Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 15

По дисциплине “Основы алгоритмизации и программирования”

На тему “ **Динамическое выделение памяти**”

Выполнил:

Студент 1 курса 6 группы

Кравченко Сергей Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 8

|  |  |
| --- | --- |
| **8** | 1. В одномерном массиве, состоящем из **k** целых элементов, вычислить количество положительных элементов массива и сумму элементов массива, расположенных после последнего элемента, равного нулю.  2. Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент. |

1.

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

int\* ptr, i, n; // Переменные

int amount=0;

int sum=0;

cout << "Введите количество элементов в массиве";

cin >> n;

if (!(ptr = (int\*)malloc(n \* sizeof(int)))) //выделение памяти и проверка,

{ //достаточно ли для нее места

cout << "Недостаточно памяти";

return 1;

}

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

{

cout << "Введите элементы [" << i + 1 << "]\n";

cin >> \*(ptr + i);

}

cout << "Массив: " << endl; // Вводим массив

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

cout << \*(ptr + i) << " ";

}

for (i = 0; i < n; i++) { // Ищем количество элементов больше нуля и их сумму

if (\*(ptr + i) > 0) {

amount++;

sum += \*(ptr + i);

}

}

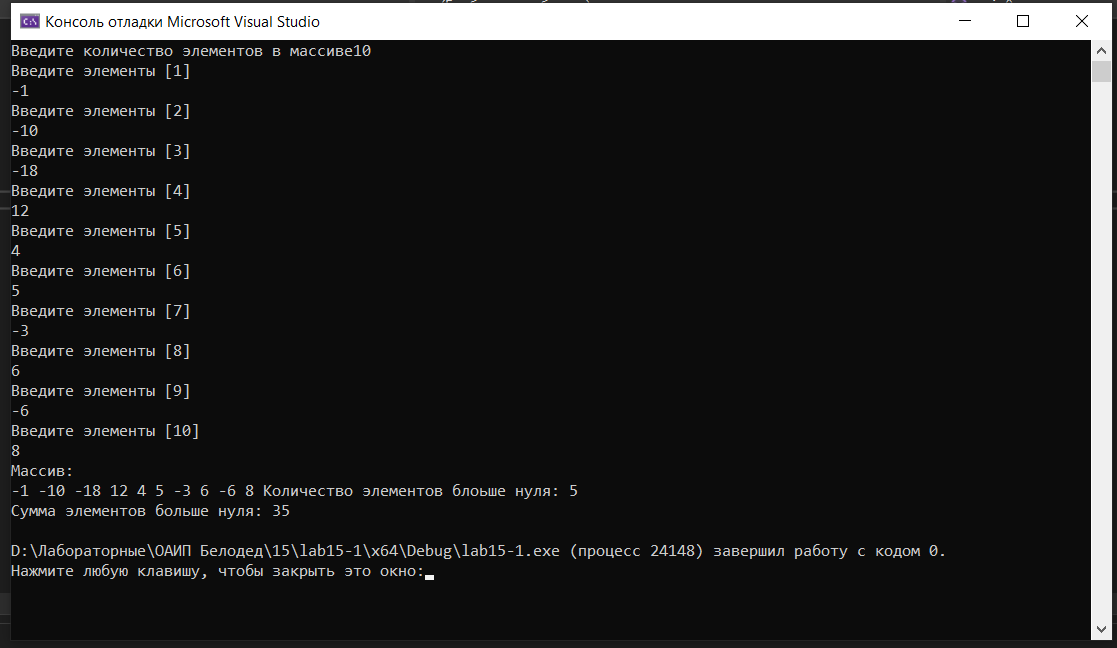
cout << "Количество элементов блоьше нуля: " << amount << endl;

cout << "Сумма элементов больше нуля: " << sum << endl;

free(ptr); //освобождение динамической памяти

return 0;

}



2.

