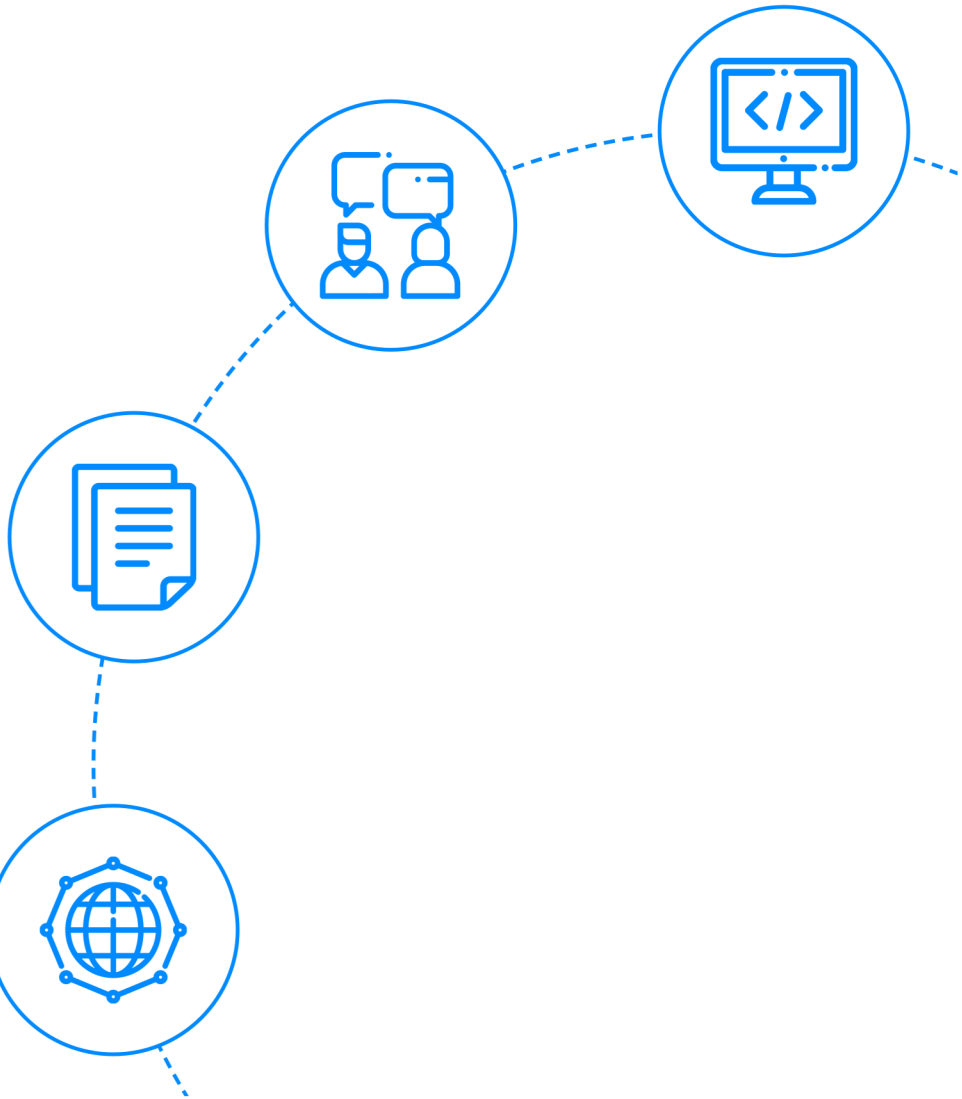
InterviewBit

C Interview Questions



To view the live version of the page, click here.

Contents

Basic Interview Questions On C

C Intermediate Interview Questions

C Intermediate Interview Questions

(.....Continued)

19. What is the difference between #include "..." and #include <...>?
20. What are dangling pointers? How are dangling pointers different from memory leaks?
21. What is the difference between ‘g’ and “g” in C?

