Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 15

По дисциплине «Основы алгоритмизации и программирования»

на тему «**Динамическое выделение памяти**»

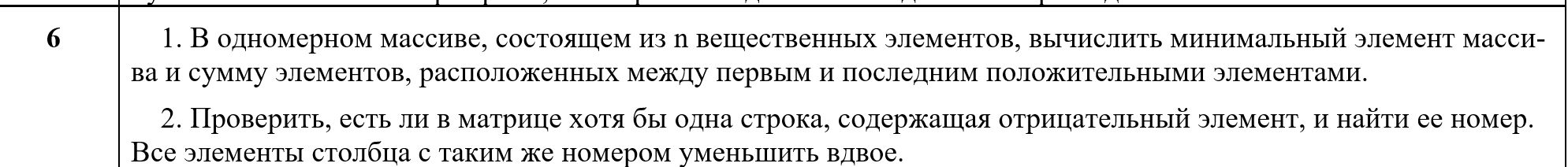
Выполнила:

Студент(ка) 1 курса 7 группы

Гриценко Анна Александровна

Преподаватель: асс. Андронова М.В.

2023, Минск



**Задание 1**

#include <iostream>//подключаем библиотеку для работы с вводом/выводом

using namespace std;//подключаем пространство имен std

int firstposelement(double\* array, int n) {//с помощью функции находим первый положительный элемент

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

if (array[i] > 0) {

return i;

}

}

int lastposelement(double\* array, int n) {//с помощью функции находим последний положительный элемент

for (int i = n; i > 0; i--) {

if (array[i] > 0) {

return i;

}

}

}

int main() {

srand(time(NULL));

setlocale(LC\_ALL, "RUS");//подключаем функцию для корректного отображения русского языка

unsigned int n;//инициализация переменной

cout << "Введите размер массива " << endl;//указываем размер массива

cin >> n;

double minelement = INT\_MAX, sum = 0; //инициализация переменных, присвоение максимального значения переменной tmp типа double

double\* array = (double\*)malloc(n \* sizeof(double));//выделяем память для динамического массива

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

array[i] = -10 + (double)rand() / RAND\_MAX \* 20; //заполнение массива случайными числами с помощью функции rand()

cout << array[i] << " "; //вывод массива в консоль

if (array[i] < minelement) {//условие для нахождения наименьшего элемента

minelement = array[i];

}

}

//с помощью цикла находим сумму элементов между первым и последним положительными элементами

for (int i = firstposelement(array, n); i < lastposelement(array, n); i++) {

sum += array[i];

