индекс элемента с минимальной разницей сумм до и после него: " << minDiffIndex << endl;

return arr;

}

int main() {

setlocale(LC\_ALL, "RUS");

// Ввод исходных данных

int size;

cout << "Введите размер массива: ";

cin >> size;

float\* arr = new float[size];

cout << "Введите элементы массива:" << endl;

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

cin >> arr[i];

}

// Вызов функции для обработки массива

\*processArray(arr, size);

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

cout << arr[i] << endl;

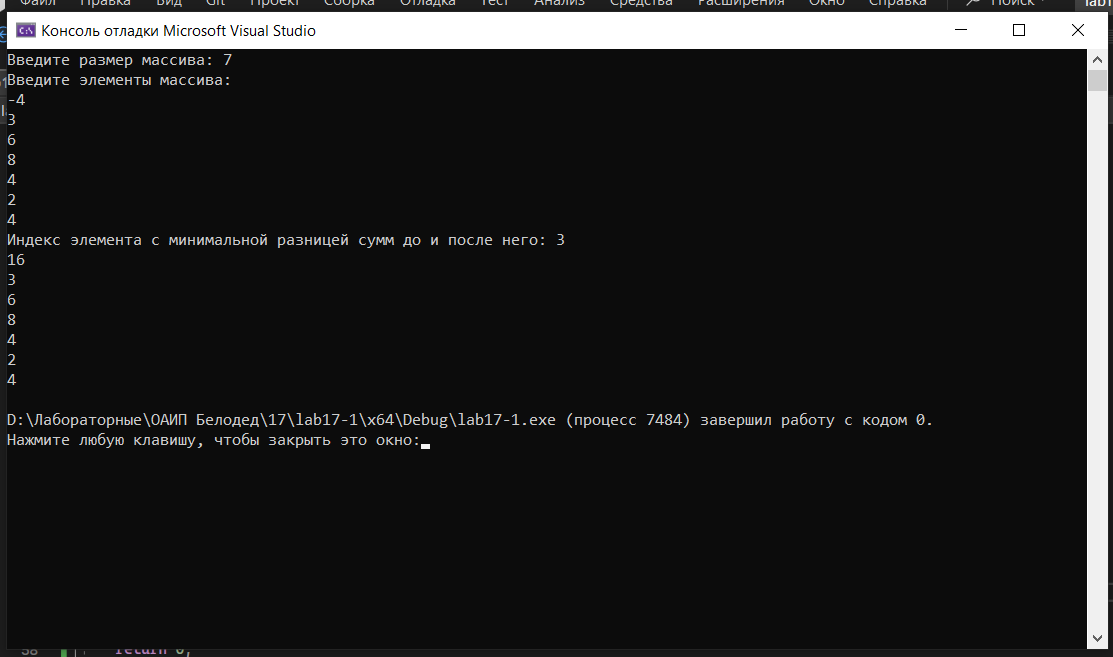
}

// Освобождение памяти

delete[] arr;

return 0;

}



2.

#include <iostream>

using namespace std;

// Функция для создания и заполнения матрицы

float\*\* creatematrix(int m, int n) {

float\*\* arr = new float\* [m];

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

arr[i] = new float[n];

}

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

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

cout << "Введите элемент [" << i + 1 << "][" << j + 1 << "]: ";

cin >> arr[i][j];

}

}

return arr;

}

// Функция для обработки матрицы

float\*\* processMatrix(float\*\* arr, int m, int n) {

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

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

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

cout << "Найдена строка с отрицательным элементом: " << i + 1 << endl;

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

arr[k][j] /= 2;

}

}

}

}

return arr;

}

int main() {

setlocale(LC\_ALL, "RUS");

int m, n;

cout << "Введите количество строк в матрице: ";

cin >> m;

cout << "Введите количество столбцов в матрице: ";

cin >> n;

float\*\* arr = creatematrix(m, n);

cout << "Первый массив: " << endl;

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

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

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

}

cout << endl;

}

\*processMatrix(arr, m, n);

cout << "Новый массив: " << endl;

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

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

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

}

cout << endl;

