

# Лабораторная работа №6

## Обработка статических массивов

**ЦЕЛЬ РАБОТЫ:** приобретение практических навыков в составлении программ по обработке статических массивов.

### Теоретический материал

#### Основные понятия

Массив - структурированный тип данных с элементами одного и того же типа, имеющий одно имя и определенное количество элементов, расположенных в памяти строго последовательно. Количество элементов определяет размер массива. Порядковый номер элемента массива называется его индексом. Число индексов называется размерностью массива, например, массив с двумя индексами называется двумерным массивом. Вектор – одномерный массив элементов, матрица – массив векторов.

Обработка массивов выполняется обычно следующим образом: объявление, ввод или инициализация элементов массива, преобразование и вывод.

#### Объявление массива

Объявить массив – выделить место в памяти компьютера, объём которой зависит от количества элементов и типа массива. Тип массива - это тип входящих в него элементов. Массивы могут быть разных типов: int, float, char, и т.д. Массив объявляют так же, как и обычные переменные, но после имени массива в квадратных скобках записывается его размер.

```
int A[10], B[20]; // 2 массива по 10 и 20 целых чисел  
float C[12]; // массив из 12 вещественных чисел
```

При объявлении массива можно сразу заполнить его начальными значениями (проинициализировать), перечисляя их внутри фигурных скобок:

```
int A[4] = { 2, 3, 12, 76 };
```

Если в списке в фигурных скобках записано меньше чисел, чем элементов в массиве, то оставшиеся элементы заполняются нулями. Если чисел больше, чем надо, компилятор сообщает об ошибке. Например,

```
int A[4] = { 2 }; // последние три элемента равны 0  
int A[4] = { 2, 3, 12, 76, 0 }; // ошибка, должно быть не более 4-х элементов
```

Для повышения универсальности программы размер массива лучше определять через константу. В этом случае для переделки программы для массива другого размера надо только поменять значение этой константы:

```
const int N = 20; //N - константа
main()
{
int A[N]; // размер массива задан через константу
...
}
```

### Обращение к элементу массива

Каждый элемент массива имеет свой порядковый номер. Чтобы обратиться к элементу массива, надо написать имя массива и затем в квадратных скобках номер нужного элемента. Важно запомнить правило: элементы массивов в языке Си нумеруются с нуля. Поэтому индекс последнего элемента массива на 1 меньше числа элементов в данном массиве.

Таким образом, если в массиве 10 элементов, он содержит элементы:

**A[0], A[1], A[2], ..., A[9]**

Номер элемента массива также называется его **индексом**. Вот примеры обращения к массиву A:

**x = (A[3] + 5)\*A[1]; // прочитать значения A[3] и A[1]**

**A[0] = x + 6; // записать новое значение в A[0]**

В языке Си не контролируется **выход за границы массива**, то есть формально вы можете записать что-то в элемент с несуществующим индексом, например в **A[345]** или в **A[-12]**. Однако при этом вы стираете какую-то ячейку в памяти, не относящуюся к массиву, поэтому последствия такого шага непредсказуемы и во многих случаях программа «зависает» или «падает» с ошибкой по памяти.

### Ввод с клавиатуры и вывод на экран

Существует много способов ввода в зависимости от вашей задачи:

- элементы массива вводятся с клавиатуры вручную;
- массив заполняется случайными числами (например, для моделирования случайных процессов);
- элементы массива читаются из файла;
- массив заполняется в процессе вычислений.

Чтобы ввести массив в память, надо каждый его элемент обработать отдельно (например, вызвав для него функцию ввода **scanf**). Для ввода и вывода массива обычно используется цикл **for**.

## Пример обработки одномерного массива

Дан массив из 50 целых чисел. Найти наибольший элемент в массиве и его порядковый номер.

```
# include <stdio.h>
#include <conio.h>
# define n 50    // определение константы n=50
int  i, max, nom, a[n];    //описание массива целых чисел из  n элементов
main()
{ for (i=0; i<n; i++)
  { printf( "\n Введите элемент массива  ");
    scanf ("%d", &a[i]) };
  for (i=1,max=a[0],nom=0; i<n; i++)
    if (max<a[i])
      {nom=i; max=a[i];}
  printf("\n Вывод элементов исходного массива : \n");
  for (i=0; i<n; i++) printf ( "%6d", a[i] );
  printf ("\n Максимальное число в массиве %4d, его индекс %4d  " , max,
nom+1);
  getch();
}
```

## Заполнение случайными числами

Пусть требуется заполнить массив равномерно распределенными случайными числами в интервале **[a,b]**. Поскольку для целых и вещественных чисел способы вычисления случайного числа в заданном интервале отличаются, рассмотрим оба варианта.