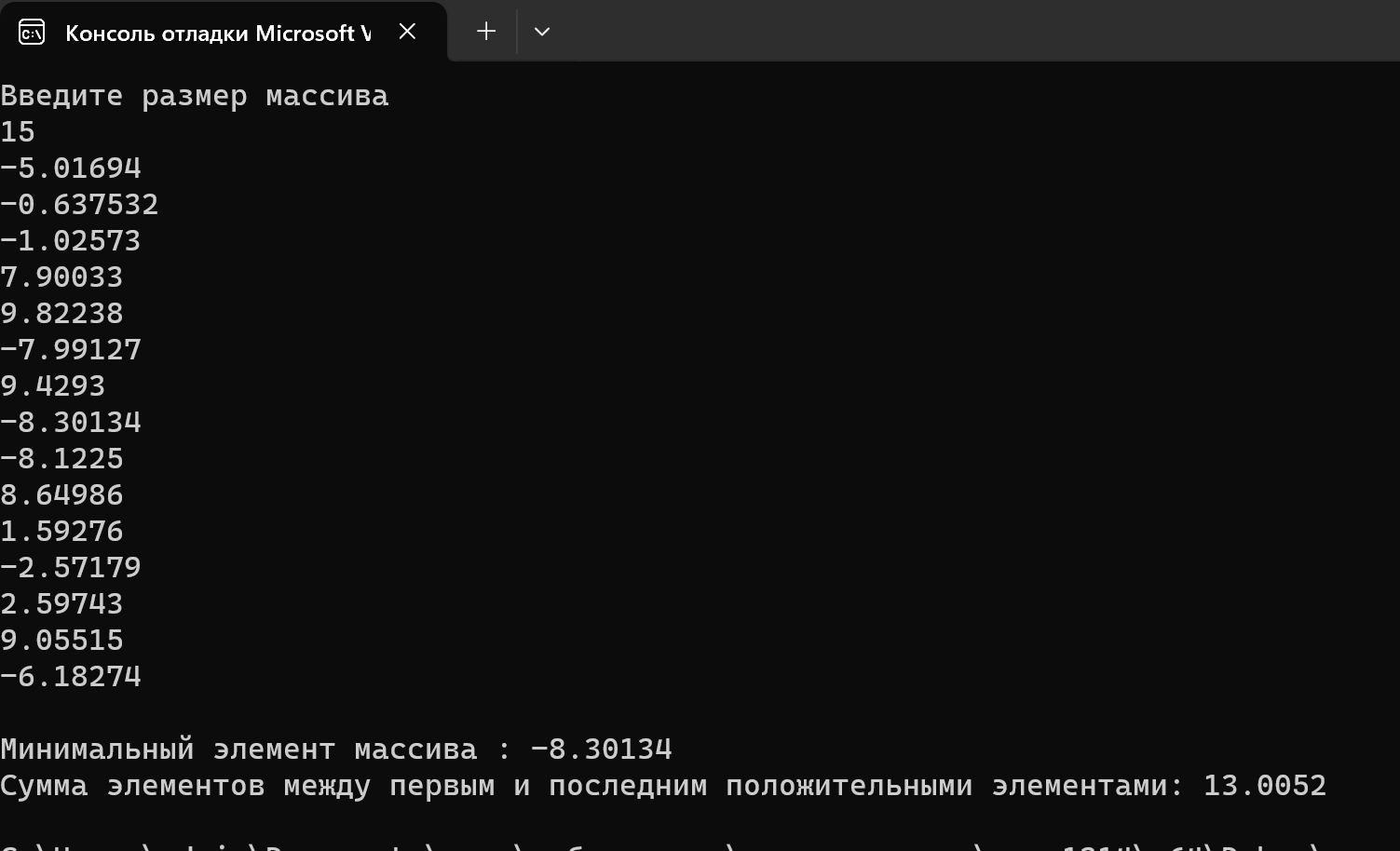
}

cout << endl;

cout << "Минимальный элемент массива : " << minelement << endl;//выводим результат в консоль

cout << "Сумма элементов между первым и последним положительными элементами: " << sum << endl;//выводим результат в консоль

}



**Задание 2**

#include <iostream>//подключаем библиотеку для работы с вводом/выводом

using namespace std;//подключаем пространство имен std

int negelement(int\*\* array, int n, int m) {//с помощью функции находим отрицательный элемент

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

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

if (array[i][j] < 0)

return i;

}

}

}

int main() {

srand(time(NULL));

setlocale(LC\_ALL, "RUS");//функция для корректного отображения русского языка

int rows, columns; //инициализируем переменные

cout << "Введите количество строк матрицы: ";//выводим сообщение на экран

cin >> rows;//вводим значение переменной с клавиатуры

cout << "Введите количество столбцов матрицы: ";//выводим сообщение на экран

cin >> columns;//вводим значение переменной с клавиатуры

int\*\* array = new int\* [rows];//создание динамического массива с помощью операции new

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

array[i] = new int[columns];

}

for (int i = 0; i < rows; i++) {//заполнение матрицы рандомными числами с помощью rand()

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

array[i][j] = rand() % (10 + 1);

}

}

//в любую ячейку матрицы подставляем отрицательный элемент

array[rand() % ((rows - 1) + 1)][rand() % ((columns - 1) + 1)] = -2;

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

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

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

}

cout << endl;

}

cout << endl;

for (int j = 0; j < rows; j++) {//уменьшаем вдвое элементы в строке с отрицательным элементом

array[negelement(array, rows, columns)][j] /= 2;

}

cout << "Измененная матрица: " << endl;

for (int i = 0; i < rows; i++) {//выводим результат измененную матрицу

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

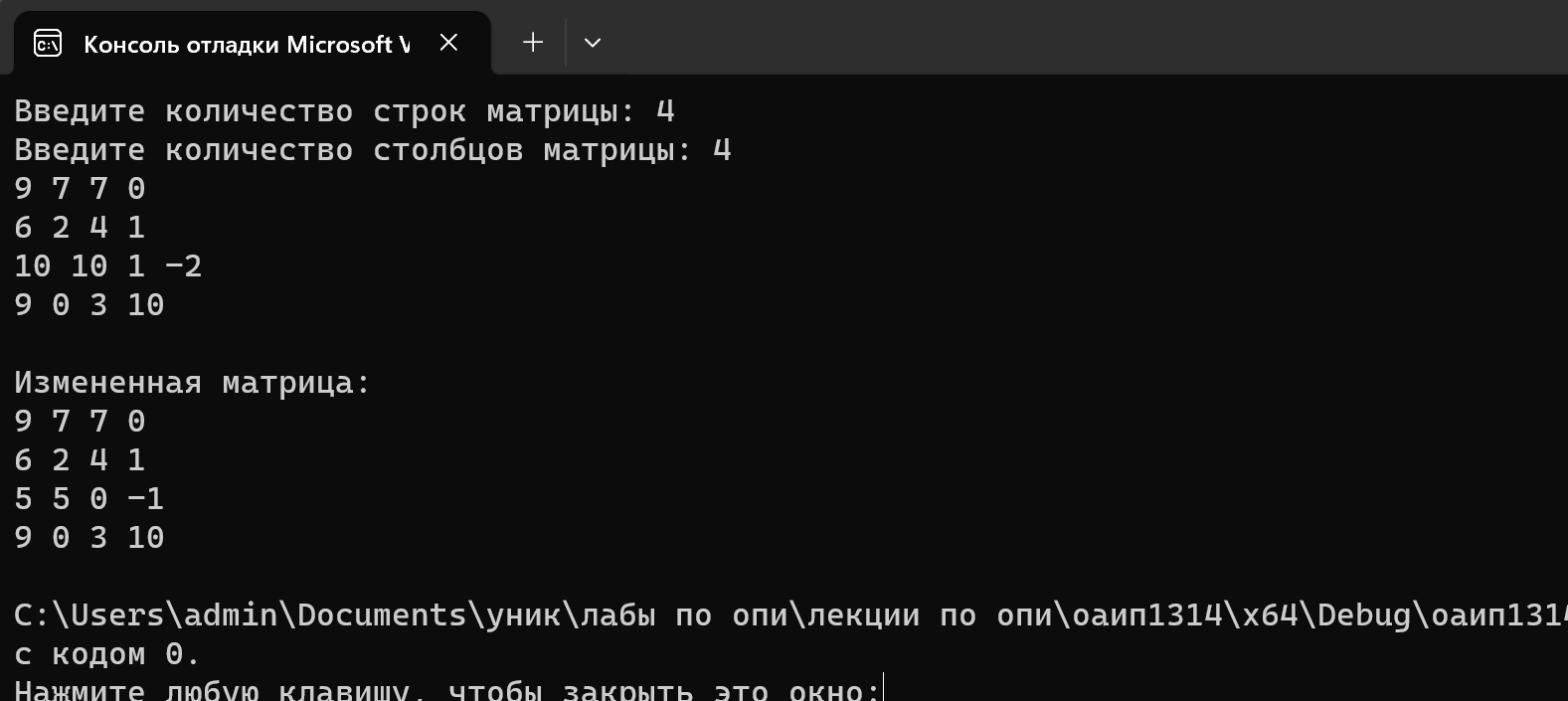
cout << array[i][j] << " ";

