

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)  
Кафедра экономической математики, информатики и статистики (ЭМИС)

## МАССИВЫ

Отчёт по лабораторной работе по дисциплине “Введение в  
программирование”

Студент гр. 543-1



/А.Е. Мухамеджан

“ \_\_\_\_ ” \_\_\_\_\_ 2023

Руководитель доцент

кафедры ЭМИС

\_\_\_\_\_ Е.А. Шельмина

“ \_\_\_\_ ” \_\_\_\_\_ 2023г.

Томск 2023

## Лабораторная работа №5

### “Массивы”

Введение: цель работы научиться использовать массивы при решении различных задач.

Теоретические сведения: Описание массива производится с помощью обычного оператора описания, при этом за именем массива в квадратных скобках должна быть записана целая положительная константа или константное выражение, равное размеру этого массива, то есть максимально возможному числу элементов. Имя массива с квадратными скобками, в которых записано индексное выражение целого типа, обозначает значение соответствующего элемента массива. В языке Си нумерация элементов массива начинается с нуля. Индексированные переменные могут использоваться в любых выражениях в тех местах, где допускается применение переменных соответствующих типов. При работе с индексированными переменными необходимо внимательно следить за тем, чтобы индексы не вышли из допустимого диапазона, определяемого описаниями массивов. Дело в том, что компилятор не проверяет факт выхода индексов за границы массива, а при ошибочном занесении данных за пределы массива может запортиться нужная информация и, скорее всего, компьютер зависнет. Язык Си не имеет встроенных средств для ввода-вывода массива целиком, поэтому массив вводят и выводят поэлементно с помощью циклов.

Инициализация массивов: Инициализация - присвоение значений вместе с описанием данных. Ранее была рассмотрена инициализация простых переменных, например: `int a = 5`.

Для инициализации массива за его именем располагают знак присваивания и список инициализации, который представляет собой заключенные в фигурные скобки и разделенные запятыми инициализирующие значения.

Констант в списке инициализации должно быть не больше, чем объявленный размер массива. Если их меньше, то элементы для которых нет констант обнуляются. Для инициализируемого массива допускается вообще не указывать размер. В этом случае размер массива определяется по количеству констант, например по описанию `int c[] = { 1, 15, 18, 11, 20 };` транслятор выделит 10 байт для хранения массива из 5 двухбайтовых целых чисел.

Частный случай инициализации массива - инициализация строк.

Двумерные массивы: Элементом массива может быть в свою очередь тоже массив. Таким образом, мы приходим к понятию двумерного массива или матрицы. Описание двумерного массива строится из описания одномерного путем добавления второй размерности, например: `int a[4][3];`

Анализ подобного описания необходимо проводить в направлении выполнения операций `[]`, то есть слева направо. Таким образом, переменная `a` является массивом из четырех элементов, что следует из первой части описания `a[4]`. Каждый элемент `a[i]` этого массива в свою очередь является массивом из трех элементов типа `int`, что следует из второй части описания.

Имя двумерного массива без квадратных скобок за ним имеет значение адреса первого элемента этого массива, то есть значение адреса первой строки - одномерного массива из трех элементов. При использовании в выражениях тип имени двумерного массива преобразуется к типу адреса строки этого массива. В нашем примере тип имени массива `a` в выражениях будет приведен к типу адреса массива из трех элементов типа `int` и может использоваться во всех выражениях, где допускается использование соответствующего адреса.

Имя двумерного массива с одним индексным выражением в квадратных скобках за ним обозначает соответствующую строку двумерного массива и имеет значение адреса первого элемента этой строки. Имя двумерного массива с двумя индексными выражениями в квадратных скобках за ним обозначает соответствующий элемент двумерного массива и

имеет тот же тип. В соответствии с интерпретацией описания двумерного массива (слева-направо) элементы последнего располагаются в памяти ЭВМ по строкам. Инициализация двумерного массива также проводится по строкам.

В языке Си допускается использовать не только двумерные, но и трехмерные, четырехмерные и т. д. массивы. Их использование ничем принципиально не отличается от использования двумерных массивов, однако на практике они применяются значительно реже.

### Ход работы

Задания для самостоятельной работы:

Задание 1. Написать на языке Си программу, осуществляющую заполнение массива случайными числами из заданного диапазона и выполняющую заданные действия с элементами массива

Условие к заданию 1 представлено на рисунке 1.