Описание функции-датчика случайных чисел находится в заголовочном файле **stdlib.h**. Получить случайные числа с равномерным распределением в интервале **[0,N-1]**:

**rand() % N;**

Для получения случайных целых чисел с равномерным распределением в интервале **[a,b]** надо использовать формулу

**k = rand % ( b – a + 1 ) + a;**

Для вещественных чисел формула несколько другая:

**x = rand()\*(b - a)/RAND\_MAX + a;**

Здесь константа **RAND\_MAX** – это максимальное случайное число, которое выдает стандартная функция **rand**.

В приведенном ниже примере массив A заполняется случайными целыми числами в интервале [-5,10], а массив X - случайными вещественными числами в том же интервале.

```
#include <stdlib.h>
#include <stdio.h>
const int N = 10;
int random (int N) { return rand()%N; }
main()
{ system("CLS");
  int i, A[N], a = -5, b = 10;
  float X[N];
  for ( i = 0; i < N; i ++ )
    A[i] = random(b-a+1) + a;
  for ( i = 0; i < N; i ++ )
    X[i] = (float)rand()*(b-a)/RAND_MAX + a;
  printf("\nЦелые\n");
  for ( i = 0; i < N; i ++ )
    printf("%4d",A[i]);
  printf("\nВещественные\n");
  for ( i = 0; i < N; i ++ )
    printf("%6.2f",X[i]);
  system("PAUSE");
  return 0;
}
```

**Многомерные массивы** задаются указанием каждого измерения в квадратных скобках. Например, оператор

```
int matr [6] [8];
```

задает описание матрицы из 6 строк и 8 столбцов. Нумерация строк и столбцов начинается с 0. При инициализации двумерного массива он представляется как массив из массивов, при этом каждый массив заключается в свои фигурные скобки, либо задается общий список в том порядке, в котором элементы располагаются в памяти:

```
int mas [] [2]= { {1,2}, {0,2}, {1,0}};
int mas [3][2]={1,2,0,2,1,0};
```

Если задать не все элементы, то остальные заполнятся нулями:

```
float X[2][3] = {{1., 3.},{6.}};
```

Здесь элементы  $X[1][2]$ ,  $X[2][1]$  и  $X[2][2]$  будут нулевыми.

### **Расположение двумерных массивов в памяти.**

Иногда, бывает полезно знать, как матрицы располагаются в памяти ЭВМ. Оказывается, во всех современных языках программирования (кроме Фортрана) элементы матрицы располагаются по строкам, то есть сначала изменяется последний индекс. Объявленная выше матрица  $X$  расположена так:

$X[0][0]$   $X[0][1]$   $X[0][2]$   $X[1][0]$   $X[1][1]$   $X[1][2]$

### **Ввод/вывод многомерных массивов**

Единственная проблема состоит в том, чтобы не перепутать переменные в двух циклах и пределы их изменения.

```
const int M = 5; // число строк
const int N = 4; // число столбцов
main() {
    int i, j, A[M][N];
    for ( i = 0; i < M; i ++ ) // цикл по строкам
        for ( j = 0; j < N; j ++ ) // цикл по столбцам строки
        {
            printf ("A[%d][%d]=", i, j); // подсказка для ввода
            scanf ("%d", & A[i][j]); // ввод A[i][j]
        }
    // работа с матрицей
}
```

Заметьте, что при изменении порядка циклов (если поменять местами два оператора `for`) изменится и порядок ввода элементов в память.

При выводе матрицы ее элементы желательно расположить в привычном виде – по строкам. Напрашивается такой прием: вывели одну строку матрицы, перешли на новую строку экрана, и т.д. Надо учитывать, что для красивого вывода на каждый элемент матрицы надо отвести равное количество символов (иначе столбцы будут неровные). Делается это с помощью форматирования – цифра после знака процента задает количество символов, отводимое на данное число.

```
printf("Матрица A\n");
for ( i = 0; i < M; i ++ )
{ // цикл по строкам
    for ( j = 0; j < N; j ++ ) // вывод одной строки (в цикле)
```

```

        printf ( "%4d", A[i][j] ); // 4 символа на число
    printf("\n"); // переход на другую строку
}