}

cout << endl;

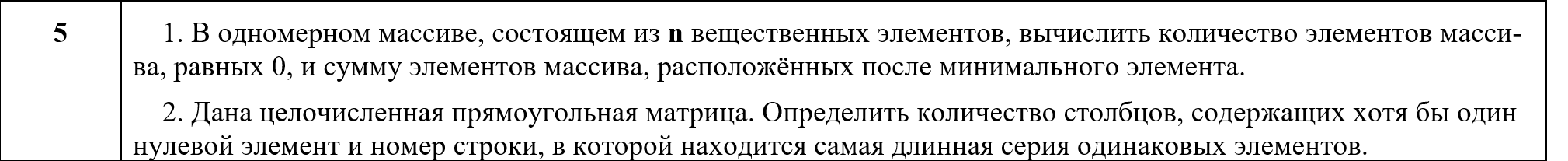
}

}



**Дополнительные задания**

**Вариант 5**

****

**Задание 1**

#include <iostream>//подключаем библиотеку для работы с вводом/выводом

using namespace std;//подключаем пространство имен std

int main() {

srand(time(NULL));

setlocale(LC\_ALL, "RUS");//подключаем функцию для корректного отображения русского языка

unsigned int n;//инициализация переменной

cout << "Введите размер массива " << endl;//указываем размер массива

cin >> n;

int minelement = 0, sum = 0; //инициализация переменных

int\* array = (int\*)malloc(n \* sizeof(int));//выделяем память для динамического массива

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

array[i] = -10 + (double)rand() / RAND\_MAX \* 20; //заполнение массива случайными числами с помощью функции rand()

cout << array[i] << " "; //вывод массива в консоль

cout << endl;

}

int zeroelements = 0;

int minsum = 0;

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

if (array[i] < array[minelement]) {

minelement = i;

}

}

//подсчитываем количество элементов, равных 0, и суммируем элементы после минимального элемента

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

if (array[i] == 0) {

zeroelements++;

}

if (i > minelement) {

minsum += array[i];

}

}

// Вывод результатов

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

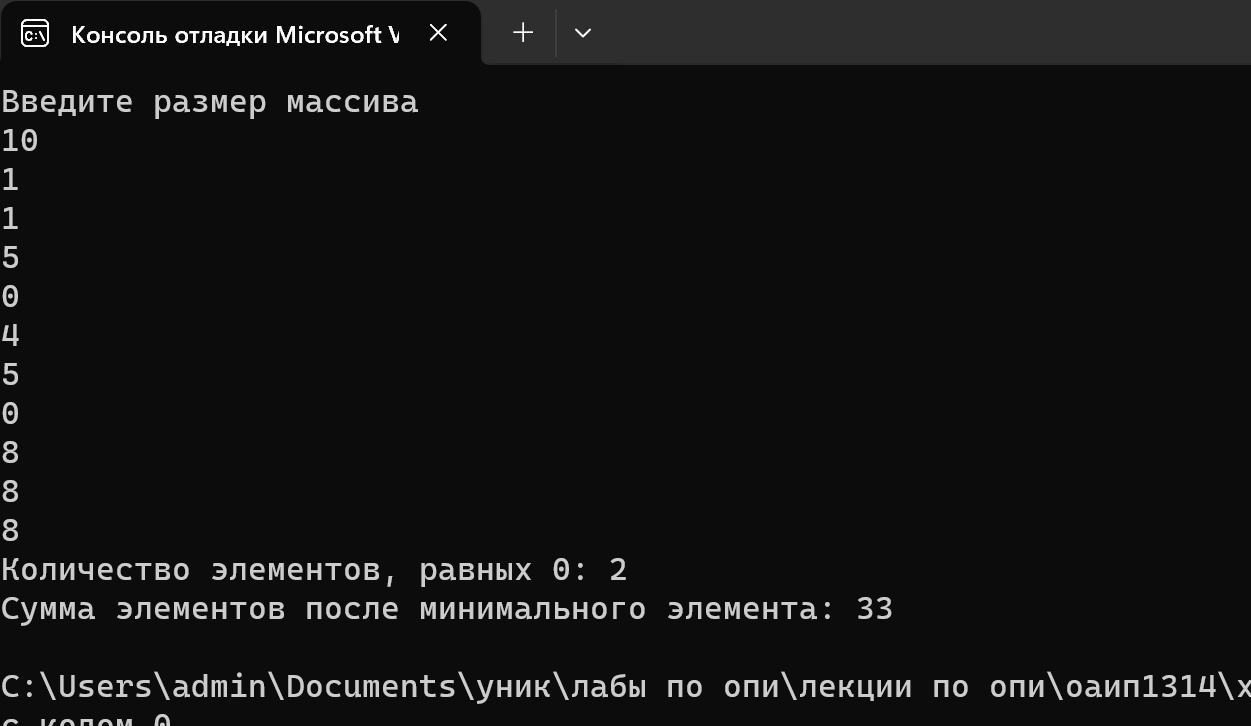
cout << "Сумма элементов после минимального элемента: " << minsum << endl;