C Interview Questions For Experienced

22. Can you tell me how to check whether a linked list is circular?
23. What is the use of a semicolon (;) at the end of every program statement?
24. Differentiate Source Codes from Object Codes
25. What are header files and what are its uses in C programming?
26. When is the "void" keyword used in a function
27. What is dynamic data structure?
28. Add Two Numbers Without Using the Addition Operator
29. Subtract Two Number Without Using Subtraction Operator
30. Multiply an Integer Number by 2 Without Using Multiplication Operator
31. Check whether the number is EVEN or ODD, without using any arithmetic or relational operators
32. Reverse the Linked List. Input: 1->2->3->4->5->NULL Output: 5->4->3->2->1->NULL
33. Check for Balanced Parentheses using Stack
34. Program to find n’th Fibonacci number
35. Write a program to find the node at which the intersection of two singly linked lists begins.
36. Merge Two sorted Linked List

Let's get Started

C, one of the most popular computer languages today because of its structure, high-level abstraction, machine-independent feature, etc.

C language was developed to write the UNIX operating system, hence it is strongly associated with UNIX, which is one of the most popular network operating systems in use today and the heart of internet data superhighway.

In this article, you will understand the questions you could expect at a fresher, intermediate, and advanced level.

Basic Interview Questions On C

1. What's the value of the expression 5["abxdef"]?

The answer is **'f'.**

Explanation: The string mentioned "abxdef" is an array, and the expression is equal to "abxdef"[5]. Why is the inside-out expression equivalent? Because a[b] is equivalent to *(a + b) which is equivalent to *(b + a) which is equivalent to b[a].

2. What is a built-in function in C?

The most commonly used built-in functions in C are sacnf(), printf(), strcpy, strlwr, strcmp, strlen, strcat, and many more.

Built-function is also known as library functions are provided by the system to make the life of a developer easy by assisting them to do the certain commonly used predefined task. For example: if you need to print output or your program into the terminal we use printf() in C.

3. In C, What is the #line used for?

In C #line is used as a preprocessor to re-set the line number in the code, which takes a parameter as line number, herewith is an example for the same.

```c
#include <stdio.h>                              /*line 1*/
                                                /*line 2*/
    int main(){                                 /*line 3*/
                                                /*line 4*/
    printf("Nello world\n");                    /*line 5*/
    //print current line                        /*line 6*/
    printf("Line: %d\n",_LINE_);                /*line 7*/
    //reset the line number by 36               /*line 8*/
    #line 36    /*reseting*/
    //print current line                        /*line 36*/
    printf("Line: %d\n",_LINE_);                /*line 37*/
    printf("Bye bye!!!\n");                     /*line 39*/
                                                /*line 40*/
    return 0;                                   /*line 41*/
}                                               /*line 42*/
```

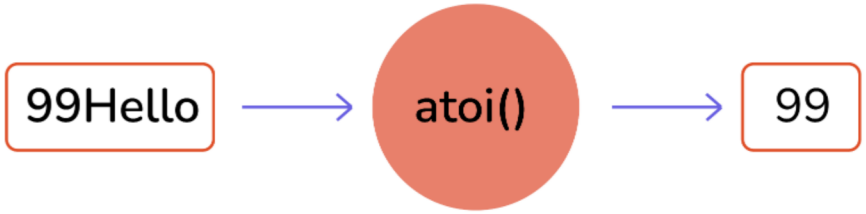
4. How can a string be converted to a number?

The function takes the string as an input that needs to be converted to an integer.

```c
int atoi(const char *string)
```

Return Value:

- On successful conversion, it returns the desired integer value
- If the string starts with alpha-numeric char or only contains alpha-num char, 0 is returned.
- In case string starts with numeric character but followed by alpha-num char, the string is converted to integer till the first occurrence of alphanumeric char.