- 2 Найти минимальный по модулю элемент в одномерном массиве  $x$ . Затем каждый элемент в массиве разделить на этот минимальный элемент.
- 3 Дан массив  $s(n)$ . Переписать в массив  $x$  ненулевые элементы массива  $s$  разделенные на 5. (сжатие массива)
- 4 Известно, что в массиве  $a(n)$  есть один элемент  $= 2$ . Найти где он находится и вычислить среднее арифметическое элементов стоящих перед ним. (выдать на экран номер элемента и сумму).
- 5 Задан одномерный массив  $a(n)$ / Найти минимальный элемент и переставить его с 2-ым элементом массива.

Рисунок 1 – Скриншот условия к заданию №1

Программа к заданию №1 представлена на рисунках 2,3,4.

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5  #include <locale.h>
6  #include <time.h>
7  int main()
8  {
9      setlocale(LC_CTYPE, "Rus");
10     srand(time(NULL));
11     const int n = 5;
12     int a[n], x[n], a4[n] = {9,5,3,2,1}, count = 0, min = 10, k = 0, indexmin, temp;
13     float sum = 0;
14     for (int i = 0; i < n; i++)
15     {
16         a[i] = -10 + rand() % 10;
17         if (abs(a[i]) < min)
18             min = a[i];
19         printf("%d ", a[i]);
20     }
21     printf("\n Введите номер задания\n");
22     int zadanie;
23     scanf("%d", &zadanie);
24     switch (zadanie)
25     {
26     case 1:

```

Рисунок 2 – Скриншот первой части программы к заданию №1

```

27     for (int j = 0; j < n - 1; j++)
28     {
29         if (a[j] == a[j + 1])
30             count++;
31     }
32     printf("%d\n", count);
33     break;
34     case 2:
35     for (int i = 0; i < n; i++) {
36         a[i] = a[i] / min;
37         printf("%d ", a[i]);
38     }
39     break;
40     case 3:
41     for (int i = 0; i < n; i++)
42     if (a[i] != 0)
43     {
44         x[k] = a[i];
45         k++;
46     }
47     break;
48     case 4:
49     int index;
50     for (int i = 0; i < n; i++) {
51         if (a4[i] == 2) {
52

```

Рисунок 3 – Скриншот второй части программы к заданию №1

```

53         index = i;
54         for (int j = 0; j < i; j++)
55             sum += a4[j];
56         break;
57     }
58 }
59 printf("2 находится на индексе '%d', среднее арифметическое элементов перед 2 = %f", index, sum / index );
60 break;
61 case 5:
62     min = 10;
63     for (int i = 0; i < n; i++) {
64         if (a[i] < min) {
65             min = a[i];
66             indexmin = i;
67         }
68     }
69     temp = a[indexmin];
70     a[indexmin] = a[2];
71     a[2] = temp;
72     for (int i = 0; i < n; i++)
73         printf("%d ", a[i]);
74     break;
75 }
76 }

```

Рисунок 4 – Скриншот последней части программы к заданию №1

Задание 2. Дан двумерный массив  $A(n \times m)$ . Ввести данные с клавиатуры. Вывести матрицу на экран в виде таблицы. Найти сумму и количество положительных элементов матрицы

Программа к заданию 2 представлена на рисунке 5.

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5  #include <locale.h>
6  #include <time.h>
7  int main()
8  {
9      setlocale(LC_CTYPE, "Rus");
10     srand(time(NULL));
11     const int n = 3, m = 3;
12     int count = 0, sum = 0;
13     int A[n][m];
14     printf("Введите по очереди каждый элемент массива на отдельной строке\n");
15     for (int i = 0; i < n; i++)
16     {
17         for (int j = 0; j < m; j++) {
18             scanf("%d", &A[i][j]);
19             if (A[i][j] > 0) {
20                 sum += A[i][j];
21                 count++;
22             }
23         }
24     }
25     printf("Кол-во положительных элементов - %d.\nСумма положительных элементов - %d", count, sum);
26 }

```

Рисунок 5 – Скриншот программы к заданию №2

Вывод: в результате выполнения лабораторной работы получены навыки для использования массивов в Си.