#include <iostream>

#include <ctime>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int m, n;

cout << "Введите количество строк: ";

cin >> m;

cout << "Введите количество столбцов: ";

cin >> n;

int count = 0;

int\*\* mass = new int\* [n]; //Эта строка выделяет память под указатели на строки двумерного массива.

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

if (!(mass[i] = new int[m])) { // Эта строка выделяет память под строку двумерного массива

cout << "Недостаточно памяти" << endl;

return 1;

}

}

srand((unsigned)time(NULL)); // Эта строка инициализирует генератор случайных чисел

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

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

mass[i][j] = rand() % 10; // Заполняет каждый элемент двумерного массива случайным числом

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

}

cout << endl;

}

int s = 0;

for (int j = 0; j < n; j++) { // Цикл проходит по каждому столбцу двумерного массива.

bool nol = false;

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

if (mass[i][j] == 0) { // Эта строка проверяет, равно ли текущее значение нулю

nol = true;

s = i;

break;

}

}

if (nol) { // Строка проверяет, был ли найден нуль в столбце

count++;

}

}

cout << "Номер первого столбца с нулем: " << s << endl;

cout << "Количество столбцов c 0: " << count << endl;

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

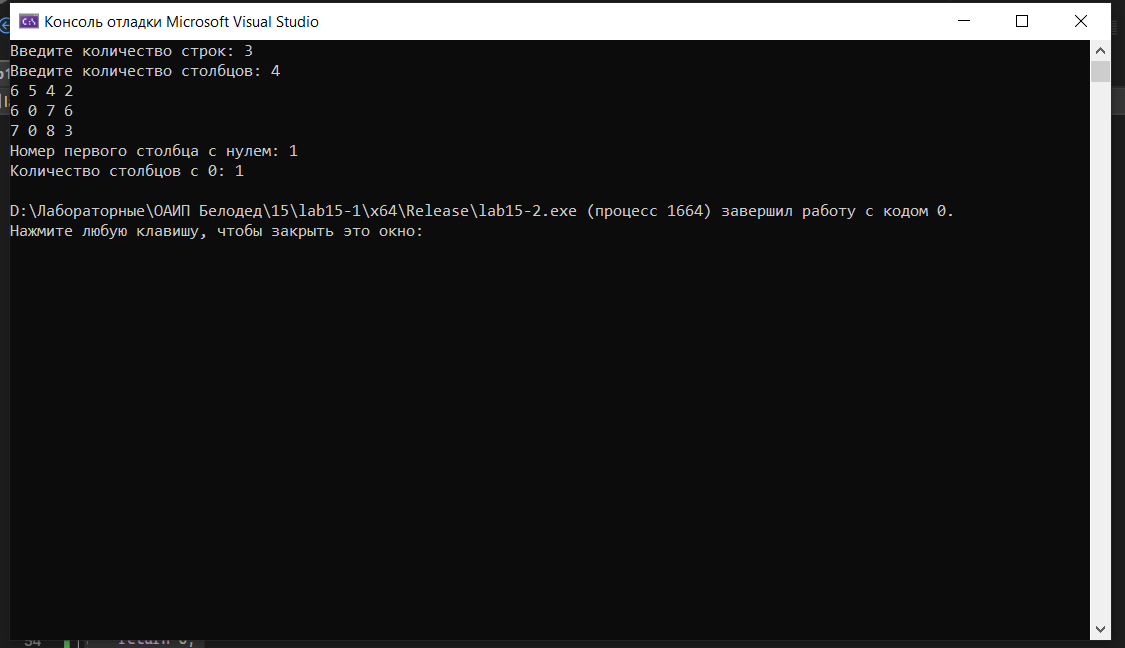
delete[] mass[i]; // Эта строка освобождает память, выделенную под строку двумерного массива

}

delete[] mass;

return 0;

