

The argument `studentGrades[1]` is the starting address of the second row of the array. Remember that a double-subscripted array is basically an array of single-subscripted arrays and that the name of a single-subscripted array is the address of the array in memory. Function `average` calculates the sum of the array elements, divides the total by the number of test results and returns the floating-point result.

Summary

Section 6.1 Introduction

- Arrays are data structures consisting of related data items of the same type.
- Arrays are “static” entities in that they remain the same size throughout program execution.

Section 6.2 Arrays

- An array is a group of memory locations related by the fact that they all have the same name and the same type.
- To refer to a particular location or element in the array, specify the name of the array and the position number of the particular element in the array.
- The first element in every array is the zeroth element. Thus, the first element of array `c` is referred to as `c[0]`, the second element of array `c` is referred to as `c[1]`, the seventh element of array `c` is referred to as `c[6]`, and, in general, the *i*th element of array `c` is referred to as `c[i - 1]`.
- Array names, like other variable names, can contain only letters, digits and underscores. Array names cannot begin with a digit.
- The position number contained within square brackets is more formally called a subscript. A subscript must be an integer or an integer expression.
- The brackets used to enclose the subscript of an array are actually considered to be an operator in C. They have the same level of precedence as the function call operator.

Section 6.3 Defining Arrays

- Arrays occupy space in memory. You specify the type of each element and the number of elements in the array so that the computer may reserve the appropriate amount of memory.
- An array of type `char` can be used to store a character string.

Section 6.4 Array Examples

- The elements of an array can be initialized when the array is defined by following the definition with an equals sign and braces, `{}`, containing a comma-separated list of initializers. If there are fewer initializers than elements in the array, the remaining elements are initialized to zero.
- The statement `int n[10] = {0};` explicitly initializes the first element to zero and initializes the remaining nine elements to zero because there are fewer initializers than there are elements in the array. It's important to remember that automatic arrays are not automatically initialized to zero. You must at least initialize the first element to zero for the remaining elements to be automatically zeroed. This method of initializing the array elements to 0 is performed at compile time for static arrays and at runtime for automatic arrays.
- If the array size is omitted from a definition with an initializer list, the number of elements in the array will be the number of elements in the initializer list.
- The `#define` preprocessor directive can be used to define a symbolic constant—an identifier that is replaced with replacement text by the C preprocessor before the program is compiled. When the program is preprocessed, all occurrences of the symbolic constant are replaced with the replacement text. Using symbolic constants to specify array sizes makes programs more scalable.

- C has no array bounds checking to prevent a program from referring to an element that does not exist. Thus, an executing program can “walk off” the end of an array without warning. You should ensure that all array references remain within the bounds of the array.
- A string such as “hello” is really a static array of individual characters in C.
- A character array can be initialized using a string literal. In this case, the size of the array is determined by the compiler based on the length of the string.
- Every string contains a special string-termination character called the null character. The character constant representing the null character is ‘\0’.
- A character array representing a string should always be defined large enough to hold the number of characters in the string and the terminating null character.
- Character arrays also can be initialized with individual character constants in an initializer list.
- Because a string is really an array of characters, we can access individual characters in a string directly using array subscript notation.
- You can input a string directly into a character array from the keyboard using `scanf` and the conversion specifier `%s`. The name of the character array is passed to `scanf` without the preceding `&` and used with nonstring variables. The `&` is normally used to provide `scanf` with a variable’s location in memory so that a value can be stored there. An array name is the address of the start of the array; therefore, the `&` is not necessary.
- Function `scanf` reads characters from the keyboard until the first white-space character is encountered—it does not check the array size. Thus, `scanf` can write beyond the end of the array.
- A character array representing a string can be output with `printf` and the `%s` conversion specifier. The characters of the string are printed until a terminating null character is encountered.
- A static local variable exists for the duration of the program but is only visible in the function body. We can apply `static` to a local array definition so that the array is not created and initialized each time the function is called and the array is not destroyed each time the function is exited in the program. This reduces program execution time, particularly for programs with frequently called functions that contain large arrays.
- Arrays that are static are automatically initialized once at compile time. If you do not explicitly initialize a static array, that array’s elements are initialized to zero by the compiler.

Section 6.5 Passing Arrays to Functions

- To pass an array argument to a function, specify the name of the array without any brackets.
- Unlike `char` arrays that contain strings, other array types do not have a special terminator. For this reason, the size of an array is passed to a function, so that the function can process the proper number of elements.
- C automatically passes arrays to functions by reference—the called functions can modify the element values in the callers’ original arrays. The name of the array evaluates to the address of the first element of the array. Because the starting address of the array is passed, the called function knows precisely where the array is stored. Therefore, when the called function modifies array elements in its function body, it’s modifying the actual elements of the array in their original memory locations.
- Although entire arrays are passed by reference, individual array elements are passed by value exactly as simple variables are.
- Such simple single pieces of data (such as individual `ints`, `floats` and `chars`) are called scalars.
- To pass an element of an array to a function, use the subscripted name of the array element as an argument in the function call.