```

## Задание


**Задача А. Обработка векторов. Значения в массив вводятся с клавиатуры (если другое не указано в задании).**

1. В данном массиве  $A[N*2]$  поменять местами элементы, стоящие на нечетных местах, с элементами, стоящими на четных местах.
2. Задан массив  $A$  из  $N$  чисел. Преобразовать этот массив так, чтобы сначала шли четные числа, а потом – нечетные.
3. Задан массив  $A$  из  $N$  чисел. Найти в этом массиве первый 0 и забить все оставшиеся числа 0, т.е., если массив  $A$  был (2,1,0,4,8,0,5), то должны получить массив  $A$  (2,1,0,0,0,0,0).
4. Задан массив  $A$  из  $N$  чисел. Образовать массив  $B$  из индексов чисел массива  $A$ , делящихся нацело на 2.
5. Задан массив  $A$  из  $N$  чисел.  $N$  – кратно 3. Это надо проверить. Образовать массив  $B$  из сумм каждой тройки чисел массива  $A$ .
6. Задан массив  $A$  из  $N$  чисел. Удалить из этого массива все 0, т.е., если массив  $A$  был (1,0,2,0,4), то получим новый массив  $B$  (1,2,4).
7. Задан массив  $A$  из  $N$  чисел и некоторое число  $B$ . Найти в массиве  $A$  все числа, равные  $B$ , подсчитать их количество. Преобразовать массив  $A$ , поставив вслед за каждым числом, равным  $B$ , второе такое же число. Если массив  $A$  был (1,2,4) и  $B = 2$ , то получим новый массив  $C$  (1,2,2,4).
8. Задан массив  $A$  из  $N$  чисел. Убрать из массива  $A$  повторяющиеся числа, стоящие рядом, т.е., если массив  $A$  был (2,0,0,1,4,4,3,5,5,5), то получим новый массив  $C$  (2,0,1,4,3,5).
9. Задан массив  $A$  из  $N$  чисел. В массиве  $A$  рядом могут стоять одинаковые числа. Образовать массив  $B$ , подсчитав количество повторяющихся чисел в массиве  $A$ , стоящих рядом. Например, массив  $A$  был (2,0,0,1,4,4,3,5,5,5), тогда массив  $B$  будет (1,2,1,2,1,3).
10. Задан массив  $A$  из  $N$  чисел. Задан массив  $B$  из  $M$  чисел. Известно, что все числа из массива  $B$  положительные, причем любое число больше или равно 0 и меньше  $N-1$ , т.е. числа из массива  $B$  задают индексы чисел из массива  $A$ . Индексы могут повторяться. Найти сумму чисел из массива  $A$ , индексы которых заданы в массиве  $B$ .
11. Задан массив  $A$  из  $N$  чисел. Создать массив  $B$ , в котором будут находиться все числа, встречающиеся в массиве  $A$  только 1 раз.
12. Задан массив  $A$  из  $N$  чисел.  $N$  – не обязательно число, кратное 3. Убрать из этого массива каждый третий элемент – образовать новый массив без каждого третьего числа из массива  $A$ .
13. Задан массив  $A$  из  $N$  чисел. Создать массив  $B$ , в котором будут записаны числа из массива  $A$ , и массив  $C$ , в котором  $C_i$  равно

количеству чисел  $V_i$  в массиве А. Например, массив А (1,0,0,2,0,3,2,-4,1). Массив В должен быть (1,0,2,3,-4), а массив С (2,3,2,1).

14. Задан массив А из N чисел. N – четное число. Образовать массив В, в котором  $V_0 = \max(A_0, A_1)$ ,  $V_1 = \max(A_2, A_3)$  и т.д.
15. Задан массив А из N чисел. Образовать массив В, в котором каждый элемент из массива А повторяется столько раз, сколько его значение. Если массив А (2,0,3,1), то новый массив В (2,2,3,3,3,1).
16. (\*)Задан массив А из N чисел. Выбрать из этого массива любое число и сделать так, чтобы все числа левее его были его меньше, а правее – больше. Алгоритм: берем любое число из массива А –  $A_j$ . Начинаем сравнивать это число со всеми числами, находящимися левее его, причем начинаем сравнение с самым первым числом, потом со вторым и т.д. Если  $A_i$  меньше  $A_j$ , то меняем их местами. Теперь мы знаем, что все числа, располагающиеся левее  $A_i$ , меньше интересующего нас числа. Начинаем сравнивать число  $A_j$  с числами, располагающимися правее этого числа, причем сравнение производим с конца. Если какое-то число оказывается меньше  $A_i$ , то меняем их местами. Теперь мы знаем, что все числа, находящиеся правее этого числа, больше его. Но ничего не знаем о числах, находящихся левее этого числа до места предыдущего положения этого числа. Вновь повторяем сравнения с числами, располагающимися левее, но не с нулевого элемента, а с элемента, который стоит на месте после предыдущего размещения числа т.д. Например, пусть задан массив А(1,2,3,8,9,12,4,2,6,1,9). Возьмем число 12. Его индекс 5. Начнем сравнивать это число с числами левее его. Все числа меньше 12. Теперь начинаем сравнивать с числами правее его. Число 9 меньше 12. Его индекс 10. Меняем местами 5-ый и 10-ый элементы. Получим массив А(1,2,3,8,9,9,4,2,6,1,12). Начинаем сравнивать число 12, стоящее теперь на 10 месте, с числами левее его, начиная с 6-ого элемента. Все числа меньше 12. Второй пример. Возьмем число 8. Его индекс 3. Все числа левее его, меньше его. Начинаем сравнивать с числами, находящимися правее. Нужно поменять местами число 8 и число 1, находящееся по индексу 9. Получим А(1,2,3,1,9,12,4,2,6,8,9). Опять сравниваем число 8, находящееся теперь по индексу 9, с числами левее его, начиная с числа по индексу 4. Поскольку 9 больше 8, то меняем их местами. Получим: А(1,2,3,1,8,12,4,2,6,9,9). Сравниваем 8, находящееся теперь по индексу 4 с числами правее его, начиная с числа, находящегося по индексу 8. Опять меняем местами. Получим: А(1,2,3,1,6,12,4,2,8,9,9). Опять сравниваем число 8, находящееся по индексу 8, с числами левее его, начиная с числа по индексу 5. Опять меняем их местами. Получим А(1,2,3,1,6,8,4,2,12,9,9). Сравниваем 8, находящееся по индексу 5, с числами правее его, начиная с индекса 7 и т.д. до тех пор пока сравнивать слева и справа будет не с чем.