}



Доп задания

Вариант 6

|  |  |
| --- | --- |
| **6** | 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить минимальный элемент массива и сумму элементов, расположенных между первым и последним положительными элементами.  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 <climits>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

int n;

cout << "Введите количество элементов в массиве: ";

cin >> n;

int\* arr; // Объявление указателя на массив

float min = FLT\_MAX; // Инициализация минимального значения максимальным возможным значением

int firstPos = -1;

int lastPos = -1;

// Выделение памяти под массив

if (!(arr = (int\*)new int[n])) {

cout << "Not enough memory" << endl; // Вывод сообщения об ошибке, если память не выделена

return 1; // Возврат из программы с кодом ошибки

}

// Заполнение массива и поиск минимального элемента и индексов первого и последнего положительных элементов

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

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

cin >> \*(arr + i);

if (\*(arr + i) < min) {

min = \*(arr + i);

}

if (\*(arr + i) > 0 && firstPos == -1) {

firstPos = i;

}

if (\*(arr + i) > 0) {

lastPos = i;

}

}

// Вычисление суммы элементов между первым и последним положительными элементами

float sum = 0;

for (int i = firstPos; i <= lastPos; i++) {

sum += \*(arr + i);

}

// Вывод минимального элемента массива и суммы элементов между первым и последним положительными элементами

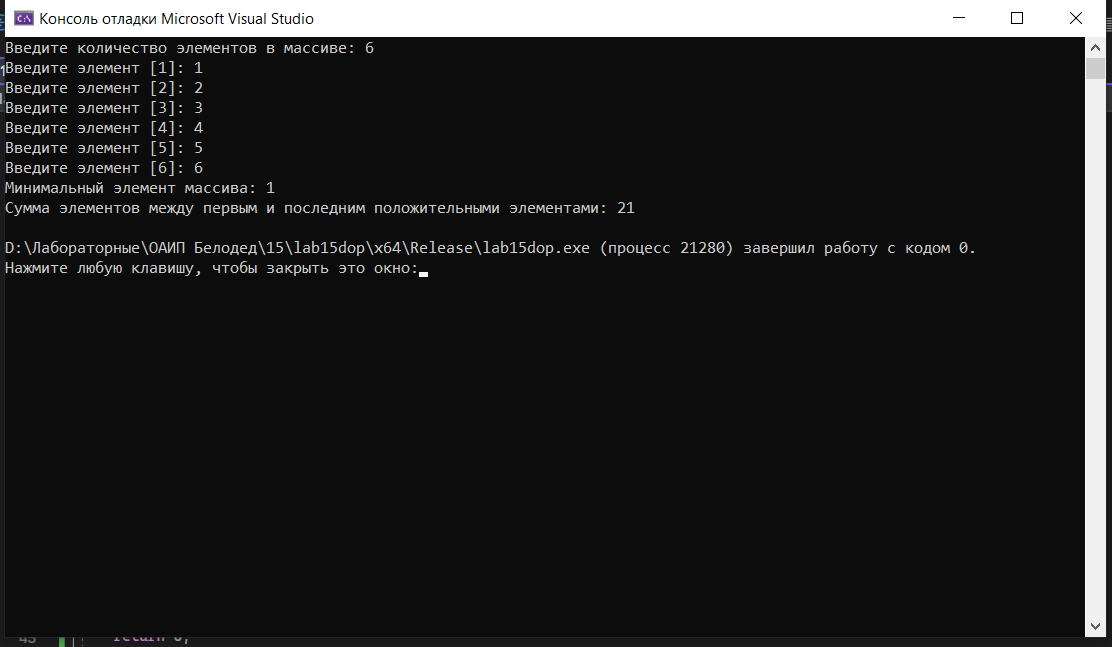
cout << "Минимальный элемент массива: " << min << endl;

cout << "Сумма элементов между первым и последним положительными элементами: " << sum << endl;

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

return 0;

}



2.