}

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

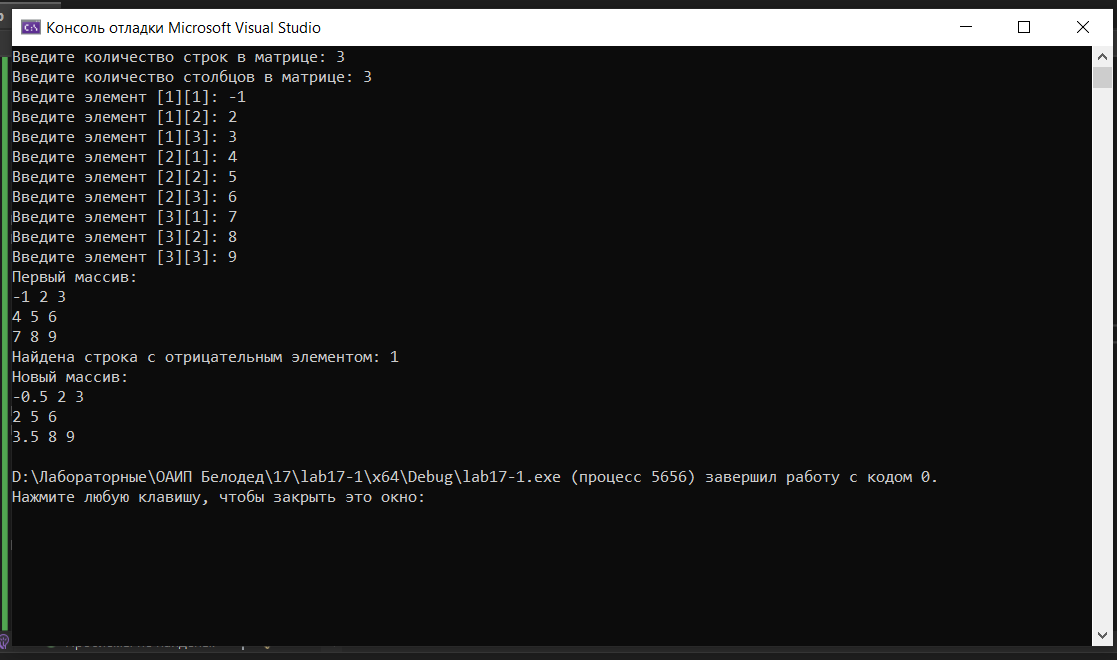
delete[] arr[i];

}

delete[] arr;

return 0;

}



Дополнительные варианты

|  |  |
| --- | --- |
| 12 | 1. Найти количество отрицательных элементов, стоящих на чётных местах в одномерном массиве.  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>

using namespace std;

int find(int\* a, int n) { // функция подсчёта количества отрицательных элементов, которые стоят на чётных местах

int s = 0;

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

if ((i + 1) % 2 == 0 and a[i] < 0) {

s++;

}

}

return s;

}

int main() {

setlocale(LC\_ALL, "RUS");

int n;

cout << "Введите размер массива ";

cin >> n;

int\* a = new int[n];

cout << "Введите элементы массива " << endl;

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

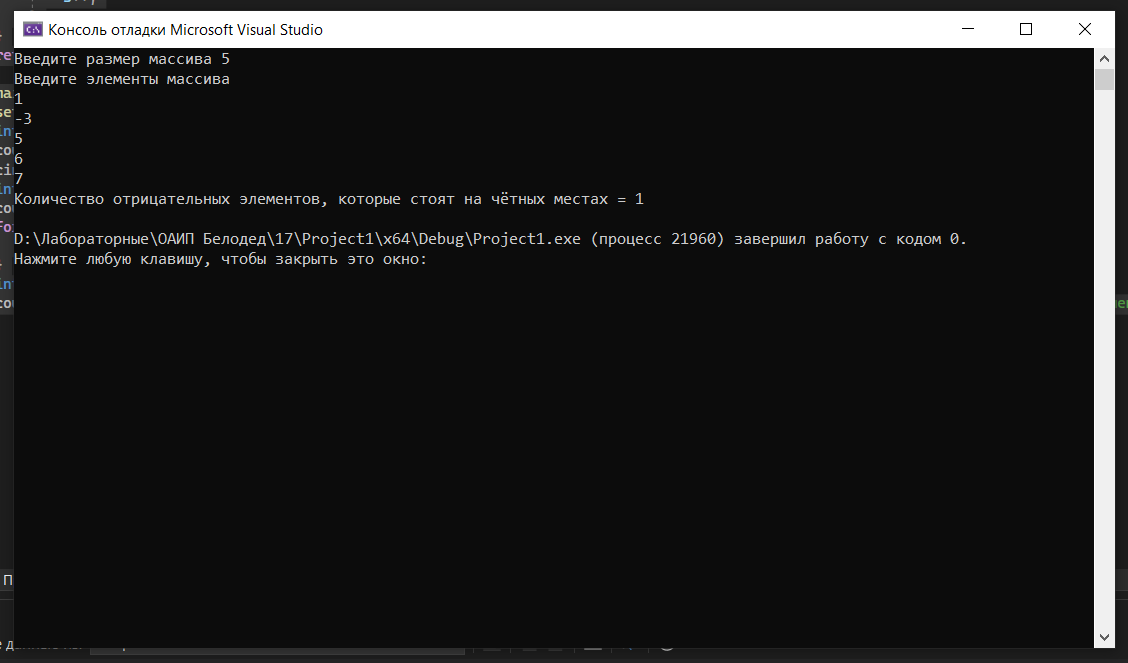
cin >> a[i];

