Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Восточно-Сибирский государственный университет технологий и управления

(ФГБОУ ВСГУТУ)

Электротехнический факультет

Кафедра «Системы Информатики»

Дисциплина

«Основы программирования»

Отчет по лабораторной работе №4

На тему:

Массивы

Вариант3

|  |  |
| --- | --- |
| Выполнил: | студент гр. Б661-2 |
|  | Коковихин А.В. |
| Проверил: | преподаватель  Жамбалов Э.Б. |

Улан-Удэ

2021

**1 Словесная постановка задачи**

Выполните индивидуальные задания.

1. В одномерном массиве, состоящем из N целых элементов, вычислите сумму элементов массива, расположенных между первым и последним нулевыми элементами. Если нулевых элементов нет или между первым и последним нулевым элементами нет других элементов, то сумма = 0. Если есть только 1 нулевой элемент, то вычисляется сумма всех элементов до него.
2. Напишите программу, заполняющую двухмерный массив A(N, M) целыми числами, начиная с 0, по возрастанию змейкой (к примеру, если массив A(3, 10), то первая строка будет с 0 до 10, вторая – с 20 до 11, третья – с 21 до 30).

**2 Решение задачи 1**

**2.1 Формальная постановка задачи**

Входные данные:

arr\_size – целое число, обозначающие размер массива чисел;

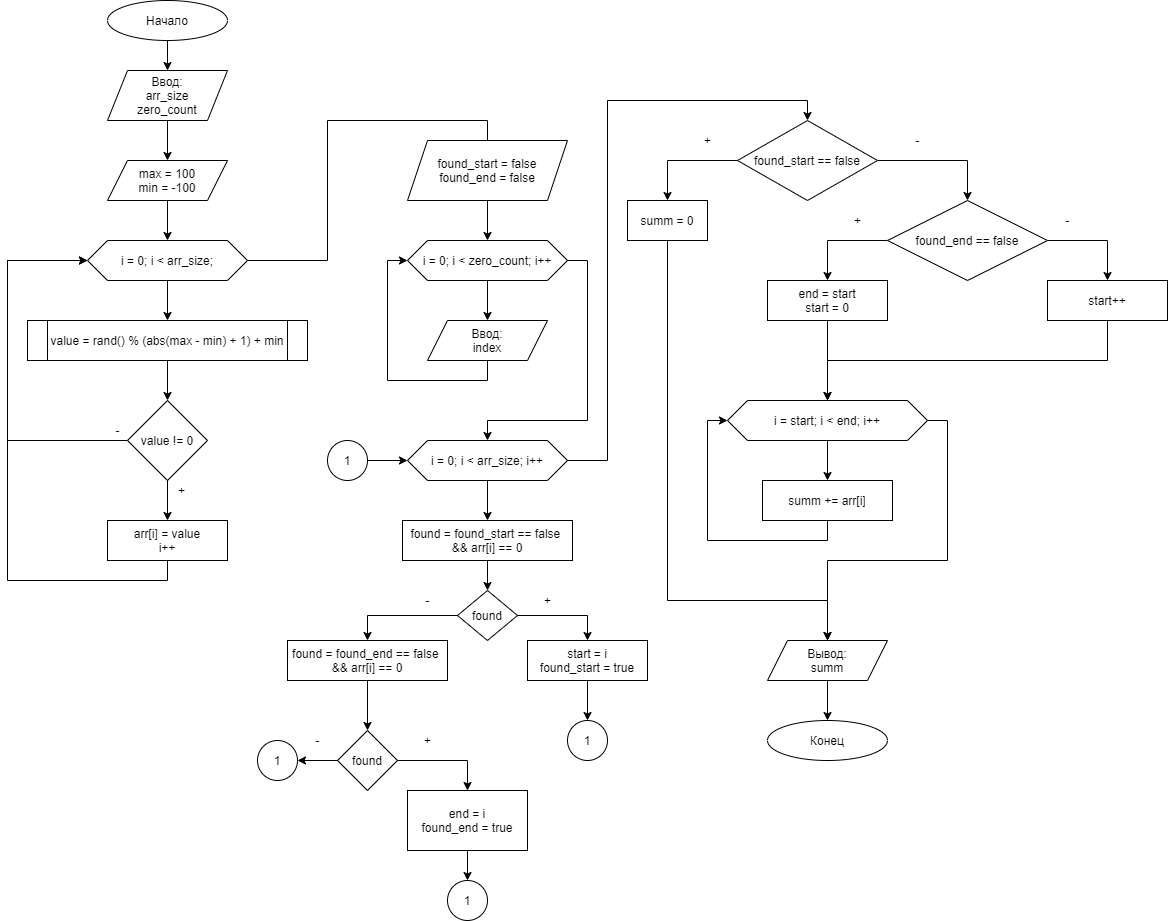
zero\_count – целое число, значащее число ячеек в массиве с нулевым значением;