// Освобождение динамической памяти

delete[] array;

return 0;

}

****

**Задание 2**

#include <iostream>//подключаем библиотеку для работы с вводом/выводом

using namespace std;//подключаем пространство имен std

int negelement(int\*\* array, int n, int m) {//с помощью функции находим отрицательный элемент

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

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

if (array[i][j] < 0)

return i;

}

}

}

int main() {

srand(time(NULL));

setlocale(LC\_ALL, "RUS");//функция для корректного отображения русского языка

int rows, columns; //инициализируем переменные

cout << "Введите количество строк матрицы: ";//выводим сообщение на экран

cin >> rows;//вводим значение переменной с клавиатуры

cout << "Введите количество столбцов матрицы: ";//выводим сообщение на экран

cin >> columns;//вводим значение переменной с клавиатуры

int\*\* array = new int\* [rows];//создание динамического массива с помощью операции new

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

array[i] = new int[columns];

}

for (int i = 0; i < rows; i++) {//заполнение матрицы рандомными числами с помощью rand()

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

array[i][j] = rand() % (10 + 1);

}

}

//в любую ячейку матрицы подставляем отрицательный элемент

array[rand() % ((rows - 1) + 1)][rand() % ((columns - 1) + 1)] = -2;

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

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

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

}

cout << endl;

}

cout << endl;

//поиск строки с самой длинной серией одинаковых элементов

int longestrow = -1;

int maxLength = 0;

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

int length = 1;

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

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

length++;

}

else {

length = 1;

}

if (length > maxLength) {

maxLength = length;

longestrow = i;

}

}

}

cout << "Строка с самой длинной серией одинаковых элементов: " << longestrow << endl;

//подсчет количества столбцов с хотя бы одним нулевым элементом

int countZero = 0;

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

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

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

countZero++;

break;

}

}

}

//вывод результата

cout << "Кол-во столбцов c нулевым(и) элементом(и): " << countZero << endl;

//освобождение динамической памяти

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

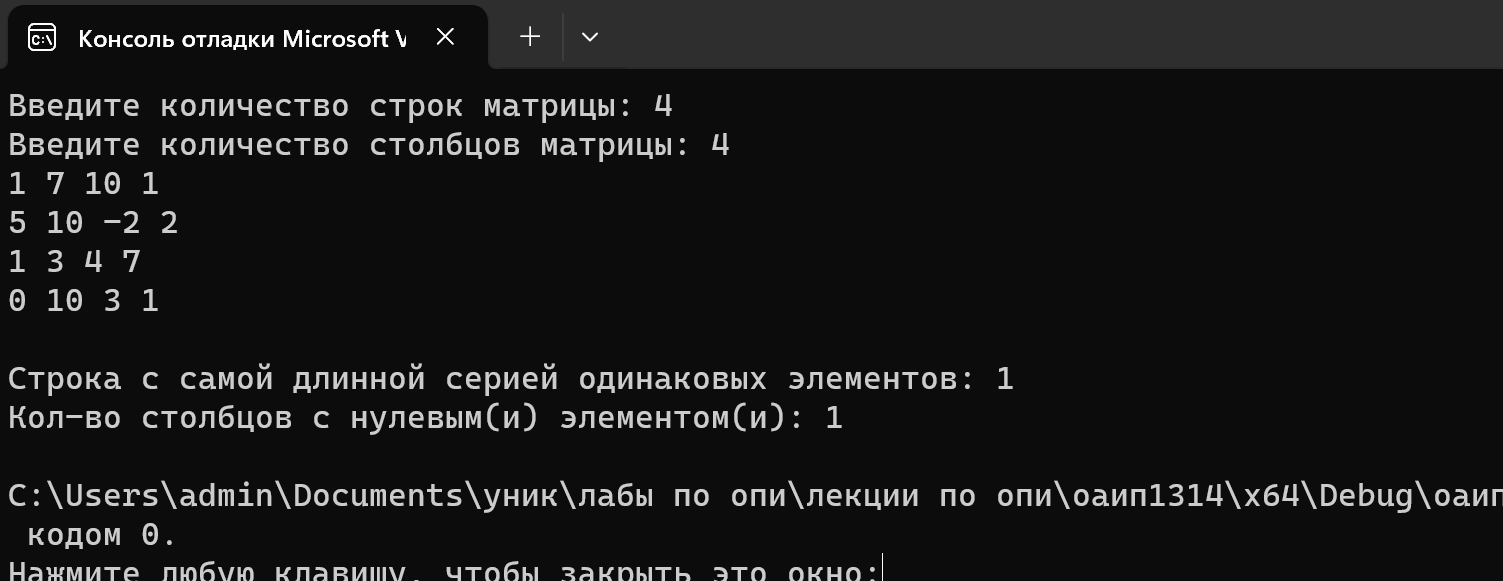
delete[] array[i];