}

int amount = find(a, n); // вызываем функцию подсчёта количества отрицательных элементов, которые стоят на чётных местах

cout << "Количество отрицательных элементов, которые стоят на чётных местах = " << amount << endl; // выводим количество таких элементов

}



2.

#include <iostream>

using namespace std;

double\*\* change(double\*\* a, int n, int m) { // функция поиска индекса первой строки с отрицательным элементом и уменьшения значения элементов столбца с таким же индексом, как и эта строка, в 2 раза

int s = 0;

bool t = false;

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

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

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

cout << "Первая строка, которая содержит отрицательный элемент имеет индекс " << i << endl;

s = i;

t = true;

break;

}

}

if (t == true) {

break;

}

}

if (t == true) {

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

a[i][s] /= 2;

}

}

else {

cout << "Ни одна из строк не содержит отрицательные элементы" << endl;

}

return a;

}

int main() { // функция main

setlocale(LC\_ALL, "RUS");

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m;

double\*\* a = new double\* [n];

cout << "Введите элементы матрицы " << endl;

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

a[i] = new double[m];

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

cin >> a[i][j];

}

}

cout << endl;

change(a, n, m); // вызываем функцию

for (int i = 0; i < n; i++) { // выводим полученную матрицу

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

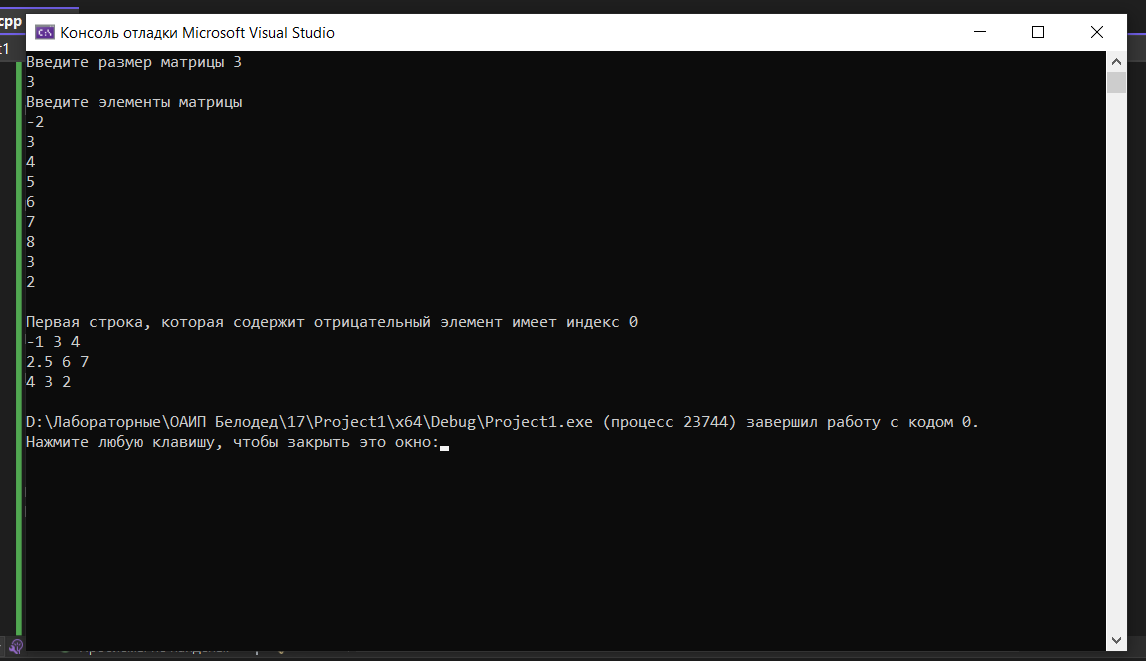
cout << a[i][j] << ' ';

