



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных  
технологий

## **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

**Тема: «Поразрядные операции и их применение»**

Выполнил студент группы ИКБО-18-22

Ракитин В.А.

Принял преподаватель

Филатов А.С.

Лабораторная работа выполнена

«\_\_»\_\_\_\_\_202\_\_ г.

*(подпись студента)*

«Зачтено»

«\_\_»\_\_\_\_\_202\_\_ г.

*(подпись руководителя)*

Москва 2023

## **1. Цель работы**

Получение навыков по определению двумерного массива для структуры данных задачи. Получение навыков по разработке алгоритмов операций на двумерном массиве в соответствии с задачей.

## **2. Постановка задачи**

1. Разработать программу по обработке данных, представленных в задаче матрицей и реализованной в программе двумерным (многомерным) статическим массивом.
  - a. Максимально допустимые размеры статического массива установить через глобальные константы Rows, Cols.
  - b. Если тип элемента массива не определен в варианте, определить массив для хранения значений целого типа.
  - c. Разработать функции по реализации алгоритмов заполнения матрицы: с клавиатуры, датчиком случайных чисел. Разработать функции по реализации алгоритма вывода массива на экран построчно.
  - d. Выполнить декомпозицию задачи варианта, разработать алгоритм решения. Реализовать функцию, выполняющую задачу и отладить ее.
  - e. Разработать программу, демонстрирующую работу всех функций.
2. Разработать программу по обработке данных, представленных в задаче матрицей и реализованной в программе двумерным (многомерным) динамическим массивом.
  - a. Размеры массива должны определяться пользователем.
  - b. Двумерный массив определить как двойной указатель и выполнить его создание операцией new.
  - c. Разработать функции по реализации алгоритмов заполнения матрицы: с клавиатуры, датчиком случайных чисел. Разработать функции по реализации алгоритма вывода массива на экран построчно.
  - d. Выполнить декомпозицию задачи варианта, разработать алгоритм решения. Реализовать функцию, выполняющую задачу и отладить ее.

- е. Разработать программу, демонстрирующую работу всех функций.
- 3. Разработать программу по задаче варианта с реализацией данных задачи с применением контейнера `vector` библиотеки STL.
  - а. Реализовать структуру хранения данных на основе шаблона `<vector>`, размеры определить при вводе с клавиатуры.
  - б. Разработать функции: заполнение структуры хранения исходных данных, вывода структуры хранения.
  - с. Выполнить декомпозицию задачи варианта, разработать алгоритм решения. Реализовать функцию, выполняющую задачу и отладить ее.
  - д. Разработать программу, демонстрирующую работу всех функций.
- 4. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Вариант №21. Условие задания:

Упражнение 1	Дана целочисленная квадратная матрица порядка $n$ . Найти номера строк, элементы каждой из которых образуют монотонную последовательность (монотонно убывающую или монотонно возрастающую).
Упражнение 2	<p>На плоскости задано множество точек с целочисленными координатами. Необходимо найти количество отрезков, обладающих следующими свойствами:</p> <ol style="list-style-type: none"> <li>1) оба конца отрезка принадлежат заданному множеству;</li> <li>2) ни один конец отрезка не лежит на осях координат;</li> <li>3) отрезок пересекается ровно с одной осью координат.</li> </ol> <p>Напишите эффективную по времени и по используемой памяти программу для решения этой задачи.</p>

### 3. Решение

Что такое двумерный массив? Двумерный массив в языке программирования C++ – это список одномерных массивов. Двумерные массивы представляют собой некую таблицу чисел. Для доступа к элементам двумерного массива нужно указывать два индекса: первый индекс определяет строку в этой таблице, а второй индекс определяет столбец таблицы.

Статический двумерный массив – это структура данных, которая создается во время компиляции (т.е. массиву выделяется память). Размер такого массива нельзя изменять во время работы программы.

Динамический двумерный массив – это массив с переменным размером. Это значит, что размер такого массива можно изменять по ходу выполнения работы программы. Такой тип массива очень удобен, т.к. это позволяет не занимать лишнюю память.