}

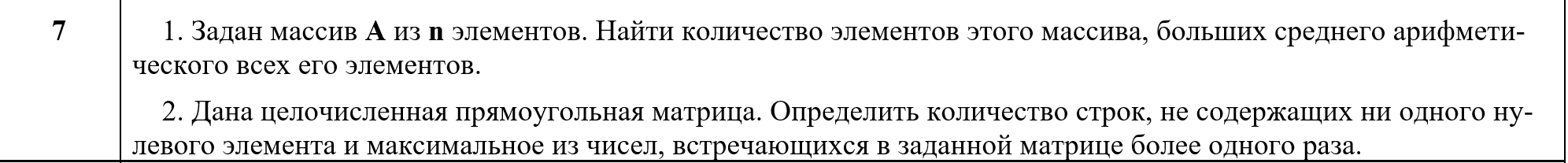
delete[] array;

return 0;

}

****

**Вариант 7**

****

**Задание 1**

#include<iostream>//подключаем библиотеку для работы с вводом/выводом

using namespace std;//подключаем пространство имен std

int main()

{

setlocale(LC\_ALL, "RUS");//функция для корректного отображения русского языка

const int MAX\_SIZE = 100; // Максимальный размер массива

int n;

int array[MAX\_SIZE];

// Ввод размера массива

cout << "Введите размер массива (n <= " << MAX\_SIZE << "): ";

cin >> n;

// Ввод элементов массива

cout << "Введите элементы массива:\n";

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

cout << "Элемент " << i + 1 << ": ";

cin >> array[i];

}

// Вычисление среднего арифметического

int sum = 0;

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

sum += array[i];

}

double average = static\_cast<double>(sum) / n;

// Подсчет количества элементов, больших среднего

int countGreaterThanAverage = 0;

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

if (array[i] > average) {

countGreaterThanAverage++;

}

}

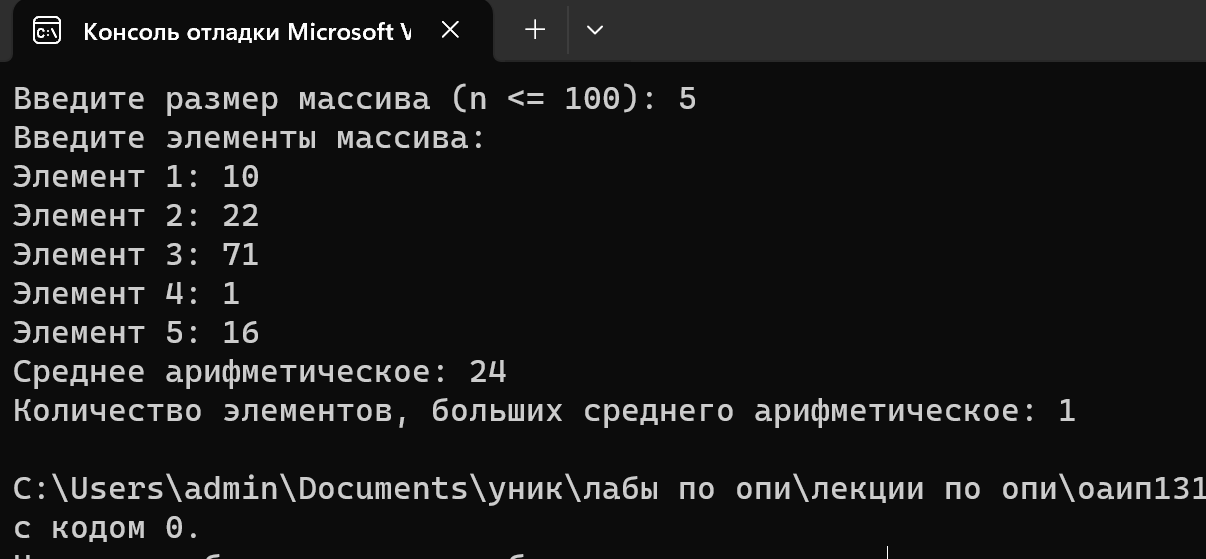
// Вывод результата

cout << "Среднее арифметическое: " << average << endl;

cout << "Количество элементов, больших среднего арифметическое: " << countGreaterThanAverage << endl;

return 0;

