C interview questions:

1. what are macros and write a macro to calculate the area of a square
   https://www.javatpoint.com/c-macros
2. Inline function vs macro. Why do we need inline when we can use a macro? Exclude the scenarios where inline will be executed as a normal function and explain.Why is debugging faster when the inline function is used?

3. Call the main function from the cmd line and pass two arguments to it and with these arguments call macro and inline function defined above. Whose execution will be faster?

4. What are the storage classes in C/C++
   C: Link
   C++: Link

5.What are pointers? He asked me to explain them along with the code.
What are different types of pointers?
JAVATPOINT
GFG

6. What is call by value and call by reference? Write a code to implement it and explain the memory level working of it.

7. Why we use the C language in hardware programming?

8. What is dangling pointer?
JAVATPOINT

9.Phases of the compiler. What is MakeFile? Explain and show how to make one?

10. Memory allocation of a c++ program
memory layout

11. Create a Singleton Class (Done using Private destructor and static object creation).

12. Explain the concept of VTable. He expected me to know where it is used and how is it implemented. I started explaining him the dynamic polymorphism through virtual function and explained how Vtable and the virtual functions are mapped and also I explained him about vptr concepts used.
https://pabloariasal.github.io/2017/06/10/understanding-virtual-tables/
He then asked me to write a singleton class. I explained to him the concept but he insisted me to write a class for the same. I was able to answer it by using static bool variable. Although it was not perfect, he seemed okay with the implementation.

13. new and delete keywords (Dynamic Memory Allocation)

14. Allocate memory dynamically (Heap memory) and then differences between

stack memory and heap memory. Different methods/operators for allocation?

15. C program Execution Flow. Content of Executable C file
    https://www.geeksforgeeks.org/how-does-a-c-program-executes/
16. Difference between memcopy and memmove
    Stackoverflow Link
17. Implement memcopy without using temporary array. What is the problem with memcopy. ?
    gfg link
18. Check Stack Segment is in growing or shrinking direction
    (Create two local variables. Local variables are stored on a stack).
    GFG LINK
    STACKOVERFLOW
19. What will return malloc(0)?How does free(ptr) decide How much memory should be cleared?
    GFG LINK
20. What is a *void pointer* and how do we use it.
21. About void pointer (since I allocated memory using malloc)
    write a code on where and how volatile keyword in C is useful in both read and write Environment?
    - Answer: For Reading environment, I wrote code to take a file as an input in and stored in a volatile keyword to check for hardware interrupts and, For Write Environment, I used mutex variables and threads to code and explain the change of behaviour during process synchronization. (he was impressed with my answers).
22. a function is passed a void pointer like myfunc(void *p). Determine memory address pointed by p is on stack or heap.
23. Maximum Memory size that can be allocated by malloc
24. What are memory leaks & how to recover from memory leakage in a program? [We can't recover within a program. Care should be taken to avoid memory leakage, i.e, by freeing unused variables.]
25. How to free dynamically allocated memory without using free().
26. How is *free* keyword different from *delete.*
27. how does free() know the size of memory to be deallocated?

gfg link

28. what will happen if i call free(NULL)? Any error?
29. What happens if i don't free memory and assign some other address to an existing pointer variable pointing to some memory on heap?

30. Code snippets were given and outputs should be determined. Codes were related to
- pointer referencing, dereferencing & incrementing
- size & memory allocations of structure & unions
- memset function
- constant pointer and pointer to a constant.
- executing volatile variables in functions
- Based on local vs global variable precedence.
- Based on various type of pointer typecasting
- https://www.geeksforgeeks.org/difference-between-const-int-const-int-const-and-int-const/

31. Structure padding & packing, structure lining and how will you avoid structure padding.Who do this padding and packing(compiler, cpu etc.)?
JAVATPOINT LINK

32. Difference between structure and unions and their code implementation.why is union preferred over structure in embedded systems.

33. How are variables of a union stored? If the union contains an integer(4 bytes) and a single character(1 byte). Will the character points to the MSB or LSB? The question focuses on how the variables of the union are pointed in the memory block allotted to the union variable. [Concept of little-endian and big-endian systems]

34. Implement your own sizeof function [Typecasting pointers].Find the size of int in C without using sizeof() operator.

35. What is the volatile keyword? How does a volatile variable change?

36. There are two 32 bit HW timer registers – TIMER_L, TIMER_H, which form a 64-bit timer(TIMER_L — lower 32 bit, TIMER_H — higher 32 bit). To design an API to determine a 64-bit timer, how many register reads are required at best case and worst case?

