

Основы программирования

Лабораторная работа 4

Тема: Массивы переменных

Цель лабораторной работы

Целями данной работы являются изучение методов создания массива переменных, а также способов их применения.

Краткие теоретические сведения

Массивы переменных

Массив – это линейный тип данных или последовательность ячеек памяти одинакового типа. По сути, *массив* – это группа переменных одного типа, расположенных строго друг за другом. Объявление массива:

```
Тип_данных имя_массива [размерность];
```

Пример:

```
int mas[10]; // массив из 10-ти целых чисел
```

Каждая *переменная* в массиве является самостоятельной единицей под названием *элемент массива*. В примере массив состоит из 10-ти элементов.

Элементы массива

Элемент массива (значение элемента массива) – значение, хранящееся в определенной ячейке памяти, расположенной в пределах массива, а также адрес этой ячейки памяти.

Для доступа к элементу массива в **квадратных скобках** указывают его номер (*индекс*). При этом следует отметить, что в языке C++ индексация элементов массива начинается с нуля:

```
a[0] = 10; a[1] = 15; a[2] = 20;
```

За счет использования индексов при работе с элементами массивов можно применять циклы. К примеру, для обхода массива часто используют цикл **for**:

```
int a[10];  
for(int i=0; i<10; i++)  
    a[i] = i;
```

Стоит помнить, что если массив содержит **N** элементов, то *индекс* последнего элемента массива **N-1**.

Инициализация массива

Если требуется можно *инициализировать* элементы массива сразу после объявления:

```
тип_данных имя_массива[размер] = { значение1, значение2, ... значениеN };
```

К примеру:

```
int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

Если размер массива не указан явно, то он выводится из количества инициализаторов:

```
int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // 10 элементов
```

Используя *унифицированную* инициализацию (*uniform initialization*), можно определить массив так (без оператора «=»):

```
int a[4]{ 1,2,3,4 };
int b[] { 1,2,3 };
```

При этом **не допускается** присвоение одному массиву другого массива. К примеру:

```
int nums1[]{ 1,2,3 };
int nums2[] = nums1;    // ошибка, нельзя так инициализировать
nums2 = nums1;          // ошибка, нельзя так присвоить
```

Можно заполнить элементы массива «вручную» непосредственно в программе, как в примере с циклом **for**. Инициализация массива пользователем запишется в виде:

```
int a[10];
for (int i=0; i<10; i++) {
    cout << "Введите значение a[" << i << "]" << endl;
    cin >> a[i];
}
```

Используя генератор случайных чисел, можно также проинициализировать массив:

```
for (int i=0; i<10; i++) {
    a[i] = rand() % 201 - 100; // случайные числа от -100 до 100
}
```

Аналогичным способом можно вывести содержимое массива на консоль.

```
for (int i=0; i<10; i++) {
    cout << a[i] << " "; // вывод элементов через пробел
}
```

Пример присвоения значений одного массива другому:

```
int nums1[3]{ 1,2,3 };
int nums2[3];
for (int i=0; i<3; i++)
    num2[i] = num1[i];
```

Пример нахождения количества отрицательных элементов массива:

```
int a[5]{ 1, -2, 3, -4, 5 }, count=0;
for (int i=0; i<5; i++)
    if (a[i]<0) count++;
```

Пример нахождения суммы положительных элементов массива:

```
int a[5]{ 1, -2, 3, -4, 5 }, sum = 0;
for (int i=0; i<5; i++)
    if (a[i]>0) sum = sum + a[i];
```

Использование констант или макросов для размера массива

Константа в качестве размера массива:

```
const int size = 10; // константа определяющая размер массива
int arr [size];
for (int i=0; i<size; i++) {
    a[i] = rand() % 100;
}
```

Данный способ определения размера массива в программе является более предпочтительным по сравнению с использованием *литералов*.

Также при объявлении массива также можно использовать директиву препроцессора **#define** (макрос), чтобы установить его размерность:

```
#define SIZE 10 //определили текст, который вставится в код
int a[SIZE];
for (int i=0; i<SIZE; i++) {
    a[i] = rand() % 100;
}
```

Многомерные массивы

В многомерных массивах каждый элемент описывается **несколькими** характеристиками. Наиболее часто используемым на практике многомерным массивом является **двумерный** массив (матрица). Объявление:

```
тип_данных имя_массива [число_строк] [число_столбцов];
```

Пример:

```
const int row = 3; // строки
const int col = 4; // столбцы
int arr[row][col]; // массив row на col (3x4)
```

Элементы в таком массиве характеризуются **двумя индексами**:

```
int arr[3][3];
```

	0	1	2
0	arr[0][0]	arr[0][1]	arr[0][2]
1	arr[1][0]	arr[1][1]	arr[1][2]
2	arr[2][0]	arr[2][1]	arr[2][2]

Инициализация многомерного массива

По большому счету *многомерные* массивы инициализируются так же, как и *одномерные*. Но при этом каждая строка заключается в отдельные фигурные скобки:

```
int array[2][2] = { { 1, 2 }, { 7, 8 } };
```

Инициализация этого *двумерного* массива случайными числами:

```
int arr[2][3];
for( int i = 0; i < 2; i++ )
    for( int j = 0; j < 3; j++ )
        arr[i][j] = rand() % 90 + 10;
```

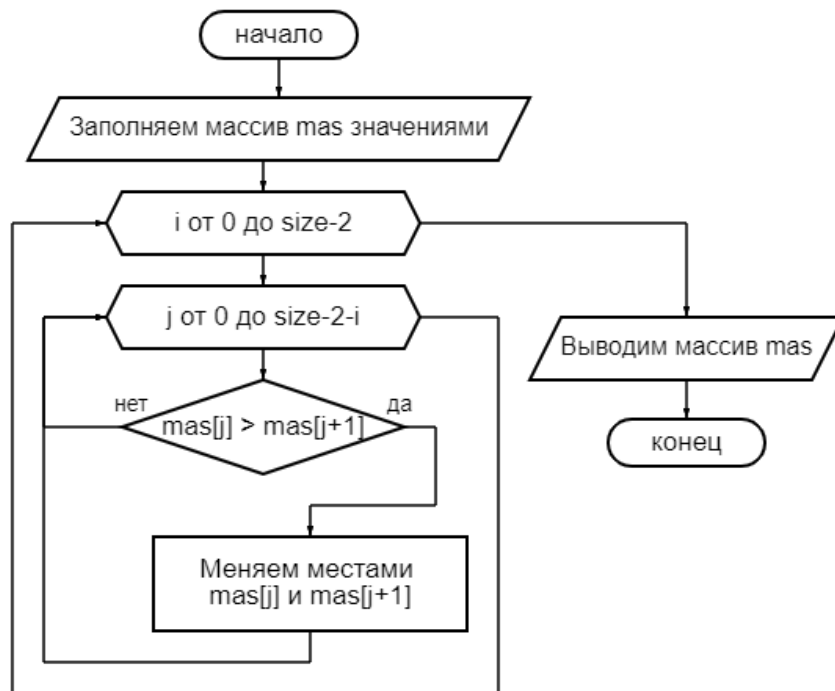
Вывод данного массива:

```
for( int i = 0; i < 2; i++ )
{
    for( int j = 0; j < 3; j++ )
        cout << arr[i][j] << " ";
    cout << endl;
}
```

```
КОНСОЛЬ ОТЛАДКИ
51 27 44
50 99 74
```

Алгоритм сортировки пузырьком

Этот метод – самый простой метод сортировки массива. Рассмотрим пример алгоритма сортировки по возрастанию для одномерного массива *mas* размером *size*:



Запишем алгоритм кодом:

```
const int size = 10; // размер массива
int mas[size];
// ввод массива
for (int i=0; i<size; i++)
    cin >> a[i];
// сортировка
for (int i=0; i<=size-2; i++)
    for (int j=0; j<=size-2-i; j++)
        if(mas[j] > mas[j+1])
        {
            int t = mas[j];
            mas[j] = mas[j+1];
            mas[j+1] = t;
        }
// вывод массива
for (int i=0; i<size; i++)
    cout << a[i] << " ";
```

Задание

Разработать программу для каждой задачи из **общего** и **индивидуального** задания.

Необходимо выводить массивы на экран сразу после заполнения. Также если массивы изменялись в ходе работы программы, также необходимо выводить их измененные версии на экран для демонстрации.

Для указания размерности массивов используйте целочисленные **константы** или **макросы** (к примеру, `const int size = 10;`). Если вместо конкретных размеров массива указаны размеры в виде символов (N или NxM), в этом случае определите размеры самостоятельно.

Также можно выполнить **дополнительное** задание, чтобы получить больше баллов.

Методические указания

Выполните **общие** и **индивидуальные** задания.

При выполнении **общих** заданий не требуется отчет, но все еще нужна защита с объяснением кода программы.

При выполнении **индивидуальных** заданий необходимо выполнить следующие этапы для каждой задачи отдельно:

- 1) словесная постановка задачи;
- 2) анализ задачи и формальная постановка задачи;
- 3) проектирование (разработка алгоритма) – лучше всего в графической форме (блок-схема), но можно и в словесной форме или псевдокод;
- 4) реализация (кодирование, отладка);
- 5) тестирование.

Результаты выполнения **индивидуальных** заданий оформить в виде отчета.

Общее задание

Задание 1

Написать программу, которая считывает массив целых чисел из консоли. Программа сперва выводит все *положительные* элементы, затем все *отрицательные*.

