Chapter 8 - Arrays

At a Glance

Lesson Contents

Chapter Notes

Overview

Chapter 8 provides an introduction to arrays. You first learn about one-dimensional arrays: how to create them, use them, initialize them, and pass them as function arguments. In a case study, you use a one-dimensional array and learn to compute averages and standard deviations. You also learn about two-dimensional arrays. Several programming examples are used to show you how to use these types of arrays. You are also briefly introduced to larger dimensional arrays. Finally, common programming and compiler errors are reviewed.

Objectives

- One-dimensional arrays
- Array initialization
- Arrays as function arguments
- Case study: Computing averages and standard deviations
- Two-dimensional arrays
- Common programming and compiler errors

One-Dimensional Arrays

Topic Tip	One-dimensional arrays are sometimes called vectors. For more information, see: **http://en.wikipedia.org/wiki/Array**.

Topic Tip	In C, the starting index value for all arrays is always 0. This starting index value is fixed by the compiler and cannot be altered. Although other high-level languages, such as Visual Basic, allow the programmer to change this starting value (even permitting negative values), C does not. In C, the first array element is always 0, and negative index values are not permitted. For more information, see the Programming Note on page 377.

Topic Tip	Note that some compilers permit double-precision variables as subscripts; in these cases, the double-precision value is truncated to an integer value.

Topic Tip	To practice this topic's concept, write a short program in which you try to access a non-existent array element. What happens? Understand that answers will vary depending on the compiler and operating system being used.

Quick Quiz 1

1. What is an atomic variable?
2. What is a data structure?
3. A(n)_____array is a list of values of the same data type that is stored using a single group name.
4. Each item in an array is called a(n)_____or component of the array.

Array Initialization

Topic Tip	Be aware that if the number of initializers is less than the declared number of elements listed in square brackets, the initializers are applied starting with array element 0. After providing this information, ask students to think of an easy way to initialize all elements to zero. They should come up with something like `int myArray[100] = {0};`. Note that this does not work for local `auto` arrays.

Quick Quiz 2

1. The NULL character (_____) is automatically appended to all strings by the C compiler.

2. The individual elements of all global and static arrays (local or global) are, by default, set to_____at compilation time.

3. Are `auto` local arrays initialized at compilation time? If so, to which value?

4. True/False: It is generally advisable to omit the size of the array in the function header line.

Case Study: Computing Averages and Standard Deviations

Topic Tip	For more information on boundary tests, see **http://en.wikipedia.org/wiki/Black_box_testing#Boundary_value_analysis**.

Two-Dimensional Arrays

Quick Quiz 3

1. What is a two-dimensional array?

2. When initializing two-dimensional arrays, the_____braces can be omitted.

3. Initialization in a two-dimensional array is done in_____order.

4. How can you view or interpret arrays of three, four, five, six or more dimensions?

Additional Resources

1. C Tutorial : Arrays:
   https://www.w3resource.com/c-programming/c-array.php

2. C++ Tutorial : Arrays and Vectors:
   https://www.educba.com/c-plus-plus-vector-vs-array/

3. Array:
   http://en.wikipedia.org/wiki/Array

4. How C Programming Works: Arrays:
   http://computer.howstuffworks.com/c10.htm

5. Black Box Testing: Boundary Value Analysis:
   http://en.wikipedia.org/wiki/Black_box_testing#Boundary_value_analysis

Key Terms

- One of the simplest data structures, called an **array数组**, is used to store and process a set of values, all of the same data type that forms a logical group.
- An **atomic variable原子变量**, which is also referred to as a **scalar variable标量变量**, is a variable whose value cannot be further subdivided or separated into a built-in data type.
- C does not check the value of the index being used (called a **bounds check边界检测**).
- In **bubble sort冒泡排序**, successive values in the list are compared, beginning with the first two elements.
- A **data structur数据结构e**, which is also known as an **aggregate data type聚合数据类型**, is a data type with two main characteristics. First, its values can be decomposed into individual data elements, each of which is either atomic or another data structure. Secondly, it provides an access scheme for locating individual data elements within the data structure.
- Each item in an array is called an **element元素** or **component构件** of the array.
- **External sorts外排序** are used for much larger data sets that are stored in large external disk or tape files, and cannot be accommodated within the computer's memory as a complete unit.
- Each individual element is referred to as an **indexed variable索引变量** or a **subscripted variable下标变量** because both a variable name and an index or subscript value must be used to reference the element.
- **Internal sorts内排序** are used when the data list is not too large and the complete list can be stored within the computer's memory, usually in an array.
- The two most common methods of performing such searches are the **linear线性** and **binary search algorithms二分搜索算法**.
- In 1958, John McCarthy developed a language at the MIT specifically for manipulating lists; this language was named **LISP**, the acronym for **List Processing表处理**.
- The **NULL空** character ('\0') is automatically appended to all strings by the C compiler.
- A **one-dimensional array一维数组**, which is also known as both a **single-dimensional array单维数组** and a **single-subscript array单下标数组**, is a list of values of the same data type that is stored using a single group name.
- The **Quicksort algorithm快速排序算法**, which is also called a **"partition" sort**, divides a list into two smaller sublists and sorts each sublist by portioning

into smaller sublists, and so on.

- The third subscript in a three-dimensional array is often called the **rank阶**.
- In a **selection sort选择排序**, the smallest value is initially selected from the complete list of data and exchanged with the first element in the list.
- In a linear search, which is also known as a **sequential search顺序搜索**, each item in the list is examined in the order it occurs until the desired item is found or the end of the list is reached.