#include <iostream>

using namespace std;

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 = new float\* [m]; // Создаем динамический массив

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

arr[i] = new float[n];

}

for (int i = 0; i < m; i++) { // Вводим элементы массива

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

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

cin >> arr[i][j];

}

}

cout << "Первый массив: " << endl; // Выводим первый массив

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++) {

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

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

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

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

arr[k][j] /= 2;

}

}

}

}

cout << "Новый массив: " << endl; // Выводим новый массив

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++) { // Очищаем память

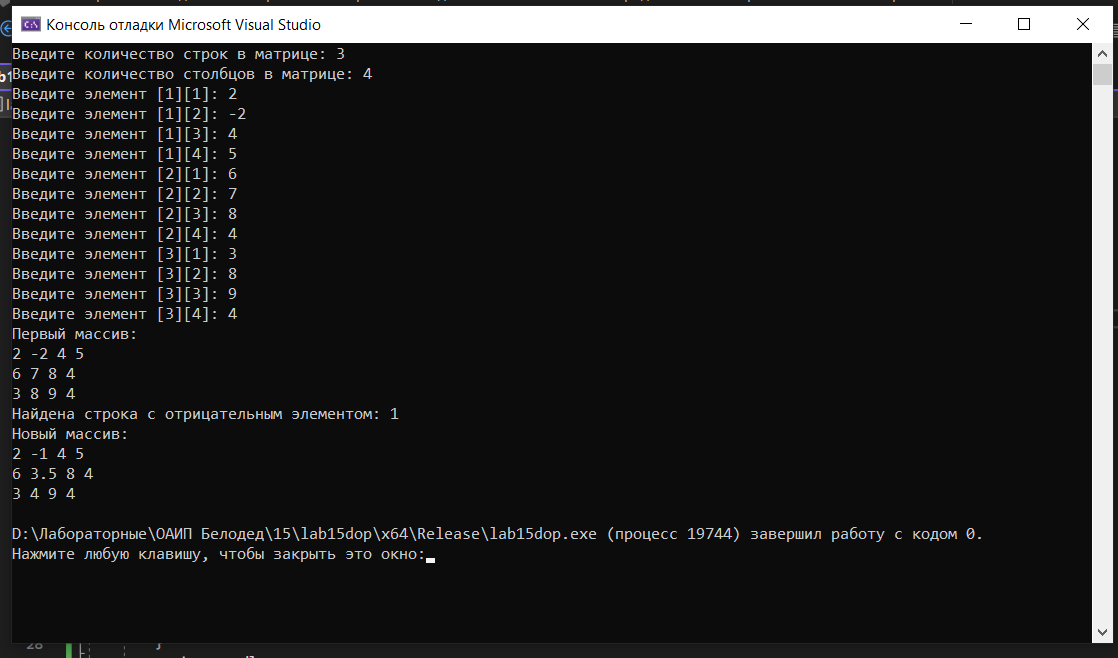
delete[] arr[i];

}

delete[] arr;

return 0;

}



Вариант 5

|  |  |
| --- | --- |
| **5** | 1. В одномерном массиве, состоящем из **n** вещественных элементов, вычислить количество элементов массива, равных 0, и сумму элементов массива, расположённых после минимального элемента.  2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, содержащих хотя бы один нулевой элемент и номер строки, в которой находится самая длинная серия одинаковых элементов. |

1.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

unsigned int n;

cout << "Введите размер будущего массива: ";

cin >> n;

double\* arr = new double[n]; // создание динамического массива

// заполнение массива случайными числами

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

\*(arr+i) = -10 + (double)rand() / RAND\_MAX \* 20;

}

// вывод массива на экран

cout << "Массив: ";

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

cout << \*(arr + i) << " ";

}

cout << endl;

// поиск минимального элемента

double min = arr[0];

int minIndex = 0;

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

if (\*(arr + i) < min) {

min = \*(arr + i);

minIndex = i;

}

}

// подсчет количества элементов, равных 0

int zeroCount = 0;

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

if (\*(arr + i) == 0) {

zeroCount++;

