

## QUICK REVIEW

---

1. A data type is simple if variables of that type can hold only one value at a time.
2. In a structured data type, each data item is a collection of other data items.
3. An array is a structured data type with a fixed number of components. Every component is of the same type, and components are accessed using their relative positions in the array.
4. Elements of a one-dimensional array are arranged in the form of a list.
5. There is no check on whether an array index is out of bounds.
6. In C++, an array index starts with 0.
7. An array index can be any expression that evaluates to a nonnegative integer. The value of the index must always be less than the size of the array.
8. There are no aggregate operations on arrays, except for the input/output of character arrays (C-strings).
9. Arrays can be initialized during their declaration. If there are fewer initial values than the array size, the remaining elements are initialized to 0.
10. The base address of an array is the address of the first array component. For example, if `list` is a one-dimensional array, the base address of `list` is the address of `list[0]`.
11. When declaring a one-dimensional array as a formal parameter, you usually omit the array size. If you specify the size of a one-dimensional array in the formal parameter declaration, the compiler will ignore the size.
12. In a function call statement, when passing an array as an actual parameter, you use only its name.
13. As parameters to functions, arrays are passed by reference only.
14. Because as parameters, arrays are passed by reference only, when declaring an array as a formal parameter, you do not use the symbol `&` after the data type.
15. A function cannot return a value of type array.
16. Although as parameters, arrays are passed by reference, when declaring an array as a formal parameter, using the reserved word `const` before the data type prevents the function from modifying the array.
17. Individual array components can be passed as parameters to functions.
18. The sequential search algorithm searches a list for a given item, starting with the first element in the list. It continues to compare the search item with the other elements in the list until either the item is found or the list has no more elements left to be compared with the search item.

19. Selection sort sorts the list by finding the smallest (or equivalently largest) element in the list and moving it to the beginning (or end) of the list.
20. For a list of length  $n$ , selection sort makes exactly  $\frac{n(n-1)}{2}$  key comparisons and  $3(n-1)$  item assignments.
21. In C++, a string is any sequence of characters enclosed between double quotation marks.
22. In C++, C-strings are null terminated.
23. In C++, the null character is represented as `'\0'`.
24. In the ASCII character set, the collating sequence of the null character is 0.
25. C-strings are stored in character arrays.
26. Character arrays can be initialized during declaration using string notation.
27. Input and output of C-strings is the only place where C++ allows aggregate operations.
28. The header file `cstring` contains the specifications of the functions that can be used for C-string manipulation.
29. Some commonly used C-string manipulation functions include `strcpy`, `strncpy`, `strcmp`, `strncmp`, and `strlen`.
30. C-strings are compared character by character.
31. Because C-strings are stored in arrays, individual characters in the C-string can be accessed using the array component access notation.
32. Parallel arrays are used to hold related information.
33. In a two-dimensional array, the elements are arranged in a table form.
34. To access an element of a two-dimensional array, you need a pair of indices: one for the row position and one for the column position.
35. In a two-dimensional array, the rows are numbered 0 to `ROW_SIZE - 1` and the columns are numbered 0 to `COLUMN_SIZE - 1`.
36. If `matrix` is a two-dimensional array, then the base address of `matrix` is the address of the array component `matrix[0][0]`.
37. In row processing, a two-dimensional array is processed one row at a time.
38. In column processing, a two-dimensional array is processed one column at a time.
39. When declaring a two-dimensional array as a formal parameter, you can omit the size of the first dimension but not the second.
40. When a two-dimensional array is passed as an actual parameter, the number of columns of the actual and formal arrays must match.
41. C++ stores, in computer memory, two-dimensional arrays in a row order form.