```
Введите 10 чисел:  
1 2 -3 4 5 -6 -7 -8 9 10  
-----  
Положительные:  
1 2 4 5 9 10  
Отрицательные:  
-3 -6 -7 -8
```

Задание 2

Отсортируйте методом пузырька одномерный массив, состоящий из N целочисленных элементов, по возрастанию. Массив должен быть заполнен случайными числами от 0 до 100.

Варианты индивидуальных заданий

Вариант 1

1) Дана температура воздуха за две недели некоторого месяца (14 дней). Определить:

- среднюю температуру за *первую* и *вторую* недели **отдельно**;
- сколько раз температура воздуха опускалась ниже *указанной* температуры (ее вводит пользователь).

2) В двумерном массиве хранится информация о количестве жильцов каждой квартиры одного подъезда **пятиэтажного** дома (на каждом этаже по **четыре** квартиры). Нумерация квартир начинается с 1. То есть последний номер квартиры – 20.

По выбранному номеру квартиры определить количество жильцов. Следите, чтобы номер квартиры был правильным.

Вариант 2

1) Пользователь вводит прибыль фирмы за год (12 месяцев). Необходимо определить месяц, в котором прибыль была **максимальна** и месяц, в котором прибыль была **минимальна**.

2) Напишите программу, заполняющую двумерный массив $N \times M$ целыми числами, начиная с 1, по возрастанию змейкой.

К примеру, если массив 3×10 , то первая строка будет с 1 до 10, вторая – с 20 до 11, третья – с 21 до 30:

1	2	3	4	5	6	7	8	9	10
20	19	18	17	16	15	14	13	12	11
21	22	23	24	25	26	27	28	29	30

Вариант 3

1) В массиве, состоящем из N целых элементов, вычислите сумму элементов массива, расположенных между *первым* и *последним* элементами, равными 0.

Если нулевых элементов нет, или есть только один нулевой элемент, то сумма равна 0.

2) Дана *матрица* целых чисел размером $N \times M$. Найдите её **наибольший** элемент матрицы. После этого найти две суммы: элементов в **столбце** и элементов в **строке**, на пересечении которых находится **наибольший** элемент.

Вариант 4

1) Пользователь вводит прибыль фирмы за год (12 месяцев). Необходимо определить месяц, в котором прибыль была **максимальна** и месяц, в котором прибыль была **минимальна**.

2) Дана целочисленная матрица $N \times M$. Вычислите и запомните сумму положительных элементов каждой **столбце** матрицы в отдельном одномерном массиве размером M .

Выведите матрицу, а под ней массив с результатами подсчета.

Вариант 5

1) Дана группа студентов, и каждый студент имеет определенную оценку за свою работу. Студентов в группе может быть от 5 до 20. Здесь вам необходимо использовать массив максимального размера, однако заполнять только определенное количество элементов – то, что введет пользователь (число от 5 до 20).

Нужно определить **среднюю оценку** студентов по группе. Также нужно определить, сколько студентов имеют оценку **выше средней**.

2) В двумерном массиве целых чисел посчитать сумму элементов:

- в каждой строке;
- в каждом столбце;
- одновременно по всем строкам и всем столбцам.

Массив должен быть заполнен случайными числами от 0 до 100. Оформить вывод результата следующим образом:

3	5	6	7		21
12	1	1	1		15
0	7	12	1		20

15	3	19	9		56

Вариант 6

1) Написать программу, которая выводит массив в **обратном** порядке, а также считает **сумму** и **произведение** всех элементов.

2) Дан двумерный массив размерностью $N \times M$. Заполняется случайными числами от -10 до 10. Необходимо найти отдельно количество *отрицательных*, *положительных* и *равных 0* элементов массива.

Вариант 7

1) В одномерном массиве, состоящем из N целых элементов, вычислите сумму модулей элементов массива, расположенных до последнего положительного элемента (если все элементы положительные, то сумму всех элементов, если все элементы отрицательные, то сумма = 0).

2) Дана матрица целых чисел $A(N, N)$. Перепишите элементы её главной диагонали в одномерный массив $Y(N)$ и разделите их на максимальный элемент **главной диагонали** (результат будет вещественным).

Вариант 8

1) В одномерном массиве, состоящем из N вещественных элементов, вычислите произведение элементов массива, расположенных между первым и последним отрицательными элементами. Если отрицательных элементов нет или между первым и последним отрицательными элементами нет других элементов, то произведение вычислить

нельзя (выведите соответствующее сообщение). Если есть только 1 отрицательный элемент, то произведение вычисляется со следующего за ним элемента и до конца массива.

2) Дана матрица целых чисел $A(N, M)$. Найдите в каждой **строке** минимальный элемент и запишите все найденные элементы в массив $B[N]$.

Вариант 9

1) В одномерном массиве, состоящем из N вещественных элементов, вычислите сумму элементов массива, расположенных между максимальным положительным элементом и максимальным отрицательным элементом. Если между ними нет других элементов, то сумма $= 0$.

2) Дана целочисленная матрица $A(N, M)$. Вычислите сумму, произведение и среднее арифметическое нечётных элементов матрицы, удовлетворяющих условию $|a_{ij}| > i+j$.

Вариант 10