Двумерный вектор – это реализация динамического двумерного массива. Двумерный вектор, в отличие от динамического двумерного массива, может самостоятельно изменять размер.

Для решения первого упражнения была написана функция `func_mass4`, которая ищет монотонно-возрастающие строки. На вход функция принимает наш двумерный массив. Далее функция проверяет, чтоб все числа в строке были больше предыдущих. Если такие строки есть, то они монотонно-возрастающие и программа выведет номера этих строк. Если таких строк нет, то программа сообщит пользователю об отсутствии таких строк.

```
int func_mass4(int ptr_arr[ROWS][COWS]) //Функция, выводящая номера
монотонно-возрастающих строк
{
    int k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if (ptr_arr[i][j - 1] < ptr_arr[i][j]) {
                Flag = 1;
            }
            else {
```

```

        Flag = 0;
        break;
    }
}
if (Flag == 1) {
    k += 1;
    cout << "Номер монотонно возрастающей строки: " << i + 1
<< endl;
}
}
if (k == 0) {
    cout << "В вашей матрице нет монотонно возрастающих строк" <<
endl;
}
return 0;
}

```

Также для решения первого упражнения была написана функция func\_mass5, которая выводит номера монотонно-убывающих строк. Работа этой функции схожа с функцией func\_mass4. Функция получает на вход двумерный массив, и ищет строки, где каждое следующее число меньше предыдущего.

```

int func_mass5(int ptr_arr[ROWS][COWS])//Функция, выводящая номера
монотонно-убывающих строк
{
    int k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if (ptr_arr[i][j - 1] > ptr_arr[i][j]) {
                Flag = 1;
            }
            else {
                Flag = 0;
                break;
            }
        }
        if (Flag == 1) {
            k += 1;
            cout << "Номер монотонно убывающей строки: " << i + 1 <<
endl;
        }
    }
    if (k == 0) {
        cout << "В вашей матрице нет монотонно убывающих строк" << endl;
    }
    return 0;
}

```

Для решения второго упражнения была написана функция func\_vek4, которая ищет отрезки, концы которых не лежат на осях координат и пересекающие ось координат ровно один раз. На вход функция принимает наш

двумерный массив. Далее функция сравнивает точки, чтоб только одна координат точек отличалось по знаку, и чтобы координаты точек не были равны нулю (иначе точка лежит на оси координат). Потом функция выводит пользователю количество таких отрезков.

```
int func_vek4(vector<vector<int>>& arr) { //Функция, решающая поставленную задачу
    int k = 0;
    for (int i = 0; i < n - 1; i++) {
        if (arr[i][0] != 0 and arr[i][1] != 0){
            for (int j = i + 1; j < n; j++) {
                if (((arr[i][0] > 0 and arr[j][0] < 0) or
(arr[i][0] < 0 and arr[j][0] > 0)) and ((arr[i][1] > 0 and arr[j][1] > 0) or
(arr[i][1] < 0 and arr[j][1] < 0))) or
                (((arr[i][1] > 0 and arr[j][1] < 0) or
(arr[i][1] < 0 and arr[j][1] > 0)) and ((arr[i][0] > 0 and arr[j][0] > 0) or
(arr[i][0] < 0 and arr[j][0] < 0)))) {
                    k += 1;
                }
            }
        }
    }
    cout << endl << "Ответ: " << k;
    return 0;
}
```

При запуске программы для первого упражнения пользователь видит пользовательское меню, где необходимо установить размер двумерного массива. Далее можно сделать выбор между ручным вводом и автоматическим вводом чисел.