index – целое число, индекс, под которым будет записан ноль;

Выходные данные:

summ – целое число, итоговая сумма чисел;

**2.2 Проектирование**



**2.3 Реализация**

Запросим у пользователя размер массива:

cout << "Введите размер массива: ";

cin >> arr\_size;

Проинициализируем массив случайными числами:

// Заполнение массива случайными числами не равными 0

for (int i = 0; i < arr\_size; )

{

int value = rand() % (abs(max - min) + 1) + min; // Случайное число на интервале: [min; max]

if (value != 0)

{

arr[i] = value;

i++; // Пока не получится число не равное 0 перехода не будет

}

}

Далее запросим у пользователя номера элементов, которые будут = 0:

cout << "Введите число нулевых элементов: ";

cin >> zero\_count;

bool first\_iteration = true;

for (int i = 0; i < zero\_count; )

{

// Вывод сообщения при первой итерации

if (first\_iteration)

{

cout << "Введите индексы нулевых элементов:\n";

first\_iteration = false;

}

int index = 0;

cin >> index;

if (0 <= index

&& index < arr\_size)

{

arr[index] = 0;

i++; // Пока не получит индекс в пределах масива перехода не будет

}

else

{

cout << "Индекс за пределами массива!" << endl;

}

}

Выведем полученный массив, и между делом найдём первый и последний ноль:

// Вывод массива и поиск нулей

cout << '|'; // Первая граница

for (int i = 0; i < arr\_size; i++)

{

printf("% -4d|", arr[i]); // Вывод числа и границы

if (found\_start == false

&& arr[i] == 0) // Первый ноль найден!

{

start = i;

found\_start = true;

}

else

if (arr[i] == 0) // Второй ноль найден

{

end = i;

found\_end = true;

}

}

cout << endl;

Осталось вычислить сумму и вывести её на экран:

// Вывод результата

cout << "Сумма";

if (found\_start == false) // Нулей нет

{

cout << " = 0";

}

else

{

if (found\_end == false) // Ноль одит

{

end = start;

start = 0;

}

else

{

start++; // Убираем начальный ноль из суммы

}

// Вычисление суммы

for (int i = start; i < end; i++)

{

i == start ? cout << ' ' << arr[i] : cout << " + " << arr[i]; // Вывод порядка действий

summ += arr[i];

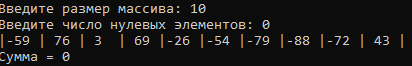
}

cout << " = " << summ; // Результат

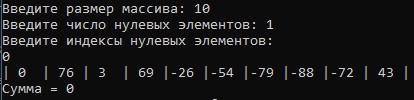
}

**2.4 Тестирование**

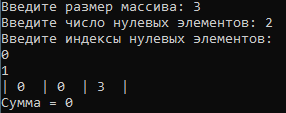
В массиве без нулей сумма будет = 0:



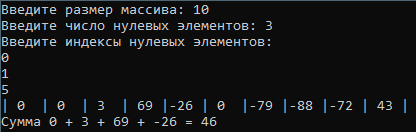
Если единственный ноль будет под индексом 0 то результат не меняется:



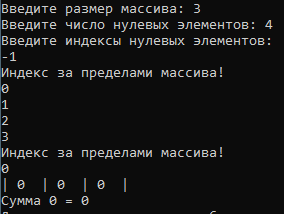
Задав два нуля на соседних ячейках, результат тоже будет 0:



Наконец введём корректные данные и получим первый результат:



И не забудем проверить защиту от выхода за пределы массива:



**3 Решение задачи 2**

**3.1 Формальная постановка задачи**

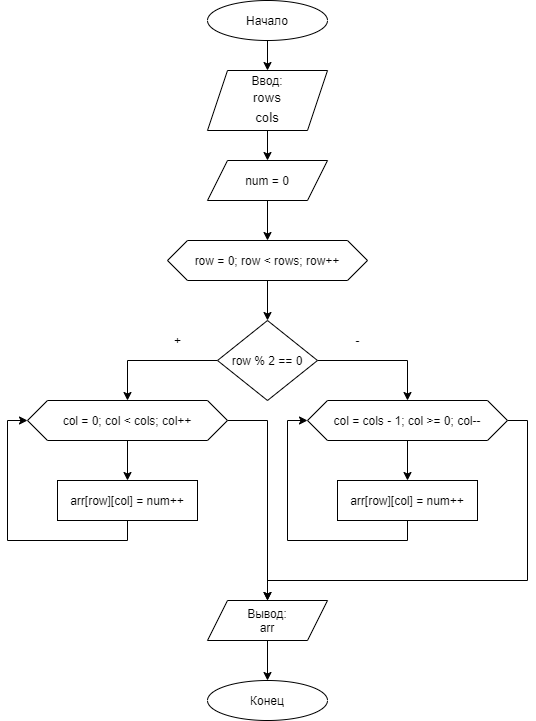
Входные данные:

rows – целое число, отвечает за число строк в массиве;

Выходные данные:

cols – целое число, влияет на число столбцов в массиве;

**3.2 Проектирование**



**3.3 Реализация**

Запросим у пользователя размер двумерного массива:

cout << "Введите число строк: ";

cin >> rows;

cout << "Введите число столбцов: ";

cin >> cols;

Далее во вложенном цикле будем инициализировать строки, а во внешнем определять направление записи:

for (int row = 0; row < rows; row++)

{

arr[row] = new int[cols];

if (row % 2 == 0)

{

// Если чётная строка: слева на право

for (int col = 0; col < cols; col++)

{

arr[row][col] = num++;

}

}

else

{

// Если не чётная строка: справа на лево

for (int col = cols - 1; col >= 0; col--)

{

arr[row][col] = num++;

}

}

}

Остаётся только вывести массив на экран в очередном вложенном цикле выводим строку, а во внешнем осуществляем переход между строками, выводя разделительные линии:

// Верхняя граница таблицы

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

cout << "-----";

}

cout << "-\n"; // Хвост..

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

cout << '|'; // Первая граница столбца

// Строка

for (int col = 0; col < cols; col++)

printf("%-4d|", arr[row][col]); // Число и перегородка: "12 |"

cout << '\n'; // Переход на след строку...

// Граница

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

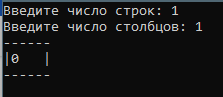
cout << "-----";

cout << "-\n"; // Хвост..

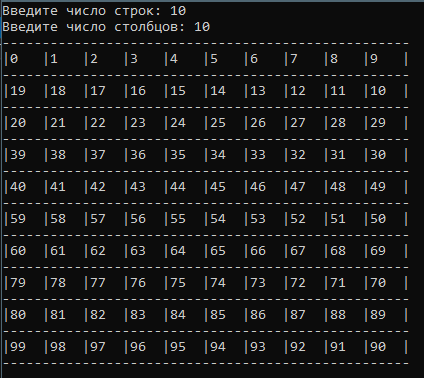
}

**3.4 Тестирование**

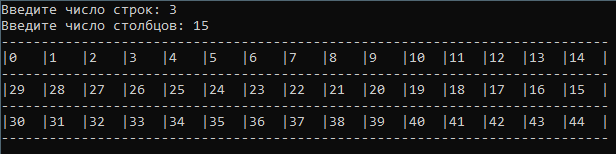
Массив 1 на 1:



Массив 10 на 10:



Ну и напоследок 3 на 15:



**Заключение**

В этой лабораторной работе я на практике применил массивы, закрепил навыки использования условных операторов и циклов, а также воспользовался приоритетом операции постфиксного инкремента сократил код цикла в двое.