1) В одномерном массиве, состоящем из N вещественных элементов, найдите произведение **отрицательных** элементов массива, расположенных до минимального элемента. Если минимальный элемент – первый, или отрицательных элементов нет, то произведение вычислить нельзя (выведите соответствующее сообщение).

2) Вычислите матрицы $S(N, M)$ и $R(N, M)$, являющиеся суммой и разностью целочисленных матриц $A(N, M)$ и $B(N, M)$.

Вариант 11

1) В одномерном массиве, состоящем из N целых элементов, вычислите сумму элементов массива, расположенных между **первым** и **вторым** отрицательными элементами. Если отрицательных элементов нет или между первым и вторым отрицательными элементами нет других элементов, то сумма $= 0$. Если есть только 1 отрицательный элемент, то вычисляется сумма всех элементов после него.

2) Дана целочисленная матрица $A(N, N)$. Вычислите сумму и количество положительных элементов матрицы, находящихся **под главной диагональю**.

Вариант 12

1) В одномерном массиве, состоящем из N вещественных элементов, вычислите среднее значение (оно равно (максимальный элемент + минимальный элемент) / 2) и количество элементов массива больше чем вычисленное среднее значение.

2) Дана целочисленная матрица $A(N, N)$, где N – четное число. Поменяйте местами элементы четных и нечетных **столбцов**.

Вариант 13

1) В одномерном массиве, состоящем из N целых элементов, вычислите сумму модулей элементов массива, расположенных до первого нулевого элемента, и сумму после него. Если нулевой элемент идет первым, то первая сумма равна 0. Если нулевой элемент идет

последним, то вторая сумма = 0. Если нулевого элемента нет в массиве, то выведите сообщение, что суммы вычислить не удалось.

2) Дана целочисленная матрица $A(N, N)$. Вычислите произведение элементов матрицы, находящихся **над главной диагональю**, и произведение элементов матрицы, находящихся **под главной диагональю**. При вычислении произведений нулевые элементы не учитываются.

Вариант 14

1) Отрадите одномерный массив, состоящий из N целых элементов (то есть первый элемент становится последним, а последний – первым, и так со всеми элементами). Отразить необходимо **с помощью изменения** массива, а не просто вывести в обратном порядке.

2) Дана целочисленная матрица $A(N, N)$. Запишите на место отрицательных элементов матрицы нули, а на место положительных – единицы. Также выведите сообщение о том, каких элементов больше, положительных или отрицательных.

Вариант 15

1) Пользователь вводит номер элемента массива. В одномерном массиве, состоящем из N целых элементов, вычислите максимальный и минимальный элементы для каждой части массива: первая – элементы до введенного номера, вторая – элементы после (то есть введенный номер делит массив на 2 части). Если номер неправильный (меньше 0 или больше $N-1$), то вместо поисков вывести сообщение об ошибке. Если номер = 0, то для **первой** половины вывести сообщение, что элементов в ней нет, а для **второй**, как и задумано, вычислить максимальный и минимальный элементы. Если номер = $N-1$, то для **второй** половины вывести сообщение, что элементов в ней нет, а для **первой**, как и задумано, вычислить максимальный и минимальный элементы.

2) Дана целочисленная матрица $A(N, N)$. Выведите на экран **нижнюю** треугольную матрицу (все элементы выше главной диагонали вывести равными нулю) и **верхнюю** треугольную матрицу (все элементы ниже главной диагонали вывести равными нулю).

Вариант 16

1) В одномерном массиве, состоящем из N вещественных элементов, вычислите среднее арифметическое отдельно элементов массива, расположенных после **минимального по модулю** элемента и до него (минимальный элемент не учитывается). Если минимальный элемент первый или последний, то вычислите только одно среднее арифметическое и выведите сообщение о том, что не смогли вычислить другое.

2) Дана целочисленная матрица $A(N, N)$. Матрица заполнена числами от 1 до $N*N$ по порядку. Программа загадывает число от 1 до $N*N$. Пользователь должен его угадать, вводя число. Количество попыток ограничено 10-ю. После каждой попытки пользователю даются подсказки «выше» или «ниже», «левее» или «правее» находится загаданное число в матрице.

Дополнительное задание

Трехмерный массив описывает журнал отметок студентов за время обучения. Здесь используются три измерения:

- 1) номер семестра;
- 2) номер студента;
- 3) номер дисциплины.

Предположим, что у нас всего 10 семестров (по 2 семестра за 5 курсов), 20 студентов в группе и по 5 дисциплин в каждом семестре. Выходит массив:

```
int journal [10][20][5];
```

Каждое значение этого массива – это оценка определенного студента за определенную дисциплину в определенном семестре. Заполните массив журнала случайными оценками от 2 до 5.

Выведите номера студентов, которые идут на красный диплом (то есть все оценки 4 и 5), а также отдельно номера студентов, которые идут на отчисление (если есть хотя бы одна оценка «2» в любом из семестров).

Помимо этого пользователь может ввести номер семестра. Необходимо вывести журнал за этот семестр, а также определить, сколько студентов в этом семестре идут на стипендию (все оценки выше 3).