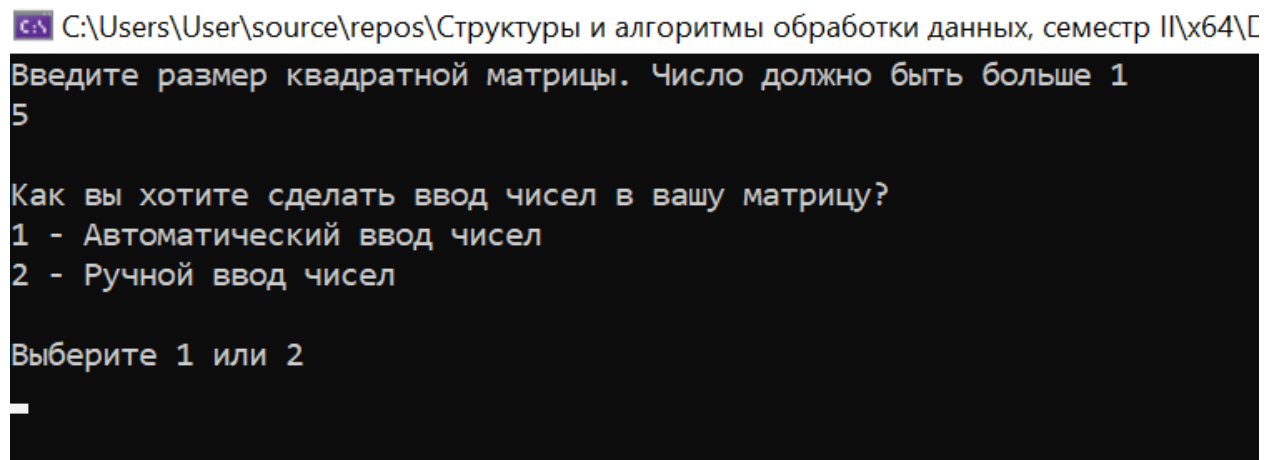


Рисунок 1. Интерфейс программы для первого упражнения

При запуске программы для второго упражнения пользователь видит пользовательское меню, где необходимо ввести количество вводимых точек. Далее также будет выбор между ручным вводом и автоматическим вводом чисел.