Converting String to Number

5. How can a number be converted to a string?

The function takes a pointer to an array of char elements that need to be converted, and a format string needs to be written in a buffer as a string

```
int sprintf(char *str, const char *format, ...)
```

```
#include<stdio.h>
#include <math.h>
int main()
{
    char str[80];

    sprintf(str, "The value of PI = %f", M_PI);
    puts(str);

    return 0;
}
```

Below is the output after running the above code:

Output: Value of Pi = 3.141593

6. Why doesn't C support function overloading?

After you compile the C source, the symbol names need to be intact in the object code. If we introduce **function overloading** in our source, we should also provide name mangling as a preventive measure to avoid function name clashes. Also, as C is not a strictly typed language many things(ex: data types) are convertible to each other in C, Therefore the complexity of overload resolution can introduce confusion in a language such as C.

When you compile a C source, symbol names will remain intact. If you introduce function overloading, you should provide a name mangling technique to prevent name clashes. Consequently, like C++, you'll have machine-generated symbol names in the compiled binary.

Additionally, C does not feature strict typing. Many things are implicitly convertible to each other in C. The complexity of overload resolution rules could introduce confusion in such kind of language

7. What is the difference between global int and static int declaration?

The difference between this is in scope. A truly global variable has a global scope and is visible everywhere in your program.

```
#include <stdio.h>

int my_global_var = 0;

int
main(void)

{
  printf("%d\n", my_global_var);
  return 0;
}
```

global_temp is a global variable that is visible to everything in your program, although to make it visible in other modules, you'd need an ”extern int global_temp; ” in other source files if you have a multi-file project.

A static variable has a local scope but its variables are not allocated in the stack segment of the memory. It can have less than global scope, although - like global variables - it resides in the .bss segment of your compiled binary.

```c
#include <stdio.h>

int
myfunc(int val)

{
    static int my_static_var = 0;

    my_static_var += val;
    return my_static_var;
}

int
main(void)

{
   int myval;

   myval = myfunc(1);
   printf("first call %d\n", myval);

   myval = myfunc(10);

   printf("second call %d\n", myval);

   return 0;
}
```

8. Difference between const char* p and char const* p?

const char* p is a pointer to a const char.

char const* p is a pointer to a char const.

Since const char and char const are the same, it's the same.

9. Why n++ execute faster than n+1 ?

n++ being a unary operation that just needs one variable whereas n = n + 1 is a binary operation that adds overhead to take more time(Also binary operation: n += 1). However, in modern platforms, it depends on few things such as Processor Architecture, C Compiler, Usage in your code, Other factors such as hardware problems.

However in the modern compiler even if you write n = n + 1 it will get converted into n++ when it goes into the optimized binary, and it will be equivalently efficient.

```
$ time perl -le '$n=0; foreach (1..100000000) { $n++ }'
real    0m2.389s
user    0m2.371s
sys 0m0.015s

$ time perl -le '$n=0; foreach (1..100000000) { $n=$n+1 }'
real    0m4.469s
user    0m4.442s
sys 0m0.020s
```

10. What are the advantages of Macro over function?

Macro on a high-level copy-paste its definitions to places wherever it is called due to which it saves a lot of time, as no time is spent while passing the control to a new function as the control is always with the callee function. However, one downside is the size of the compiled binary is large but once compiled the program comparatively runs faster.

C Intermediate Interview Questions

11. Specify different types of decision control statements?

All statements written in a program are executed from top to bottom one by one. Control statements are used to execute/transfer the control from one part of the program to another depending on the condition.

- If-else statement.
  - normal if-else statement.
  - Else-if statement
  - nested if-else statement.
- Switch statement.

12. What is the difference between struct and union in C?

A struct is a group of complex data structure which is stored in a block of memory where each member on the block gets separate memory location which makes them accessible at once

Whereas in union all the member variables are stored at the same location on the memory as a result to which while assigning a value to a member variable will change the value of all other members.

```
/* struct & union defination*/
struct bar {
    int a;  // we can use a & b both simultaneously
    char b;
}   bar;

union foo {
    int a;  // we can't use both a and b simultaneously
    char b;
}   foo;

/* using struc and union variables*/

struct bar y;
y.a = 3;    // OK to use
y.b = 'c'; // OK to use

union foo x;
x.a = 3; // OK
x.b = 'c'; // NOl this affects the value of x.a!
```

13. What is call by reference in functions?