}



**Задание 2**

#include<iostream>//подключаем библиотеку для работы с вводом/выводом

#include <climits>

using namespace std;//подключаем пространство имен std

int main()//основная функция

{

setlocale(LC\_ALL, "ru");//функция для корректного отображения русского языка

srand((unsigned)time(NULL));

const int MAX\_ROWS = 100; // Максимальное количество строк

const int MAX\_COLS = 100; // Максимальное количество столбцов

int array[MAX\_ROWS][MAX\_COLS];

int rows, cols;

// Ввод размеров матрицы

cout << "Введите количество строк матрицы: ";

cin >> rows;

cout << "Введите количество столбцов матрицы: ";

cin >> cols;

// Ввод элементов матрицы

std::cout << "Введите элементы матрицы:\n";

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

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

cout << "Элемент [" << i << "][" << j << "]: ";

cin >> array[i][j];

}

}

// Подсчет строк без нулевых элементов

int countRowsWithoutZeros = 0;

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

bool hasZero = false;

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

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

hasZero = true;

break;

}

}

if (!hasZero) {

countRowsWithoutZeros++;

}

}

// Поиск максимального числа, встречающегося более одного раза

int maxRepeatingNumber = INT\_MIN;

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

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

int currentNumber = array[i][j];

bool foundDuplicate = false;

// Проверка, есть ли такое число еще где-то в матрице

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

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

if (k != i || l != j) {

if (array[k][l] == currentNumber) {

foundDuplicate = true;

break;

}

}

}

if (foundDuplicate) {

break;

}

}

// Обновление максимального повторяющегося числа

if (foundDuplicate && currentNumber > maxRepeatingNumber) {

maxRepeatingNumber = currentNumber;

}

}

}

// Вывод результатов в консоль

cout << "Количество строк без нулевых элементов: " << countRowsWithoutZeros << endl;

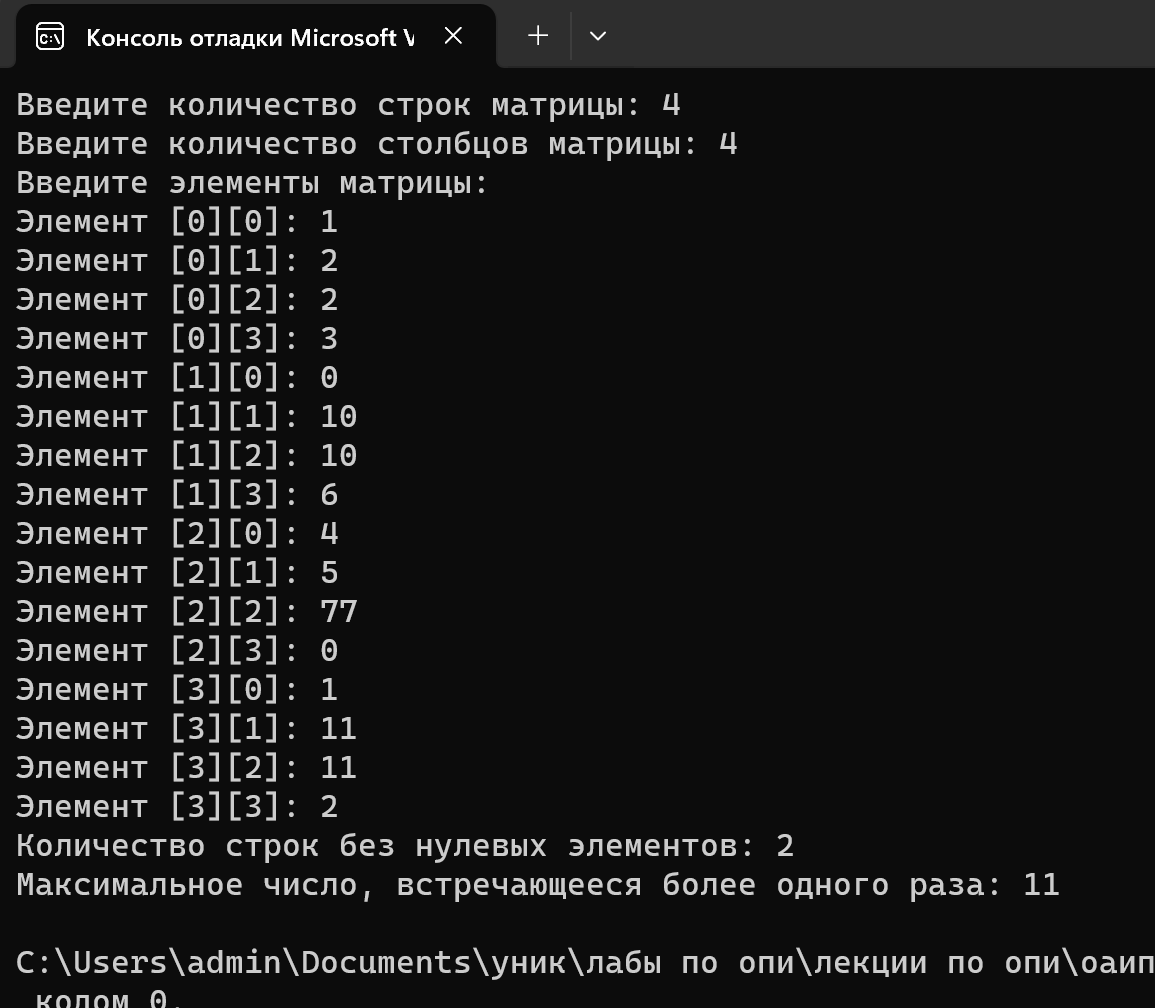
if (maxRepeatingNumber != INT\_MIN) {

cout << "Максимальное число, встречающееся более одного раза: " << maxRepeatingNumber << endl;