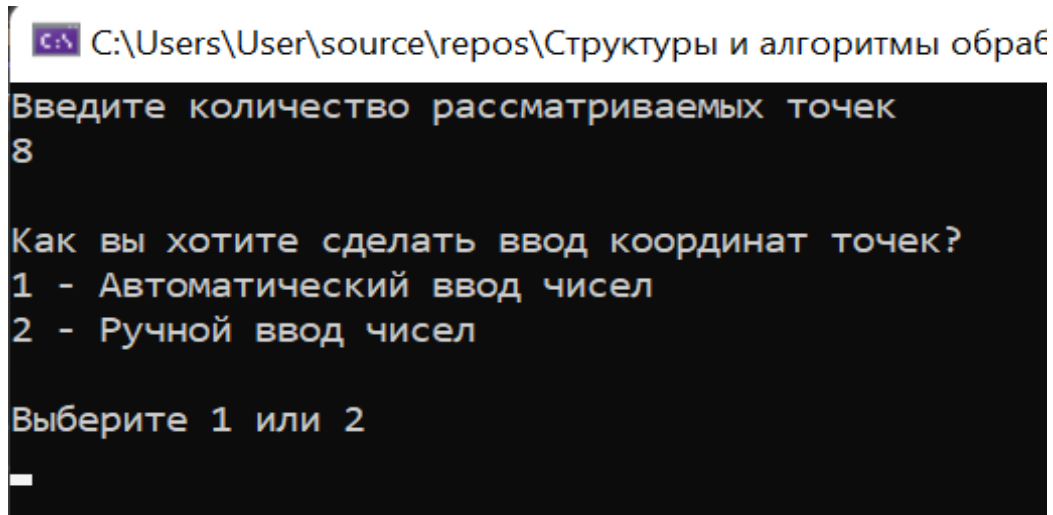


Рисунок 2. Интерфейс программы для второго упражнения

#### 4. Тестирование

Протестируем программой выполнение первого упражнения. Выберем размер двумерного массива  $3 \times 3$  и введём следующие 9 чисел: 23, 46, 47, 23, 20, 24, 13, 14, 89. Первые три числа лежат на первой строке, вторые числа на второй и так далее. Первая строка является возрастающей, т.к.  $23 < 46 < 47$ . Вторая строка не является ни возрастающей, ни убывающей, т.к.  $23 > 20 < 24$ . Третья строка является возрастающей, т.к.  $13 < 14 < 89$ . Программа должна вывести пользователю номера монотонно-возрастающих строк (в нашем случае – это 1 и 3) и сообщить об отсутствии монотонно-убывающих строк. На рисунке 3 программа выдала верный результат.

```
C:\Users\User\source\repos\Структуры и алгоритмы обработки данных, семестр II\64\Debug\Структуры и алгоритмы обработки данных, семестр II.exe
Введите размер квадратной матрицы. Число должно быть больше 1
3

Как вы хотите сделать ввод чисел в вашу матрицу?
1 - Автоматический ввод чисел
2 - Ручной ввод чисел

Выберите 1 или 2
2

Введите любые 9 целых чисел через Enter:
arr[0][0] = 23
arr[0][1] = 46
arr[0][2] = 47
arr[1][0] = 23
arr[1][1] = 20
arr[1][2] = 24
arr[2][0] = 13
arr[2][1] = 14
arr[2][2] = 89

Выводим ваш двумерный массив:

23    46    47
23    20    24
13    14    89

Номер монотонно возрастающей строки: 1
Номер монотонно возрастающей строки: 3

В вашей матрице нет монотонно убывающих строк
```

Рисунок 3. Решение первого упражнения

Протестируем программой выполнение второго упражнения. Введём ручным вводом следующие пять точек, а точнее – их координаты на плоскости:  $(-3; 10)$ ,  $(-2; 3)$ ,  $(4; 5)$ ,  $(3; -1)$ ,  $(12; 4)$ . Обозначим данные точки буквами А, В, С, D, Е соответственно. На рисунке 4 мы видим их расположение этих пяти точек относительно осей координат.