37. What is a segmentation fault & How to avoid it?

38. If you are to allocate the memory block to a different variable, how will you do it? [allocation, deallocation, reallocation of the main memory. An exact answer was not required. The interviewer wants to know how I would come up with a solution.]

39. If you are writing a program to allocate memory to variables, how will you avoid memory corruptions due to the allocation of pre-allocated memory?

40. Suppose in a normal program if a variable or a memory location is corrupted

how to identify what portion is corrupted?

41. Double pointers.

42. Write code snippet for dynamic memory allocation to 1-dimensional, 2-dimensional arrays.

43. What are function pointers & why do we need them as we can call a function from any other function

44. Enumeration and max range of enum in C,

45. Pointer declaration related question like how given pointer declare is read out like int * a[9]; or int(*a[9]);

https://www.geeksforgeeks.org/pointer-array-array-pointer/

46. Compiler related question like full phases of a written program. Like conversation of c.obj file-> C.exe file etc. The symbol table, loader, linker, etc. (refer Ravula compiler lecture video)
https://www.geeksforgeeks.org/introduction-of-compiler-design/

47. What will be the output of the below program.

```
int main() {
            char a=120,b=140;
            int i;
            i=a+b;
            printf("%d",i);
            }
```

Note: ans=4, also need to explain how the answer will be 4.
What will be the size of the following structure.

```
struct node{
        int a;
        float b;
        double c;
       }
```

48. Virtual function and virtual class
49. Function overriding and function overloading difference
50. How to implement function overriding in C

51. Extern keyword related questions
52. What are the differences between Global, Extern, Static and Volatile keywords?
53. Different scopes of variables and to explain when each of them is used and why, when and how do we use the *extern* keyword.
54. Returning ref of the local variable from the function.
55. Declaring an array using constant passed to a function.
56. They asked me to dynamically allocate memory and make appropriate changes.
57. What is a callback function?

MORE SUGGESTED QUESTIONS:
https://www.interviewbit.com/c-interview-questions/

1. File handling questions in C.
2. How to store sum of two large integers.
3. Use of pragma in c.
4. size of below
   structure struct st

```
{
  short int s;
  char      c;
  int       i;
} st_var;
```

5. How to check overflow in addition program of two numbers
6. write a C++ code to implement stack overflow and heap memory overflow.
7. Which one would be better between Hash Table and Queue.
8. Describe two of the features of the OOPs with real life example.
9. how will you identify buffer overflow if I multiply two integers.

write a code to identify whether it is big-endian or little-endian.

Initially, I proposed using shift operators, but he explained that it won't work. So Luckily I could come up with storing numbers as hex values in strings and check from left and right and coded my solution.

- Answer: arranging the variable declaration example (change declaration of variable 32 bits + 8 (padding 24) bits + 32 bits to 8 bits + 32 bits + 32 bits). He was impressed because this is an actual method instead of using __attribute__((packed)) this method is actually good.

10. Swap two nibbles.
11. Set Kth bit of a number.
12. Identify weather a machine is Little ENDian or Big ENDian.
13. What is polymorphism?
14. What is inheritance?
15. Diamond Problem.
16. Generic templates.

17. Stack Smashing and how does function calls store in stack .
18. what are the different intermediate files while compiling and running a C program
19. How does a process look like in memory, the difference between *global variables* and *static variables*?
20. What are *Static* and *Dynamic* Libraries?
https://www.geeksforgeeks.org/static-vs-dynamic-libraries/
21. Let us suppose there are multiple .c files and you want a function from a particular file can only be accessed in that c file only. (Ans- Use static keyword)

22. What happened when you enter a URL on browser. Explain all in detail.
    https://medium.com/@maneesha.wijesinghe1/what-happens-when-you-type-an-url-in-th e-browser-and-press-enter-bb0aa2449c1a

23. How to pass an array with the help of pointers. While passing 2d array, why it is necessary to pass number of columns?