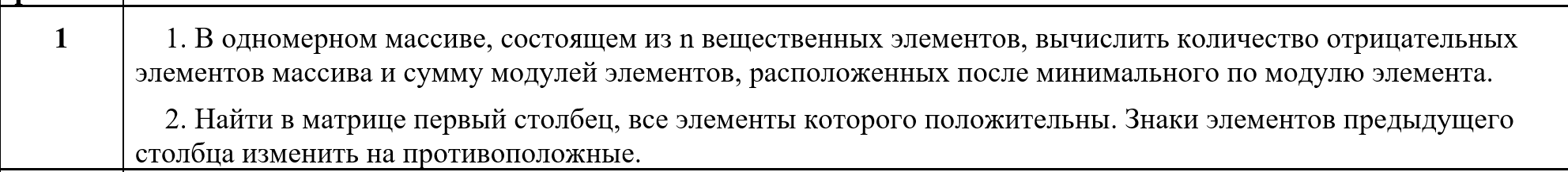
}

return 0;

}

****

**Вариант 1**

****

**Задание 1**

#include<iostream>//подключаем библиотеку для работы с вводом/выводом

#include <cmath>

using namespace std;//подключаем пространство имен std

int main()

{

setlocale(LC\_ALL, "RUS");//функция для корректного отображения русского языка

const int MAX\_SIZE = 100; // Максимальный размер массива

int n;

double array[MAX\_SIZE];

// Ввод размера массива

cout << "Введите размер массива (n <= " << MAX\_SIZE << "): ";

cin >> n;

// Ввод элементов массива

cout << "Введите элементы массива:\n";

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

cout << "Элемент " << i + 1 << ": ";

cin >> array[i];

}

// Поиск минимального по модулю элемента

int minAbsIndex = 0;

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

if (abs(array[i]) < std::abs(array[minAbsIndex])) {

minAbsIndex = i;

}

}

// Подсчет количества отрицательных элементов

int countNegatives = 0;

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

if (array[i] < 0) {

countNegatives++;

}

}

// Сумма модулей элементов после минимального по модулю

double sumAfterMinAbs = 0;

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

sumAfterMinAbs += abs(array[i]);

}

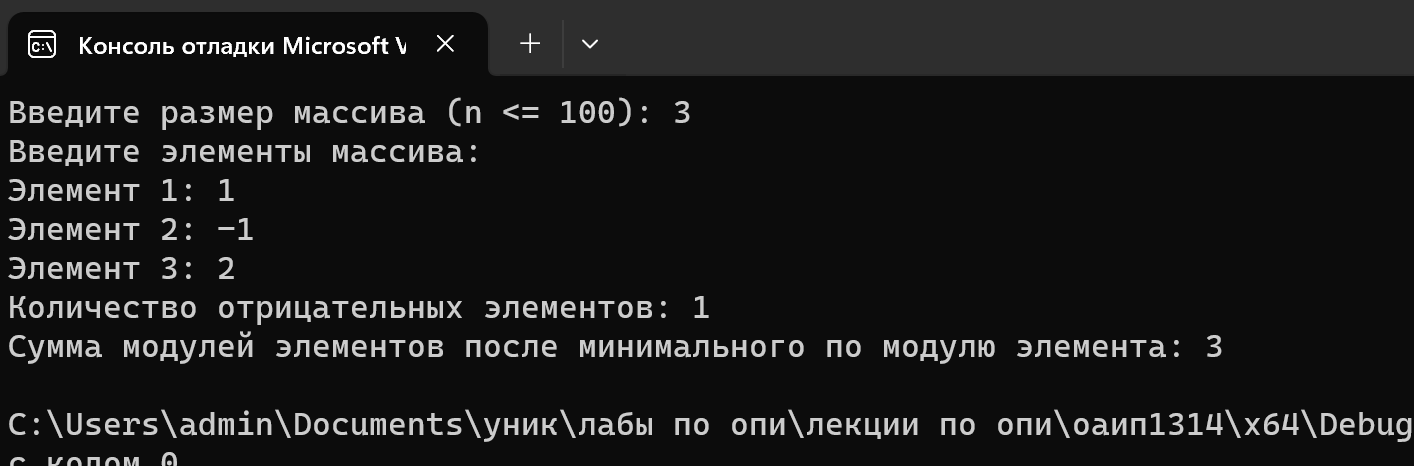
// Вывод результата в консоль

cout << "Количество отрицательных элементов: " << countNegatives << endl;

cout << "Сумма модулей элементов после минимального по модулю элемента: " << sumAfterMinAbs << endl;

return 0;