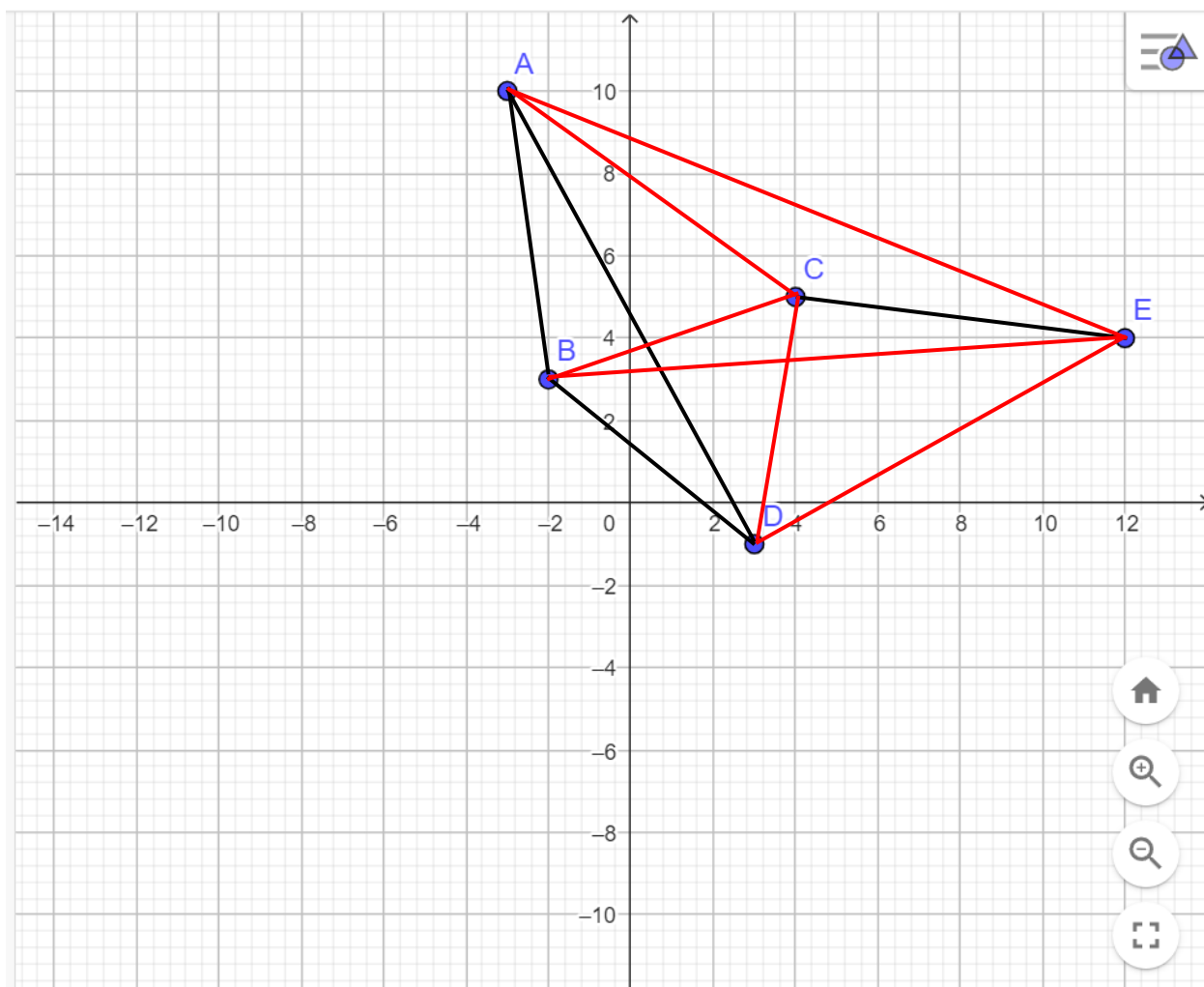


Рисунок 4. Расположение точек относительно осей координат.

У нас получилось 10 отрезков:  $AB$ ,  $AC$ ,  $AD$ ,  $AE$ ,  $BC$ ,  $BD$ ,  $BE$ ,  $CD$ ,  $CE$ ,  $DE$ . Смотрим, какие отрезки пересекают ось координат ровно один раз. Отрезки, которые удовлетворяют условию задачи, изображены на рисунке красным цветом. Отрезки  $AC$ ,  $AD$ ,  $AE$ ,  $BC$ ,  $BE$  и  $DE$  пересекают ось координат ровно один раз. Концы этих отрезков не лежат на осях координат. Следовательно, программа выведет количество этих отрезков, равная 6. На рисунке 5 мы видим, что программа дала верный ответ.

```
Консоль отладки Microsoft Visual Studio
Введите количество рассматриваемых точек
5
Как вы хотите сделать ввод координат точек?
1 - Автоматический ввод чисел
2 - Ручной ввод чисел
Выберите 1 или 2
2
Введите любые 10 целочисленных координат X и Y через Enter:
arr[0][0] = -3
arr[0][1] = 10
arr[1][0] = -2
arr[1][1] = 3
arr[2][0] = 4
arr[2][1] = 5
arr[3][0] = 3
arr[3][1] = -1
arr[4][0] = 12
arr[4][1] = 4
Выводим ваш двумерный массив:
-3    10
-2     3
 4     5
 3    -1
12     4
Ответ: 6
```

Рисунок 5. Решение второго упражнения

## 5. Вывод

В результате выполнения работы я:

1. Получил навыки по определению двумерного массива для структуры данных задачи
2. Получил навыки по разработке алгоритмов операций на двумерном массиве

## 6. Исходный код программы

```
#include <iostream> //Библиотека для ввода/вывода в консоль
#define ROWS 10
#define COWS 10
using namespace std; //Пространство имён std
int n, Flag = 0;

int func_mass2(int ptr_arr[ROWS][COWS]) //Функция ручного ввода значений в массив
{
    cout << endl << "Введите любые " << n * n << " целых чисел через
Enter: " << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << "arr[" << i << "][" << j << "] = ";
            if (cin >> ptr_arr[i][j]) { //Проверка, что введено целое
число
                Flag = 1;
            }
            else {
                Flag = 0;
            }
        }
    }
}
```

```

        break;
    }
    }
    if (Flag == 0) {
        cout << "Некорректный ввод!" << endl; //Реакция на
некорректный ввод
        break;
    }
}
return 0;
}

int func_mass1(int ptr_arr[ROWS][COWS]) //Функция автоматического ввода
значений в массив
{
    srand((unsigned)time(NULL));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            ptr_arr[i][j] = rand() % 100; //Заполняем массив случайными
числами
        }
    }
    return 0;
}

int func_mass3(int ptr_arr[ROWS][COWS]) //Вывод двумерного динамического
массива
{
    cout << endl << "Выводим ваш двумерный массив:" << endl << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << ptr_arr[i][j] << "\t";
        }
        cout << endl;
    }
    return 0;
}

int func_mass4(int ptr_arr[ROWS][COWS]) //Функция, выводящая номера
монотонно-возрастающих строк
{
    int k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if (ptr_arr[i][j - 1] < ptr_arr[i][j]) {
                Flag = 1;
            }
            else {
                Flag = 0;
                break;
            }
        }
        if (Flag == 1) {
            k += 1;
            cout << "Номер монотонно возрастающей строки: " << i + 1
<< endl;
        }
    }
    if (k == 0) {
        cout << "В вашей матрице нет монотонно возрастающих строк" <<
endl;
    }
    return 0;
}