}

}

// суммирование элементов после минимального

double sum = 0;

for (int i = minIndex + 1; i < n; i++) {

sum += \*(arr + i);

}

// вывод результатов

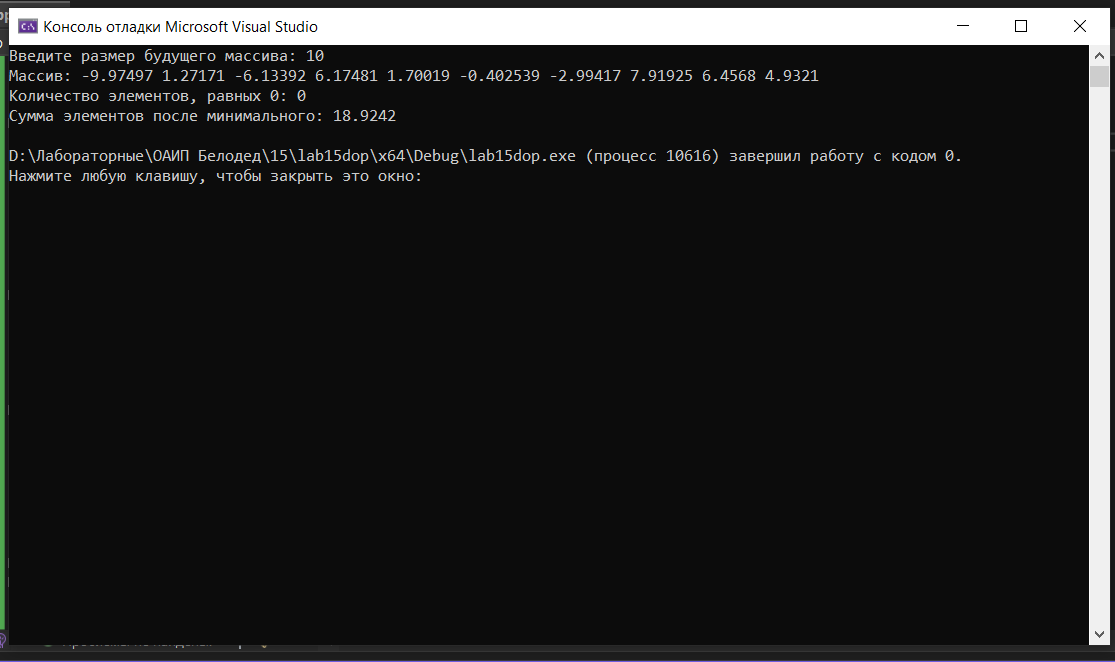
cout << "Количество элементов, равных 0: " << zeroCount << endl;

cout << "Сумма элементов после минимального: " << sum << endl;

delete[] arr; // освобождение памяти, выделенной для массива

return 0;

}



2.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

int rows, cols;

cout << "Введите количество строк: ";

cin >> rows;

cout << "Введите количество столбцов: ";

cin >> cols;

int\*\* arr = new int\* [rows]; // создание динамического двумерного массива

// заполнение матрицы

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

arr[i] = new int[cols];

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

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

cin >> arr[i][j];

}

}

int zeroCols = 0; // количество столбцов, содержащих хотя бы один нулевой элемент

int longestSeriesRow = 0; // номер строки, в которой находится самая длинная серия одинаковых элементов

int longestSeriesLength = 0; // длина самой длинной серии одинаковых элементов

// поиск столбцов с нулевыми элементами

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

bool hasZero = false;

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

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

hasZero = true;

break;

}

}

if (hasZero) {

zeroCols++;

}

}

// поиск строки с самой длинной серией одинаковых элементов

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

int currentSeriesLength = 1;

for (int j = 1; j < cols; j++) {

if (arr[i][j] == arr[i][j - 1]) {

currentSeriesLength++;

}

else {

if (currentSeriesLength > longestSeriesLength) {

longestSeriesLength = currentSeriesLength;

longestSeriesRow = i;