- truyền array vào function như thế nào? `int function(int array[])`
chỉ cần đặt brackets[] để biết là mảng mấy chiều. kích thước không cần khai báo.
~~If it's included, the compiler checks that it's greater than zero, then ignores it.~~
- When an array parameter is preceded by the `const` qualifier, the `elements of the array become constant in the function body`, and any attempt to modify an element of the array in the function body results in a compile-time error.

Section 6.6 Sorting Arrays

- ~~Sorting data (i.e., placing the data into a particular order such as ascending or descending) is one of the most important computing applications.~~
- ~~One sorting technique is called the bubble sort or the sinking sort, because the smaller values gradually "bubble" their way upward to the top of the array like air bubbles rising in water, while the larger values sink to the bottom of the array. The technique is to make several passes through the array. On each pass, successive pairs of elements are compared. If a pair is in increasing order (or if the values are identical), we leave the values as they are. If a pair is in decreasing order, their values are swapped in the array.~~
- ~~Because of the way the successive comparisons are made, a large value may move down the array many positions on a single pass, but a small value may move up only one position.~~
- ~~The chief virtue of the bubble sort is that it's easy to program. However, the bubble sort runs slowly. This becomes apparent when sorting large arrays.~~

Section 6.7 Case Study: Computing Mean, Median and Mode Using Arrays

- ~~The mean is the arithmetic average of a set of values.~~
- ~~The median is the "middle value" in a sorted set of values.~~
- ~~The mode is the value that occurs most frequently in a set of values.~~

Section 6.8 Searching Arrays

- ~~The process of finding a particular element of an array is called searching.~~
- ~~The linear search compares each element of the array with the search key. Since the array is not in any particular order, it's just as likely that the value will be found in the first element as in the last. On average, therefore, the search key will be compared with half the elements of the array.~~
- ~~The linear searching method works well for small or unsorted arrays. For sorted arrays, the high-speed binary search technique can be used.~~
- ~~The binary search algorithm eliminates from consideration one half of the elements in a sorted array after each comparison. The algorithm locates the middle element of the array and compares it to the search key. If they are equal, the search key is found and the array subscript of that element is returned. If they are not equal, the problem is reduced to searching one half of the array. If the search key is less than the middle element of the array, the first half of the array is searched, otherwise the second half of the array is searched. If the search key is not found in the specified subarray (piece of the original array), the algorithm is repeated on one quarter of the original array. The search continues until the search key is equal to the middle element of a subarray, or until the subarray consists of one element that is not equal to the search key (i.e., the search key is not found).~~
- ~~When using a binary search, the maximum number of comparisons required for any array can be determined by finding the first power of 2 greater than the number of array elements.~~

Section 6.9 Multiple-Subscripted Arrays

- A common use of multiple-subscripted arrays (also called multidimensional arrays) is to represent tables of values consisting of information arranged in rows and columns. To identify a par-

ticular table element, we must specify two subscripts: The first (by convention) identifies the element's row and the second (by convention) identifies the element's column.

- Tables or arrays that require two subscripts to identify a particular element are called double-subscripted arrays.
- Multiple-subscripted arrays can have more than two subscripts.
- A multiple-subscripted array can be initialized when it's defined, much like a single-subscripted array. The values are grouped by row in braces. If there are not enough initializers for a given row, the remaining elements of that row are initialized to 0.
- The first subscript of a multiple-subscripted array parameter declaration is not required, but all subsequent subscripts are required. The compiler uses these subscripts to determine the locations in memory of elements in multiple-subscripted arrays. All array elements are stored consecutively in memory regardless of the number of subscripts. In a double-subscripted array, the first row is stored in memory followed by the second row.
- Providing the subscript values in a parameter declaration enables the compiler to tell the function how to locate an element in the array. In a double-subscripted array, each row is basically a single-subscripted array. To locate an element in a particular row, the compiler must know how many elements are in each row so that it can skip the proper number of memory locations when accessing the array.

Terminology

array 196	null character 207
binary search 223	position number 196
bubble sort 216	replacement text 201
const keyword 215	scalable 202
double-subscripted array 229	scalar 212
element 196	search key 223
index (or subscript) 196	searching 223
initializer 199	sinking sort 216
key value 223	subscript 196
linear search 223	survey data analysis 218
<i>m</i> -by- <i>n</i> array 229	symbolic constant 201
multidimensional array 229	table 229
multiple-subscripted array 229	value 196
name 196	zeroth element 196

Self-Review Exercises

- 6.1** Answer each of the following:
- Lists and tables of values are stored in _____.
 - An array's elements are related by the fact that they have the same _____ and _____.
 - The number used to refer to a particular element of an array is called its _____.
 - A(n) _____ should be used to specify the size of an array because it makes the program more scalable.
 - The process of placing the elements of an array in order is called _____ the array.
 - Determining whether an array contains a certain key value is called _____ the array.
 - An array that uses two subscripts is referred to as a(n) _____ array.
- 6.2** State whether the following are *true* or *false*. If the answer is *false*, explain why.
- An array can store many different types of values.
 - An array subscript can be of data type `double`.