```

```

int func_mass5(int ptr_arr[ROWS][COWS])//Функция, выводящая номера
монотонно-убывающих строк
{
    int k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if (ptr_arr[i][j - 1] > ptr_arr[i][j]) {
                Flag = 1;
            }
            else {
                Flag = 0;
                break;
            }
        }
        if (Flag == 1) {
            k += 1;
            cout << "Номер монотонно убывающей строки: " << i + 1 <<
endl;
        }
    }
    if (k == 0) {
        cout << "В вашей матрице нет монотонно убывающих строк" << endl;
    }
    return 0;
}

int main()
{
    setlocale(LC_ALL, "Russian"); //Установка русского языка для вывода в
консоль
    int arr[ROWS][COWS], a;
    cout << "Введите размер квадратной матрицы. Число должно быть больше 1
и меньше " << ROWS << endl;
    cout << "Максимальный размер матрицы : " << ROWS << "x" << COWS <<
endl;
    if (cin >> n) { //Проверка, что введено целое число
        if (n > 1 && n <= ROWS && n <= COWS) { //Проверка, что введённое
целое число больше 1 и меньше максимального размера статического двумерного
массива
            cout << endl << "Как вы хотите сделать ввод чисел в вашу
матрицу?" << endl;
            cout << "1 - Автоматический ввод чисел" << endl;
            cout << "2 - Ручной ввод чисел" << endl;
            cout << endl << "Выберите 1 или 2" << endl;
            cin >> a; //Выбор пользователя
            switch (a)
            {
                case 1:
                    Flag = 1;
                    func_mass1(arr); // Вызываем функцию, заполняющую
статический массив случайными числами
                    getchar();
                    break;
                case 2:
                    func_mass2(arr); //Вызываем функцию, заполняющую
статический массив числами с помощью ручного ввода
                    getchar();
                    break;
                default:
                    cout << "Некорректный ввод!" << endl; //Реакция на
некорректный ввод
                    break;
            }
        }
    }
}

```

```

    }
    else {
        Flag = 0;
        cout << "Некорректный ввод!" << endl; //Реакция на
некорректный ввод
    }
}
else {
    Flag = 0;
    cout << "Некорректный ввод!" << endl; //Реакция на некорректный
ввод
}
if (Flag == 1) {
    func_mass3(arr); //Вызываем функцию, которая выводит наш
статический массив
    cout << endl;
    func_mass4(arr); //Вызываем функцию, которая выводит номера
монотонно-возрастающих строк
    cout << endl;
    func_mass5(arr); //Вызываем функцию, которая выводит номера
монотонно-убывающих строк
    getchar();
}
return 0; //Передача информации операционной системе о завершении
работы программы
}

```

Таблица 1. Код программы на статическом двумерном массиве

```

#include <iostream> //Библиотека для ввода/вывода в консоль
using namespace std; //Пространство имён std

int n, Flag = 0;

int func_mass2(int **ptr_arr) //Функция ручного ввода значений в массив
{
    cout << endl << "Введите любые " << n * n << " целых чисел через
Enter: " << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << "arr[" << i << "][" << j << "] = ";
            if (cin >> ptr_arr[i][j]) { //Проверка, что введено целое
число
                Flag = 1;
            }
            else {
                Flag = 0;
                break;
            }
        }
        if (Flag == 0) {
            cout << "Некорректный ввод!" << endl; //Реакция на
некорректный ввод
            break;
        }
    }
    return 0;
}
}

