1. All variables in C must be initialized when declared. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
2. In C, variable names are case-sensitive. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
3. The float data type in C can store integer values. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
4. The char data type in C is used to store floating point numbers. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
5. Variables in C can be redeclared with a different type. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
6. A variable declared inside a function is global by default. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
7. Static variables retain their values between function calls. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
8. A constant variable in C can be changed later. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
9. Global variables are accessible throughout the program. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
10. The void data type is used to declare a variable. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
11. A pointer stores the memory address of a variable. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
12. The \* operator is used to access the value at the address stored by a pointer. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
13. The & operator is used to get the address of a variable. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
14. Pointers can point to functions in C. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
15. NULL is a valid value for a pointer. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
16. Pointers must always be initialized before use. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
17. Pointer arithmetic is allowed in C. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
18. You can perform arithmetic on void pointers. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
19. Array names act like pointers in C. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
20. Dangling pointers are safe and recommended. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
21. Strings in C are arrays of characters ending with a null character. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
22. The size of an array must be a constant value in C. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
23. You can change the size of an array after declaration. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
24. Array indices in C start at 1. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
25. A string literal can be modified directly. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
26. Multi-dimensional arrays are supported in C. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
27. An array name can be reassigned like a pointer. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
28. sizeof(array) gives the number of elements in the array. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
29. You can pass arrays to functions in C. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
30. C strings are terminated using '\0'. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
31. Functions in C must always return a value. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
32. C supports recursion. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
33. The main function in C must always return int. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
34. Variables declared inside a function are local to it. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
35. You can declare a function after main. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
36. Function overloading is allowed in C. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
37. You can pass arguments by reference using pointers. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
38. Default arguments are supported in C functions. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
39. C allows functions to return arrays. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
40. Function prototypes are optional in C. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
41. malloc is used for dynamic memory allocation. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
42. free is used to deallocate memory allocated by malloc. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
43. Memory allocated using malloc is automatically initialized to zero. \_\_\_\_\_\_\_\_\_ (Answer:

FALSE)

1. Stack memory is automatically managed. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
2. Heap memory needs to be freed manually. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
3. The sizeof operator gives the memory size in bytes. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
4. calloc initializes memory to zero. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
5. Variables declared with 'register' are stored in RAM. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
6. Static variables are stored in the heap. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
7. You cannot dynamically allocate memory for structs. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
8. The switch statement can handle floating point expressions. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
9. The break statement exits the innermost loop. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
10. continue skips the current iteration of the loop. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
11. A while loop executes at least once. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
12. A do-while loop executes at least once. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
13. for, while, and do-while are all loop constructs in C. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
14. The goto statement is recommended for clean code. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
15. if statements must be followed by else. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
16. The ternary operator is a compact form of if-else. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
17. Nested loops are not allowed in C. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
18. The time complexity of linear search is O(n). \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
19. The time complexity of binary search is O(n). \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
20. The time complexity of merge sort is O(n log n). \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
21. The time complexity of bubble sort is O(n^2). \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
22. The space complexity of quicksort is O(log n) in best case. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
23. Insertion sort is stable. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
24. Merge sort is in-place. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
25. Quicksort is always faster than bubble sort. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
26. Accessing an array element takes O(1) time. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
27. Binary search requires a sorted array. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
28. A struct can contain different data types. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
29. A union occupies memory equal to the largest member. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
30. An enum is used to create symbolic constants. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
31. C supports object-oriented programming. \_\_\_\_\_\_\_\_\_ (Answer: FALSE)
32. C is a compiled language. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
33. Bitwise operators operate on individual bits. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
34. Header files contain function declarations and macros. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
35. The preprocessor runs before the compiler. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
36. #define creates a macro. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)
37. #include is used to include standard libraries. \_\_\_\_\_\_\_\_\_ (Answer: TRUE)