Урок 5. Массивы

Цель урока: изучить определение, назначение, виды и описание массивов, приобрести навыки в решении задач с помощью массивов.

Массив — это структура однотипных элементов, занимающих непрерывную область памяти. С массивом связаны следующие его свойства: имя, тип, размерность, размер. Формат описания массива следующий:

```
тип элементов имя  [константное выражение]
```

Константное выражение определяет размер массива, т. е. число элементов этого массива. Например, согласно описанию

```
int A[10];
```

объявлен массив с именем A, содержащий 10 элементов целого типа. Элементы массива обозначаются индексированными именами. Нижнее значение индекса равно 0:

```
A[0], A[1], A[2], A[3], A[4], A[5], A[6], A[7], A[8], A[9]
```

В C/C++ нельзя определять произвольные диапазоны для индексов. Размер массива, указанный в описании, всегда на единицу больше максимального значения индекса.

Размер массива может явно не указываться, если при его объявлении производится инициализация значений элементов. Например:

```
int p[] = {2, 4, 6, 10, 1};
```

В этом случае создается массив из пяти элементов со следующими значениями:

```
p[0] = 2, p[1] = 4, p[2] = 6, p[3] = 10, p[4] = 1
```

В результате следующего объявления массива

```
int M[6] = {5, 3, 2};
```

будет создан массив из шести элементов. Первые три элемента получат инициализированные значения. Значения остальных будут либо неопределенными, либо равны нулю, если массив внешний или статический. Рассмотрим несколько примеров программ обработки одномерных массивов.

Пример 1. Ввод с клавиатуры и вывод на экран одномерного массива.

```
//Ввод и вывод массива

#include  <iostream>
#include <conio.h>

using namespace std;

void main( )
{
int i, A[5];
void clrscr( );
for(i = 0; i < 5; i++)
      {
      cout << " A[" << i << "] = ";  cin >> A[i];}
for ( i = 0;  i < 5;  i++)
      cout << " A[" << i << "] = "<< A[i] <<" ";
      }
```

Пример 2. Ввод вещественного массива и вычисление среднего значения.

```
//Среднее значение массива

#include <iostream>
#include <conio.h>
#include <clocale>

using namespace std;

void  main( )
{
setlocale(LC_ALL,"RUS");
const  n = 10;
int i;
double A[n], SA;
void clrscr( );
for(i = 0; i < n; i++)
{
cout << "A["<< i << "] = "; cin >> A[i];}
SA = 0;
for(i = 0; i < n; i++)
SA = SA + A[i];
SA = SA/n;
cout << "/переднее  значение =" << SA;
}
```

В этой программе обратите внимание на определение размера массива через константу.

Пример 3. Сортировка массива «методом пузырька».

```
//Сортировка массива

#include <iostream>
#include <conio.h>

using namespace std;

void main( )
{
      int X[] = {6, 4, 9, 3, 2, 1, 5, 7, 8, 10};
      int i, j, n, A;
      void clrscr( );
      n = sizeof(X)/sizeof(X[0]) ;
      for(i = 0;  i < n - 1; i++)
      for (j = 0; j < n - 1 - i; j++)
      if (X[j] > X[j + 1])  {A = X[j]; X[j] = X[j + 1]; X[j + 1] = A;}
      for ( i = 0; i < n; i++)
      cout << X [i] << " ";
}
```

В данной программе массив инициализирован. Его размер равен числу заданных значений. Чтобы сделать программу универсальной по отношению к размеру массива, значение размера вычисляется автоматически и заносится в переменную n. Для этого используется операция sizeof( ) — определение размера в байтах. Результат sizeof(X) равен размеру в памяти всего массива x — 20 байтам. Результат sizeof (X[0]) равен размеру одного элемента массива — 2 байтам. Отношение этих величин равно 10 —

числу элементов массива. Внимательно проанализируйте организацию перебора значений параметров вложенных циклов — i , j .
В результате выполнения этой программы на экран выведется упорядоченная числовая последовательность

```
1 2 3 4 5 6 7 8 9  10
```

Многомерные массивы. Двумерный массив трактуется как одномерный массив, элементами которого является массив с указанным в описании типом элементов. Например, оператор

```
float R[5][10];
```

объявляет массив из пяти элементов, каждый из которых есть массив из десяти вещественных чисел. Отдельные величины этого массива обозначаются именами с двумя индексами:

```
R[0][0], R[0][1], ..., R[4][9];
```

Объединять индексы в одну пару скобок нельзя, т.е. запись R[2, 3] ошибочна.
Пример описания трехмерного массива:

```
double X[3][7][20];
```

Порядок расположения элементов многомерного массива в памяти такой, что прежде всего, меняется последний индекс, затем предпоследний и т.д., и лишь один раз пробегает свои значения первый индекс.