}

cout << endl;

}

}



|  |  |
| --- | --- |
| 4 | 1. Создать одномерный массив, содержащий 15 элементов, наполнить его случайными значениями в интервале от 1 до 200. Определить сумму всех нечетных элементов массива.  2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные. |

1.

#include <iostream>

using namespace std;

// Функция для вычисления суммы нечетных элементов массива

void sum(int\* arr, int n) {

int sum = 0;

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

if (arr[i] % 2 != 0)

sum += arr[i];

}

cout << "Сумма нечетных элементов массива - " << sum; // Вывод суммы нечетных элементов

}

int main() {

srand(time(NULL));

setlocale(LC\_CTYPE, "Rus");

int n = 15;

int\* mass = new int[n]; // Выделение памяти под массив

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

mass[i] = rand() % 2; // Заполнение массива случайными числами

cout << mass[i] << " ";

}

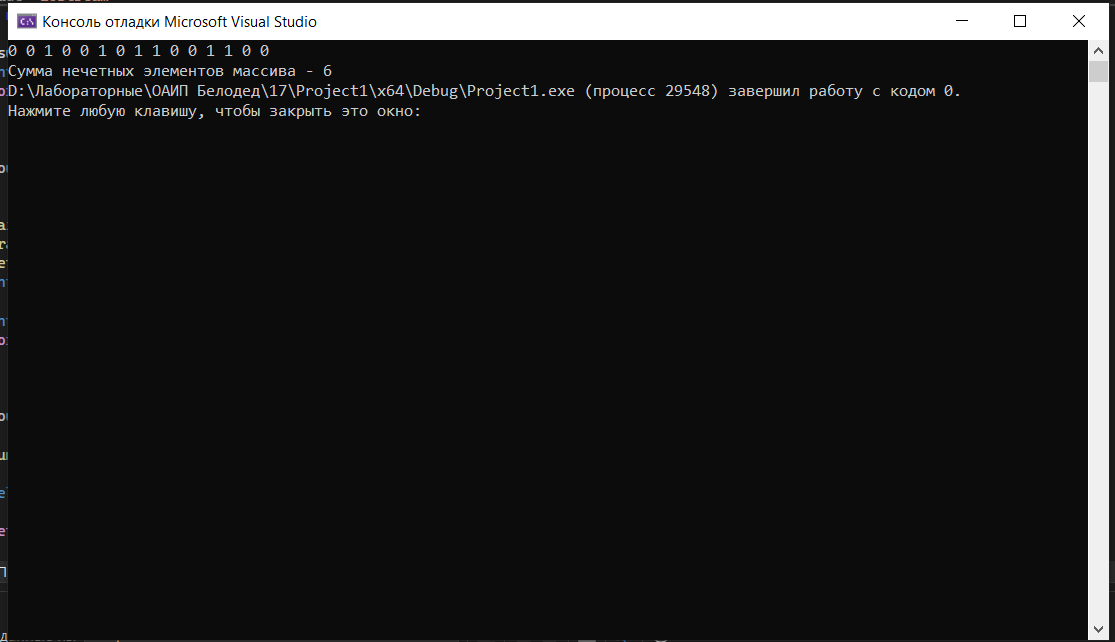
cout << endl;

sum(mass, n); // Вызов функции для вычисления суммы нечетных элементов

delete[] mass; // Освобождение памяти, выделенной под массив

return 0;

}



2.