**Задача Б. Обработка многомерных массивов. Значения в массиве генерируются случайным образом (если другое не указано в задании).**

1. В массиве  $A[1..N, 1..N]$  определить номера строки и столбца какой-нибудь седловой точки. Некоторый элемент массива называется седловой точкой, если он является одновременно наименьшим в своей строке и наибольшим в своем столбце.
2. Массив  $A[1..5, 1..7]$  содержит вещественные числа. Требуется ввести целое число  $K$  и вычислить сумму элементов  $A[I, J]$ , для которых  $I+J=K$ . Прежде, однако следует убедиться, что значение  $K$  позволяет найти решение, в противном случае нужно напечатать сообщение об ошибке.
3. Дан массив  $A[1..N, 1..N]$ . Составить программу, которая прибавила бы каждому элементу данной строки элемент, принадлежащий этой строке и главной диагонали.
4. Дана матрица  $N \times M$ . Переставляя ее строки и столбцы, переместить наибольший элемент в верхний левый угол. Определить можно ли таким же образом поместить минимальный элемент в нижний правый угол.
5. Дан двумерный массив  $A[1..m, 1..n]$ . Реализовать поиск минимального из максимальных элементов каждой строки матрицы и такой же поиск для столбцов.
6. Элемент двумерного массива называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером  $N \times N$  найти максимум среди всех локальных минимумов.
7. Дан двумерный массив  $A[1..m, 1..n]$ . Написать программу построения одномерного массива  $B[1..m]$ , элементы которого соответственно равны а) суммам элементов строк, б) произведениям элементов строк
8. Дана действительная квадратная матрица порядка  $n$ . Найти наибольшее из значений элементов, расположенных в заштрихованной части матрицы, выполнить для каждого из вариантов:  

9. Даны два числа  $k_1$  и  $k_2$  и матрица  $A[1..m, 1..n]$ . Поменять местами столбцы матрицы с номерами  $k_1$  и  $k_2$ , а потом поменять соответствующие строки.
10. Дан двумерный массив  $A[1..m, 1..n]$ . Реализовать поиск максимального из минимальных элементов каждой строки матрицы и такой же поиск для столбцов.



11. Дана действительная квадратная матрица порядка  $N$  ( $N$  — нечетное), все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей.
12. Расположить столбцы матрицы  $D [M, N]$  в порядке возрастания элементов  $k$ -ой строки.
13. Заполнить матрицу  $A[N, N]$  последовательностью от 1 до  $N*N$  в виде спирали и вывести на экран:

```
1 2 3
8 9 4
7 6 5
```

Заменить элементы побочной диагонали на случайные числа.

14. Задана квадратная матрица. Получить транспонированную матрицу, т.е. матрицу, где столбцы и строки меняются местами.
15. Для заданной квадратной матрицы сформировать одномерный массив из ее диагональных элементов. Найти след матрицы, суммируя элементы одномерного массива. Преобразовать исходную матрицу по правилу: четные строки разделить на полученное значение, нечетные оставить без изменения.
16. (\*)“Тестирование коллектива”. Пусть целочисленная матрица размером  $n \times m$  содержит информацию о студентах некоторой группы из  $n$  человек. В первом столбце проставлена масса (кг), во втором — рост (см), в третьем — успеваемость (средний балл) и т.д. (используйте свои дополнительные показатели). Студент называется среднестатистическим по  $k$ -му параметру (уникальным по  $k$ -му параметру), если на нем достигается минимум (максимум) модуля разности среднего арифметического чисел из  $k$ -го столбца и значения  $k$ -го параметра этого студента. Студент называется самым уникальным (самым средним), если он уникален (является среднестатистическим) по самому большому количеству параметров. По данной матрице определить самых уникальных студентов и самых средних.