}

currentSeriesLength = 1;

}

}

if (currentSeriesLength > longestSeriesLength) {

longestSeriesLength = currentSeriesLength;

longestSeriesRow = i;

}

}

// вывод результатов

cout << "Количество столбцов, содержащих хотя бы один нулевой элемент: " << zeroCols << endl;

cout << "Номер строки, в которой находится самая длинная серия одинаковых элементов: " << longestSeriesRow << endl;

// освобождение памяти, выделенной для массива

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

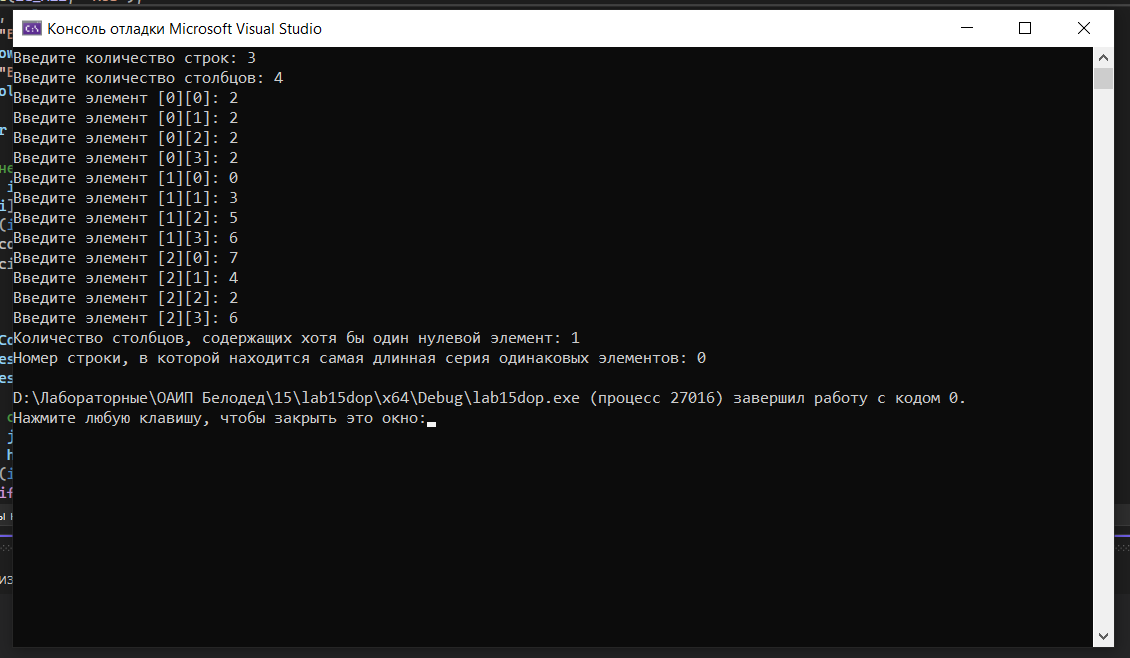
delete[] arr[i];

}

delete[] arr;

return 0;

}



Вариант 4

|  |  |
| --- | --- |
| **4** | 1. В одномерном массиве, состоящем из **n** вещественных элементов, вычислить номер минимального элемента массива и сумму элементов массива, расположенных между первым и вторым отрицательными элементами.  2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента. |

1.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

int n;

cout << "Введите количество элементов в массиве: ";

cin >> n;

double\* arr = new double[n]; // создание динамического массива

// заполнение массива

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

cout << "Введите элемент массива: ";

cin >> \*(arr + i);

}

int minIndex = 0; // индекс минимального элемента

double min = arr[0]; // минимальный элемент

bool foundFirstNegative = false; // флаг для отслеживания первого отрицательного элемента

double sum = 0; // сумма элементов между первым и вторым отрицательными элементами

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

// поиск минимального элемента

if (\*(arr+i) < min) {

min = \*(arr + i);

minIndex = i;