При описании многомерных массивов их также можно инициализировать. Делать это удобно так:

```
int M[3][3] = {11, 12, 13, 21, 22, 23, 31, 32, 33};
```

Рассмотрим примеры программ обработки матриц — числовых двумерных массивов.

Пример 4. Вычисление и вывод на экран таблицы умножения в форме матрицы Пифагора.

```
// Матрица Пифагора

#include <stdio.h>
#include <conio.h>

void  main( )
{
int i, j, A[10][10];
void clrscr( );
for(i = 1;  i <= 9; i++)
      {
            for(j = 1; j <= 9; j++)
            {
            A[i][j] = i*j;
            printf("%5d", A[i][j]);
            }
      printf ("\n");
      }
}
```

По данной программе в двумерном массиве A не будут заполнены нулевая строка и нулевой столбец. По-прежнему интерпретируем первый индекс двумерного массива как номер строки матрицы, а второй индекс — как номер столбца.

Пример 5. Заполнение матрицы случайными числами в диапазоне от 0 до 99 и поиск в ней максимального значения.

```
#include <iostream>
#include <iomanip.h>
#include <conio.h>
#include <stdlib.h>
#include <clocale>
#define     n  5

using namespace std;

void  main( )
{
setlocale(LC_ALL,"RUS");
int  i, j, ImaxA, JmaxA, A[n][n] ;
void clrscr( );
randomize( );  //Установка  датчика  случайных  чисел
for(i = 0; i < n; i++)
{
      for(j = 0; j < n; j++)
{
A[i][j] = rand( ) % 100;
cout << setw(6)<< A[i][j];
}
cout << endl;
}
ImaxA = JmaxA = 0;
for( i = 0;  i < n; i++)
{
for (j = 0; j < n; j++)
if(A[i][j] > A[ImaxA][JmaxA])
{
ImaxA = i; JmaxA = j; }
}
cout << "Максимальное_значение:[" << ImaxA << "][" << JmaxA << "] =" <<
A[ImaxA][JmaxA];
}
```

В результате тестирования этой программы получен следующий результат:

```
46  23  57  35  18
8   48  68  4   70
56  98  16  71  40
70  84  66  67  11
20  44  37  57  38
```

Максимальное значение: A[2][1] = 98

В данной программе имеются новые элементы, использование которых требует пояснения. В стандартной библиотеке с заголовочным файлом stdlib.h содержится функция, прототип которой имеет вид: int rand( ).

Результатом этой функции является целое случайное число из диапазона от 0 до RAND_MAX. Значение константы RAND_MAX определено в заголовочном файле

stdlib.h и обычно равно 32767 — максимально допустимому целому числу. Для получения случайных чисел в диапазоне от 0 до N - 1 достаточно вычислить остаток от целого деления rand( ) на N. Функция с прототипом void randomize( ) выполняет первоначальную настройку датчика случайных чисел так, чтобы последовательность чисел не повторялась при повторном выполнении программы.

Другим новым элементом в данной программе является использование манипуляторов для управления потоковым выводом с помощью стандартного объекта cout. Манипуляторы объявляются в заголовочном файле iomanip.h. Манипулятор setw(n) влияет на формат следующего элемента выходного потока. Он указывает на то, что значение будет выводиться в n позиций на экране (в программе n = 6). Другой использованный манипулятор — endl — обозначает конец строки и переводит экранный курсор на начало новой строки. Его действие аналогично действию управляющего символа \n.

Упражнения

1. Дан вектор $\{z_i\}$, $i = 1, \ldots, 50$. Вычислить длину этого вектора:

   $L = \sqrt{z_1^2 + z_2^2 + \cdots + z_{50}^2}$.

2. Вычислить полином 10-й степени по формуле Горнера:

   $a_{10}x^{10} + a_9x^9 + \ldots + a_1x + a_0 = ((\ldots(a_{10}x + a_9)x + a_8)x + \ldots + a_1)x + a_0$.

3. Для вектора $\{ x_i \}$, $i = 1, ..., 20$, подсчитать количество компонент, значения которых лежат в интервале [0, 1].
4. Даны два вектора $\{ x_i \}, \{ y_i \}$, $i = 1, ..., 10$, упорядоченные по возрастанию. Слить их в один вектор $\{ z_i \}$, $i = 1, ..., 20$ так, чтобы сохранилась упорядоченность.
5. Дан массив, состоящий из 100 целых чисел.
   а) Вывести все числа, которые встречаются в этом массиве несколько раз.
   б) Вывести все числа, которые встречаются в массиве только по одному разу.
6. С помощью датчика случайных чисел заполнить двоичную матрицу 5x10. Определить номер строки с наибольшим количеством нулей.
7. Транспонировать целочисленную матрицу 5x5, т.е. отразить относительно главной диагонали.
8. В двоичной матрице 10x10 найти совпадающие строки.

Информационные источники:

1. **Семакин И.Г.**, Шестаков А.П. Основы программирования: Учебник.- М.: Мастерство, 2002.- 432 с.