

Top 100+ C Interview Questions and answers

Topics : [C Interview Questions](#)

Written on [November 23, 2023](#)

Basics of C:

1. What is C?

- **Answer:** C is a general-purpose, procedural programming language developed by Dennis Ritchie.

2. Explain the structure of a C program.

- **Answer:** A C program consists of functions, and each function contains declarations, statements, and expressions.

3. What is the difference between `#include <file>` and `#include "file"`?

- **Answer:** `<file>` is used for system header files, and `"file"` is used for user-defined header files.

4. What is the purpose of the `main()` function in C?

- **Answer:** `main()` is the entry point of a C program, and execution starts from here.

Variables and Data Types:

5. What is a variable in C?

- **Answer:** A variable is a named storage location in memory.

6. What are the basic data types in C?

- **Answer:** `int`, `float`, `double`, `char`, `void`.

7. What is the size of the `char` data type?

- **Answer:** The size of `char` is 1 byte.

8. Explain the concept of a pointer.

- **Answer:** A pointer is a variable that holds the memory address of another variable.

Control Flow:

9. What is the purpose of the `if` statement?

- **Answer:** The `if` statement is used for conditional execution of code.

10. Explain the difference between `while` and `do-while` loops.

- **Answer:** `while` tests the condition before the loop, while `do-while` tests it after the loop.

11. What is a `for` loop, and how is it used?

- **Answer:** A `for` loop is used for iterative execution, and it consists of initialization, condition, and iteration expressions.

Functions:

12. What is a function in C?

- **Answer:** A function is a block of code that performs a specific task.

13. Explain the difference between formal parameters and actual parameters.

- **Answer:** Formal parameters are used in the function definition, and actual parameters are used in the function call.

14. What is recursion?

- **Answer:** Recursion is a technique where a function calls itself directly or indirectly to solve a problem.

Arrays and Pointers:

15. What is an array in C?

- **Answer:** An array is a collection of elements of the same data type stored in contiguous memory locations.

16. Explain the concept of a pointer to an array.

- **Answer:** A pointer to an array is a pointer that points to the first element of the array.

17. How do you find the length of an array in C?

- **Answer:** Divide the size of the array by the size of one element (`sizeof(array) / sizeof(array[0])`).

Strings:

18. What is a string in C?

- **Answer:** A string is an array of characters terminated by a null character `\0`.

19. How do you concatenate two strings in C?

- **Answer:** Use the `strcat()` function or concatenate character by character.

20. Explain the purpose of the `strlen()` function.

- **Answer:** `strlen()` returns the length of a string (number of characters).

Structures and Unions:

21. What is a structure in C?

- **Answer:** A structure is a user-defined data type that groups related data members under one name.

22. Explain the difference between a structure and a union.

- **Answer:** In a structure, all members have their own memory space. In a union, all members share the same memory space.

23. How do you access members of a structure?

- **Answer:** Use the dot `(.)` operator: `structInstance.member`.

File Handling:

24. What is a file in C?

- **Answer:** A file is a collection of data stored on a secondary storage device.

25. How do you open a file in C?

- **Answer:** Use the `fopen()` function to open a file.

26. Explain the purpose of the `fclose()` function.

- **Answer:** `fclose()` is used to close a file that was opened using `fopen()`.

Memory Management:

27. What is dynamic memory allocation in C?

- **Answer:** Dynamic memory allocation allows a program to allocate memory at runtime using `malloc()`, `calloc()`, or `realloc()`.

28. What is the purpose of the `free()` function?

- **Answer:** `free()` is used to deallocate memory previously allocated by `malloc()` or related functions.

Preprocessor Directives:

29. What is a preprocessor directive?

- **Answer:** A preprocessor directive is a command that starts with a # symbol, and it is executed before the actual compilation.

30. What is the purpose of #define in C?

- **Answer:** #define is used to create symbolic constants and macros.

Bitwise Operations:

31. What are bitwise operations?

- **Answer:** Bitwise operations manipulate individual bits of binary numbers.

32. Explain the & (bitwise AND) operator.

- **Answer:** & performs a bitwise AND operation.

33. How do you set a particular bit in C?

- **Answer:** Use the | (OR) operator with a mask to set a bit: $x = x | (1 \ll n)$.

Dynamic Memory Allocation:

34. What is the purpose of malloc() and free()?

- **Answer:** malloc() allocates memory dynamically, and free() deallocates previously allocated memory.

35. What is a memory leak?

- **Answer:** A memory leak occurs when a program allocates memory but fails to deallocate it, leading to a gradual loss of available memory.

Enumerations:

36. What is an enumeration in C?

- **Answer:** An enumeration is a user-defined data type that consists of named integer constants.

37. How do you declare an enumeration?

- **Answer:** enum Weekday {Monday, Tuesday, ...};

Miscellaneous:

38. **What is the purpose of the sizeof operator in C?**

- **Answer:** sizeof returns the size, in bytes, of a variable or data type.

39. **Explain the purpose of the const keyword.**

- **Answer:** const is used to declare constants, and it indicates that the variable's value cannot be changed.

40. **What is the difference between ++i and i++?**

- **Answer:** Both increment the value of i by 1, but ++i is the pre-increment operator, and i++ is the post-increment operator.

41. **Explain the typedef keyword.**

- **Answer:** typedef is used to create an alias for existing data types.

42. **What is the purpose of the return statement in a function?**

- **Answer:** The return statement is used to specify the value a function should return.

43. **What is the purpose of the break statement?**

- **Answer:** The break statement is used to exit from a loop or switch statement.