#include <iostream>

using namespace std;

// Функция для создания и заполнения матрицы

double\*\* createAndFillMatrix(int n, int m) {

double\*\* a = new double\* [n];

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

a[i] = new double[m];

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

cin >> a[i][j];

}

}

return a;

}

// Функция для обработки матрицы

double\*\* change(double\*\* a, int n, int m) {

int k = 0;

bool t = false;

// Проверяем каждую строку матрицы на наличие положительного элемента

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

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

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

cout << "Первая строка, которая содержит положительный элемент :" << i + 1 << endl;

k = i;

t = true;

break;

}

}

if (t == true) {

break;

}

}

// Если была найдена строка с положительным элементом и она не является первой строкой

if (t == true && k != 0) {

// Меняем знак элементов первой строки на обратный

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

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

a[k - 1][j] \*= -1;

}

}

// Если ни одна из строк не содержит положительных элементов

else {

cout << "Ни одна из строк не содержит положительные элементы" << endl;

}

return a;

}

int main() {

setlocale(LC\_CTYPE, "RUS");

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m;

double\*\* a = createAndFillMatrix(n, m);

change(a, n, m);

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

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

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

}

cout << endl;

}

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

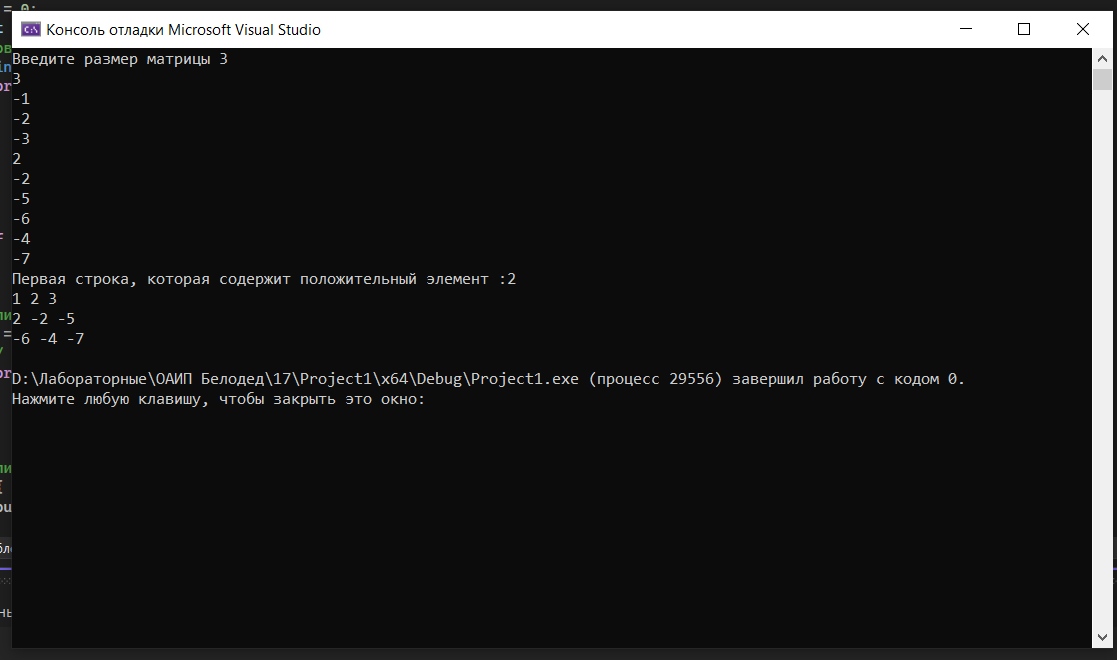
delete[] a[i];

}

delete[] a;

return 0;

}



|  |  |
| --- | --- |
| 16 | 1. В одномерном массиве найти максимальный из отрицательных элементов и поменять его местами с последним элементом массива.  2. Проверить, все ли строки матрицы содержат хотя бы один отрицательный элемент. Если да, то изменить знаки всех элементов матрицы на обратные. |

1.