}

// вычисление суммы элементов между первым и вторым отрицательными элементами

if (\*(arr + i) < 0) {

if (foundFirstNegative) {

break; // выходим из цикла, если уже найден первый отрицательный элемент

}

else {

foundFirstNegative = true;

}

}

else if (foundFirstNegative) {

sum += \*(arr + i);

}

}

// вывод результатов

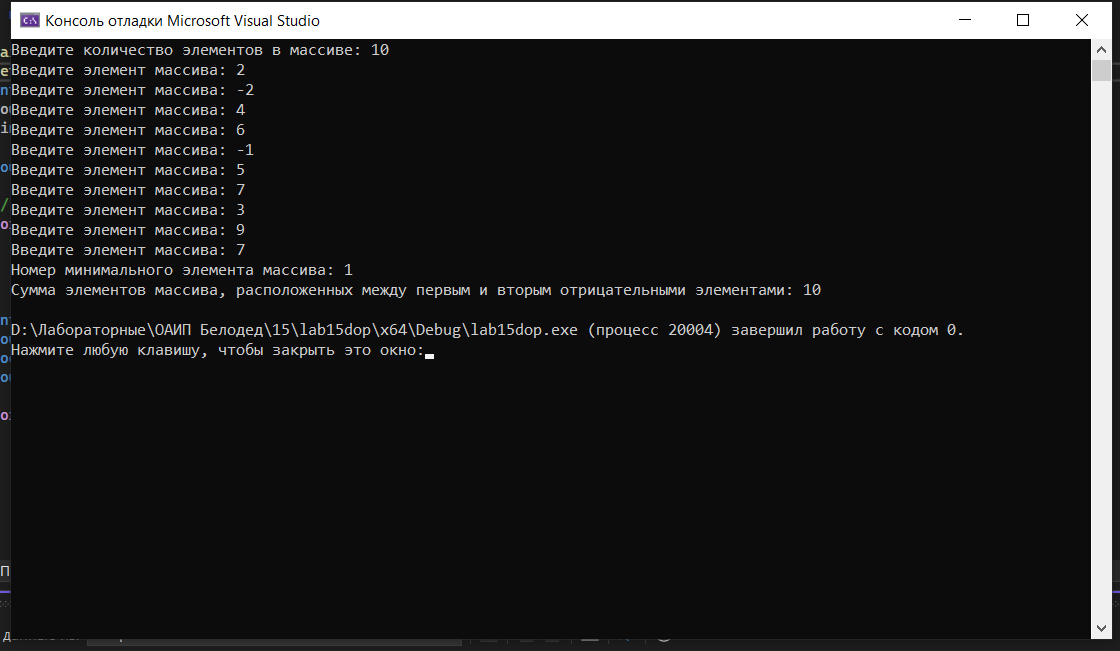
cout << "Номер минимального элемента массива: " << minIndex << endl;

cout << "Сумма элементов массива, расположенных между первым и вторым отрицательными элементами: " << sum << endl;

delete[] arr; // освобождение памяти, выделенной для массива

return 0;

}



2.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int m, n;

cout << "Введите количество строк: ";

cin >> m;

cout << "Введите количество столбцов: ";

cin >> n;

int count = 0; // Счетчик столбцов без нулей

// Создание динамического двумерного массива

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

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

mass[i] = new int[m];

}

srand((unsigned)time(NULL)); // Инициализация генератора случайных чисел

// Заполнение матрицы случайными числами и вывод их на экран

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

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

mass[i][j] = rand() % 10;

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

}

cout << endl;

}

// Поиск столбцов без нулей

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

bool nol = false;

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

if (mass[i][j] == 0) {

nol = true;

break;

}

}

if (!nol) {

count++;

}

}

// Вывод количества столбцов без нулей

cout << "Количество столбцов без 0: " << count << endl;

// Освобождение памяти, выделенной под матрицу

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

delete[] mass[i];

}

delete[] mass;

return 0;

}