44. **Explain the difference between NULL and 0.**

- **Answer:** NULL is a macro representing a null pointer, while 0 is the integer literal zero.

45. **What is a comment, and how do you write comments in C?**

- **Answer:** Comments are explanatory notes in the code. In C, you can use // for single-line comments and /* */ for multi-line comments.

46. **What is the purpose of the continue statement?**

- **Answer:** The continue statement is used to skip the rest of the loop and move to the next iteration.

47. **Explain the concept of a constant pointer.**

- **Answer:** A constant pointer is a pointer whose address cannot be changed but the value it points to can be modified.

48. **What is the purpose of the volatile keyword?**

- **Answer:** The volatile keyword indicates that a variable may be changed by some external factor, preventing the compiler from optimizing code involving that variable.

49. **Explain the difference between #include and #include<>.**

- **Answer:** #include is used for user-defined header files, and #include<> is used for system header files.

50. How is a structure passed to a function in C?

- **Answer:** A structure is usually passed to a function by passing the structure variable as an argument.

51. What is the purpose of the static keyword?

- **Answer:** The static keyword has different meanings depending on the context. In the case of a variable, it makes the variable retain its value between function calls. In the case of a function, it limits the function's scope to the file where it is declared.

52. Explain the purpose of the const pointer.

- **Answer:** A const pointer is a pointer that cannot be used to modify the value it points to.

53. What is the purpose of the inline keyword?

- **Answer:** The inline keyword suggests to the compiler that it should attempt to generate inline code for a function.

54. Explain the concept of a macro in C.

- **Answer:** A macro in C is a fragment of code that has been given a name.

55. What is the purpose of the register keyword?

- **Answer:** The register keyword suggests to the compiler that a variable will be heavily used and should be kept in a processor register if possible.

56. Explain the purpose of the do-while loop.

- **Answer:** The do-while loop is used to execute a block of code repeatedly until the specified condition becomes false. It guarantees that the loop body is executed at least once.

57. What is the purpose of the volatile keyword in C?

- **Answer:** The volatile keyword indicates to the compiler that a variable's value may change at any time without any action being taken by the code the compiler finds nearby.

58. Explain the difference between a shallow copy and a deep copy.

- **Answer:** A shallow copy copies the elements of the source object, but not the objects they reference. A deep copy copies the elements and recursively copies the objects they reference.

59. What is the purpose of the restrict keyword in C99?

- **Answer:** The restrict keyword is a declaration of intent given by the programmer to the compiler, indicating that for the scope of the pointer declaration, the target object pointed to by the pointer is the only object accessed during that scope.

60. **Explain the concept of function pointers in C.**

- **Answer:** Function pointers in C are variables that store addresses of functions.

61. **What is a self-referential structure?**

- **Answer:** A self-referential structure is a structure that contains a pointer to the same type of structure.

62. **Explain the purpose of the const pointer in C.**

- **Answer:** A const pointer is a pointer that cannot be used to modify the value it points to.

63. **What is the purpose of the sizeof operator in C?**

- **Answer:** sizeof is a compile-time unary operator that returns the size, in bytes, of its operand.

64. **Explain the purpose of the const qualifier in C.**

- **Answer:** The const qualifier is used to specify that a variable's value should not be modified.

65. **What is the purpose of the volatile qualifier in C?**

- **Answer:** The volatile qualifier is used to indicate that a variable's value may be changed by external factors, and therefore the compiler should not optimize or cache the variable.

66. **What is the purpose of the auto keyword in C?**

- **Answer:** In modern C, the auto keyword is rarely used. In C, it was historically used to declare automatic variables, but it is implicit in most cases.

67. **Explain the concept of a multi-dimensional array in C.**

- **Answer:** A multi-dimensional array is an array of arrays. It can be thought of as an array of rows, where each row is itself an array.

68. **What is the purpose of the const keyword when used with a pointer in C?**

- **Answer:** When used with a pointer in C, the const keyword indicates that the value being pointed to is constant and cannot be modified.

69. **Explain the purpose of the volatile keyword when used with a pointer in C.**

- **Answer:** When used with a pointer in C, the volatile keyword indicates that the value being pointed to may change at any time without any action being taken by the code nearby.

70. **What is the purpose of the restrict keyword in C99?**

- **Answer:** The restrict keyword is a declaration of intent given by the programmer to

the compiler, indicating that for the scope of the pointer declaration, the target object pointed to by the pointer is the only object accessed during that scope.

71. Explain the concept of a union in C.

- **Answer:** A union in C is a user-defined data type that allows different data types to be stored in the same memory location.

72. What is a type qualifier in C?

- **Answer:** A type qualifier in C is a keyword that is applied to a data type to modify its behavior.

Pointer Arithmetic:

73. Explain pointer arithmetic in C.

- **Answer:** Pointer arithmetic involves performing arithmetic operations on pointers, such as adding or subtracting an integer.

Function Pointers:

74. What are function pointers in C?

- **Answer:** Function pointers are pointers that point to functions instead of data.

Bit Manipulation:

75. How do you set, clear, and toggle bits in C?

- **Answer:** Use `|` to set, `& ~` to clear, and `^` to toggle bits.

Dynamic Memory Allocation:

76. What is the difference between `malloc()` and `realloc()`?

- **Answer:** `malloc()` allocates memory, and `realloc()` changes the size of a previously allocated block.

Structures Padding and Packing:

77. Explain structure padding and packing in C.

- **Answer:** Structure padding involves adding empty bytes to align members, while packing reduces or eliminates padding.

Function Overloading:

78. Does C support function overloading?

- **Answer:** No, C does not support function overloading; functions must have unique names.