```

```

int func_mass1(int **ptr_arr) //Функция автоматического ввода значений в массив
{
    srand((unsigned)time(NULL));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            ptr_arr[i][j] = rand() % 100; //Заполняем массив
случайными числами
        }
    }
    return 0;
}

int func_mass3(int **ptr_arr) //Вывод двумерного динамического массива
{
    cout << endl << "Выводим ваш двумерный массив:" << endl << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << ptr_arr[i][j] << "\t";
        }
        cout << endl;
    }
    return 0;
}

int func_mass4(int **ptr_arr) //Функция, выводящая номера монотонно-
возрастающих строк
{
    int k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if (ptr_arr[i][j - 1] < ptr_arr[i][j]) {
                Flag = 1;
            }
            else {
                Flag = 0;
                break;
            }
        }
        if (Flag == 1) {
            k += 1;
            cout << "Номер монотонно возрастающей строки: " << i + 1
<< endl;
        }
    }
    if (k == 0) {
        cout << "В вашей матрице нет монотонно возрастающих строк" <<
endl;
    }
    return 0;
}

int func_mass5(int **ptr_arr) //Функция, выводящая номера монотонно-
убывающих строк
{
    int k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if (ptr_arr[i][j - 1] > ptr_arr[i][j]) {
                Flag = 1;
            }
            else {
                Flag = 0;
                break;
            }
        }
    }
}

```

```

    }
    }
    if (Flag == 1) {
        k += 1;
        cout << "Номер монотонно убывающей строки: " << i + 1 <<
endl;
    }
}
if (k == 0) {
    cout << "В вашей матрице нет монотонно убывающих строк" << endl;
}
return 0;
}

int main()
{
    setlocale(LC_ALL, "Russian"); //Установка русского языка для вывода в
консоль
    int a;
    cout << "Введите размер квадратной матрицы. Число должно быть больше
1" << endl;
    if (cin >> n and n > 1) { //Проверка, что введено целое число и оно
больше 1
        Flag = 1;
    }
    int** arr = new int* [abs(n)]; //Создание двумерного динамического
массива
    for (int i = 0; i < n; i++) {
        arr[i] = new int[abs(n)];
    }
    if (Flag == 1) {
        cout << endl << "Как вы хотите сделать ввод чисел в вашу
матрицу?" << endl;
        cout << "1 - Автоматический ввод чисел" << endl;
        cout << "2 - Ручной ввод чисел" << endl;
        cout << endl << "Выберите 1 или 2" << endl;
        cin >> a; //Выбор пользователя
        switch (a)
        {
            case 1:
                Flag = 1;
                func_mass1(arr); //Вызываем функцию, заполняющую
динамический массив случайными числами
                getchar();
                break;
            case 2:
                func_mass2(arr); //Вызываем функцию, заполняющую
динамический массив числами с помощью ручного ввода
                getchar();
                break;
            default:
                cout << "Некорректный ввод!" << endl; //Реакция на
некорректный ввод
                Flag = 0;
                break;
        }
    }
    else {
        Flag = 0;
        cout << "Некорректный ввод!" << endl; //Реакция на некорректный
ввод
    }
    if (Flag == 1) {

```

```

        func_mass3(arr); //Вызываем функцию, которая выводит наш
динамический массив
        cout << endl;
        func_mass4(arr); //Вызываем функцию, которая выводит номера
монотонно-возрастающих строк
        cout << endl;
        func_mass5(arr); //Вызываем функцию, которая выводит номера
монотонно-убывающих строк
        getchar();
    }
    delete[] arr; //Удаляем динамический массив
    return 0; //Передача информации операционной системе о завершении
работы программы
}