#include <iostream>

using namespace std;

int\* poisk\_and\_zamena(int\* a, int n) { // функция, которая ищет максимальный отрицательный элемент и меняет его местами с последним элементом

int max = 0;

int position = -1;

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

if (a[i] < 0 and max == 0) {

max = a[i];

position = i;

}

else if (a[i] < 0 and a[i] > max) { //переприсваиваем значения

max = a[i];

position = i;

}

}

if (max == 0) {

cout << "В массиве нет отрицательных элементов " << endl;

return a;

}

int l = a[position];

a[position] = a[n - 1];

a[n - 1] = l;

return a;

}

int main() { // функция main

setlocale(LC\_ALL, "RUS");

int n;

cout << "Введите размер массива ";

cin >> n; // считываем размер массива

int\* a = new int[n];

cout << "Введите элементы массива " << endl;

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

cin >> a[i];

}

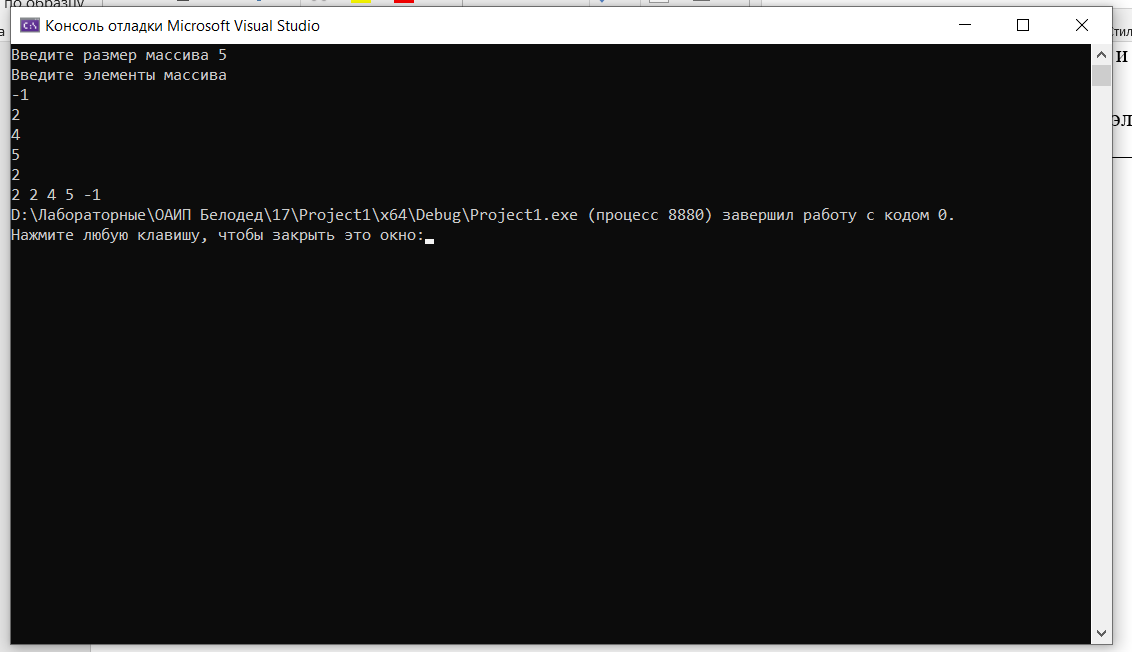
poisk\_and\_zamena(a, n); // вызываем функцию

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

cout << a[i] << ' ';

}

}



2.

#include <iostream>

using namespace std;

int\*\* search\_and\_change(int\*\* a, int n, int m) { // функция, которая проверяет наличие во всех строках хотя бы 1 отрицательного элемента, и, если во всех строках есть такой, меняет знаки всех элементов на противоположные

int s = 0;

for (int i = 0; i < n; i++) { // проходимся по матрице

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

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

s++;

break;

}

}

}

if (s == n) { // проверка условия

cout << "В каждой строке есть хотя бы 1 отрицательный элемент " << endl;

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

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

a[i][j] = -a[i][j];

}

}

}

else {

cout << "Не в каждой строке есть отрицательный элемент" << endl;

}

return a;

}

int main() { // основная функция

setlocale(LC\_ALL, "RUS");

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m; // считываем размер матрицы

int\*\* a = new int\* [n];

cout << "Введите элементы матрицы " << endl;

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

a[i] = new int[m];

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

cin >> a[i][j];

}

}

search\_and\_change(a, n, m); // вызываем функцию

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

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

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

}

cout << endl;

}

}

