52 C Programming Interview Questions to Hire Top Developers

Questions

1. Can you explain the difference between a pointer and a regular variable?
2. What is the significance of the 'static' keyword in C programming?
3. How does memory management work in C, and how do you prevent memory leaks?
4. Describe the process of compiling a C program. What are the main stages?
5. What are the different storage classes in C, and how do they impact variable scope?
6. Can you illustrate how to use a structure in C and why it is useful?
7. How do you handle errors in C? Can you provide an example of error handling?
8. What is the difference between 'malloc' and 'calloc' in C?
9. Explain the concept of recursion with a simple example in C.
10. How do you create a multi-file project in C, and what are the benefits?
11. What is the purpose of the 'extern' keyword in C?
12. How does the 'const' keyword affect a variable in C?
13. Can you explain what a 'dangling pointer' is?
14. What is the difference between 'break' and 'continue' statements in C?
15. How do you define a macro in C and what are its uses?
16. What is the role of the 'volatile' keyword in C?
17. How do you perform file operations in C?
18. Can you explain what a segmentation fault is and how you might troubleshoot it?
19. Can you explain the concept of function pointers in C and provide a practical use case?
20. How does bitwise manipulation work in C, and when would you use it?
21. What are preprocessor directives in C, and how do they differ from regular C statements?
22. Explain the difference between shallow copy and deep copy in C, particularly with structs.
23. How does type casting work in C, and what are the potential risks?
24. What is the purpose of the 'union' keyword in C, and how does it differ from a struct?
25. Can you describe how to implement a linked list in C?
26. What is the significance of the 'restrict' keyword in C99?
27. How do you handle command-line arguments in a C program?
28. Explain the concept of function inlining in C and its potential benefits.
29. What are variadic functions in C, and how are they implemented?
30. How do you use function pointers to implement callbacks in C?
31. What is the purpose of the 'volatile' keyword in C, especially in embedded systems?
32. Can you explain how to create and use a circular buffer in C?
33. What are the differences between stack and heap memory allocation in C?
34. Can you explain the concept of memory fragmentation in C?
35. How would you detect a memory leak in a C program?
36. What's the difference between stack and heap memory allocation in C?
37. How does the C memory model handle multithreading?
38. What strategies would you use to minimize memory usage in a C program?
39. Explain the concept of memory alignment in C and why it matters.
40. How would you implement a custom memory allocator in C?
41. How would you implement a stack using an array in C? What are the potential limitations?
42. Can you explain how to create a binary search tree in C and discuss its time complexity for search operations?
43. What is a hash table, and how would you implement one in C to handle collisions?
44. How do you implement a queue using two stacks in C?
45. Can you describe the difference between a singly linked list and a doubly linked list? When would you choose one over the other?
46. How would you implement a priority queue in C? What data structure would you use?
47. Explain how you would use a trie (prefix tree) in C for efficient string searching.
48. Can you describe how to implement a graph using an adjacency list in C?
49. How would you design a circular buffer in C, and what are its applications?
50. Can you explain the concept of a heap data structure and how to implement a min-heap in C?
51. How would you implement a sparse matrix in C to efficiently store and operate on mostly zero elements?