```

Таблица 2. Код программы на динамическом двумерном массиве

```

#include <iostream> //Библиотека для ввода/вывода в консоль
#include <vector> //Библиотека векторов
using namespace std; //Пространство имён std

int n, Flag = 0;

int func_vek2(vector<vector<int>>& arr) //Функция ручного ввода значений в
вектор
{
    vector<int> v; //Создаем одномерный векторный массив
    int ak;
    cout << endl << "Введите любые " << 2*n << " целочисленных координат X
и Y через Enter: " << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < 2; j++) {
            cout << "arr[" << i << "][" << j << "] = ";
            if (cin >> ak) { //Проверка, что введено целое число
                Flag = 1;
                v.push_back(ak); //Вставка элемента в конец вектора
            }
            else {
                Flag = 0;
                break;
            }
        }
        if (Flag == 0) {
            break;
        }
        arr.push_back(v); //Добавляем вектор в двумерный вектор
        v.clear(); //Удаляем все элементы вектора
    }
    return 0;
}

int func_vek1(vector<vector<int>>& arr) //Функция автоматического ввода
значений в вектор
{
    vector<int> v; //Создаем одномерный векторный массив
    srand((unsigned)time(NULL));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < 2; j++) {

```



```

        v.push_back( - 99 + (rand() % 199)); //Заполняем массив
        случайными числами
    }
    arr.push_back(v); //Добавляем вектор в двумерный вектор
    v.clear(); //Удаляем все элементы вектора
}
return 0;
}

int func_vek3(vector<vector<int>>& arr) //Вывод двумерного векторного
массива
{
    cout << endl << "Выводим ваш двумерный массив:" << endl << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < 2; j++) {
            cout << arr[i][j] << "\t";
        }
        cout << endl;
    }
    return 0;
}

int func_vek4(vector<vector<int>>& arr) { //Функция, решающая поставленную
задачу
    int k = 0;
    for (int i = 0; i < n - 1; i++) {
        if (arr[i][0] != 0 and arr[i][1] != 0){
            for (int j = i + 1; j < n; j++) {
                if (((arr[i][0] > 0 and arr[j][0] < 0) or
(arr[i][0] < 0 and arr[j][0] > 0)) and ((arr[i][1] > 0 and arr[j][1] > 0) or
(arr[i][1] < 0 and arr[j][1] < 0))) or
                    (((arr[i][1] > 0 and arr[j][1] < 0) or
(arr[i][1] < 0 and arr[j][1] > 0)) and ((arr[i][0] > 0 and arr[j][0] > 0) or
(arr[i][0] < 0 and arr[j][0] < 0)))) {
                    k += 1;
                }
            }
        }
    }
    cout << endl << "Ответ: " << k;
    return 0;
}

int main()
{
    setlocale(LC_ALL, "Russian"); //Установка русского языка для вывода в
консоль
    vector <vector <int>> arr; //Создание двумерного векторного массива
    int a1;
    cout << "Введите количество рассматриваемых точек" << endl;
    if (cin >> n and n > 1) { //Проверка, что введено целое число и оно
больше 1
        Flag = 1;
    }
    else {
        Flag = 0;
    }
    if (Flag == 1) {
        cout << endl << "Как вы хотите сделать ввод координат точек?" <<
endl;
        cout << "1 - Автоматический ввод чисел" << endl;
        cout << "2 - Ручной ввод чисел" << endl;
        cout << endl << "Выберите 1 или 2" << endl;
        cin >> a1; //Выбор пользователя
    }
}

```

```

        switch (a1)
        {
        case 1:
            Flag = 1;
            func_vek1(arr); //Вызываем функцию, заполняющую векторный
массив случайными числами
            break;
        case 2:
            func_vek2(arr); //Вызываем функцию, заполняющую векторный
массив числами с помощью ручного ввода
            break;
        default:
            cout << "Некорректный ввод!" << endl; //Реакция на
некорректный ввод
            Flag = 0;
            break;
        }
    }
    if (Flag == 1) {
        func_vek3(arr); //Выводим наш массив
        func_vek4(arr); //Решаем задачу
    }
    else {
        Flag = 0;
        cout << "Некорректный ввод!" << endl; //Реакция на некорректный
ввод
    }
    return 0; //Передача информации операционной системе о завершении
работы программы
}

```

Таблица 3. Код программы на векторном двумерном массиве