}

****

**Задание 2**

#include<iostream>//подключаем библиотеку для работы с вводом/выводом

using namespace std;//подключаем пространство имен std

int main()

{

setlocale(LC\_ALL, "RUS");//функция для корректного отображения русского языка

srand(time(NULL));

int rows, columns; //инициализируем переменные

cout << "Введите количество строк матрицы: ";//выводим сообщение на экран

cin >> rows;//вводим значение переменной с клавиатуры

cout << "Введите количество столбцов матрицы: ";//выводим сообщение на экран

cin >> columns;//вводим значение переменной с клавиатуры

int\*\* array = new int\* [rows];//создание динамического массива с помощью операции new

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

array[i] = new int[columns];

}

for (int i = 0; i < rows; i++) {//заполнение матрицы рандомными числами с помощью rand()

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

array[i][j] = rand() % (10 + 1);

}

}

//в любую ячейку матрицы подставляем отрицательный элемент

array[rand() % ((rows - 1) + 1)][rand() % ((columns - 1) + 1)] = -2;

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

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

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

}

cout << endl;

}

cout << endl;

// Поиск первого столбца с положительными элементами

int positiveColumn = -1;

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

bool allPositive = true;

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

if (array[i][j] <= 0) {

allPositive = false;

break;

}

}

if (allPositive) {

positiveColumn = j;

break;

}

}

// Если найден столбец с положительными элементами, изменяем знаки элементов предыдущего столбца

if (positiveColumn > 0) {

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

array[i][positiveColumn - 1] = -array[i][positiveColumn - 1];

}

// Вывод измененной матрицы

cout << "Измененная матрица:\n";

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

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

std::cout << array[i][j] << ' ';

}

cout << '\n';

}

}

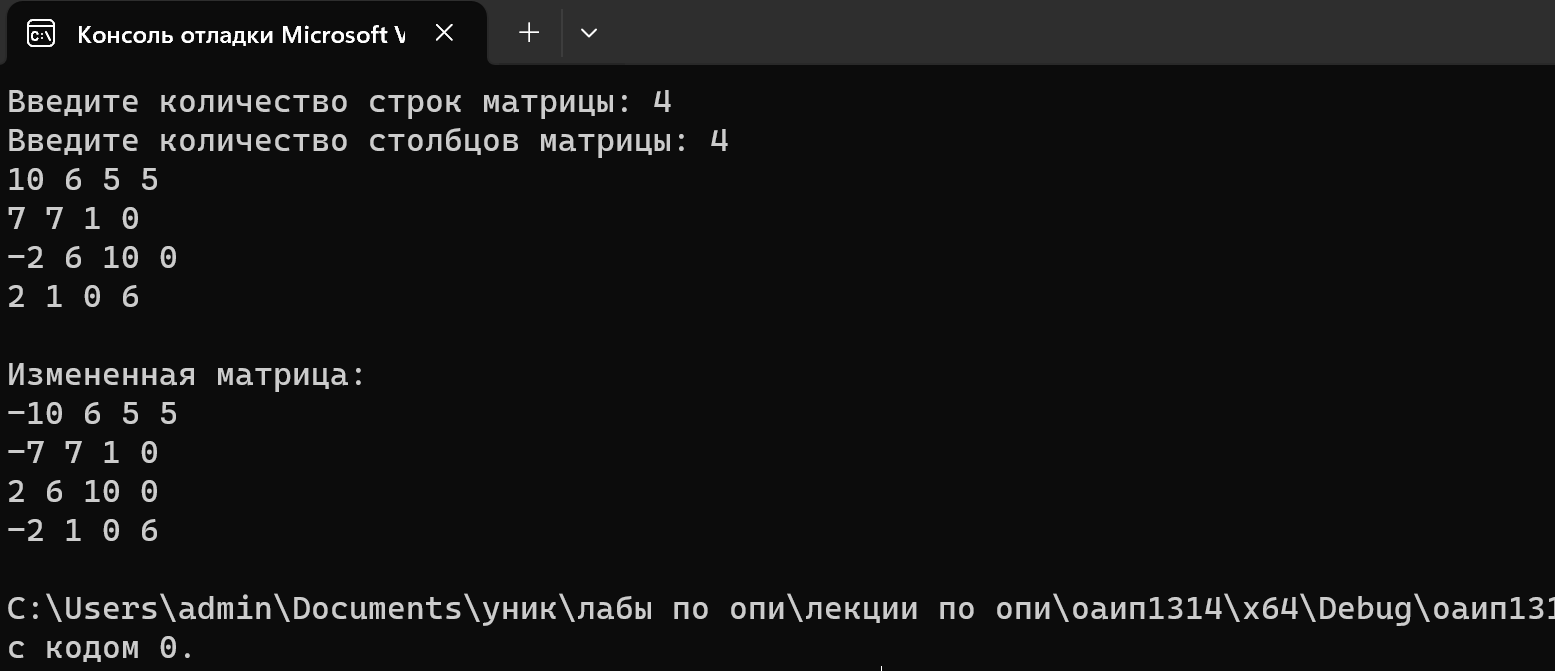
else {

cout << "В матрице нет столбца, все элементы которого положительны.\n";

}

return 0;

}

****