50+ most asked C Interview Questions and Answers

Hi everyone, welcome back. In this module, I will be discussing the 50+ most asked C Interview Questions and Answers. If you want to learn more about the C programming language, check out our C programming course. Let us now discuss it.

50+ most asked C Interview Questions and Answers

1. Distinguish between calloc() and malloc().

calloc()	malloc()
The calloc() function is used for allocating multiple blocks of memory dynamically.	The malloc() function is used for allocating a single block of memory dynamically.

It sets the default value for each block to '0'.	Default garbage values are assigned to each block, initially.
Syntax: ptr =(typecast*) calloc(n,size)	Syntax: ptr = (typecast*) malloc(size)

2. Distinguish between actual and formal parameters.

The actual parameters are the variables or values passed during the function call whereas the formal parameters are the variables or passed during the function definition.

3. Can a C program be built or run without a main() function?

The execution of a program starts from the main() function so without the main() function, a program will be compiled but never run. Hence, it is mandatory to use the main() function in a program.

4. Why is C referred to as a mid-level programming language?

C is referred to as a mid-level programming language because it possesses characteristics of both low-level and high-level computer languages.

5. List some of the features that make up the 'C' programming language.

- Mid-level programming language
- Better memory management using pointers
- Simple
- Portable
- Structured

6. What are the functions printf() and scanf() in a C program for?

Both printf() and scanf() are predefined functions in the stdio.h header file. The printf() is used to print any message to the output screen and scanf() is used to take inputs from the user.

7. Define call by reference versus call by value in C language.

In call-by-value, any change in the formal parameters does not reflect the change in the actual parameters. The copy of the variable is passed as actual arguments.

In call by reference, any change in the formal parameters reflects the change in the actual parameters. The address of a variable is passed as actual arguments.

8. Can I use the int data type to hold a value of 32768?

No, I cannot use int data type to store a value of 32768 because the range of int data type is -32768 to 32767.

9. Define array in C.

An array is a user-defined data type used to store a group of elements of the same type referenced under a single name.

10. Define pointers in C.

A pointer is a variable that is used to store or hold the address of another variable of the same type.

11. What is a static variable in C?

A static variable in C is a variable whose lifetime is throughout the program i.e., it is destroyed after the complete execution of the program. Its default is 0. They are initialized only once.

12. What exactly is typecasting in C?

When changing one data type to another, this is called typecasting.

13. What do you understand about static memory allocation?

- In static memory allocation, allocation of memory is done at compile time.
- The memory allocated cannot be resized & de-allocated at run time.
- It is utilized in the array.
- It is quicker than dynamic memory allocation.
- The lifetime of a static variable is till the end or termination of program execution.

14. Name the functions that are used for dynamic memory allocation in the C language.

- malloc()
- calloc()
- free()
- realloc()

15. In C, write a program that prints “Hello, World!” with no semicolon (;) is needed.

```
#include<stdio.h>
int main() {
   if(printf("Hello, World!")){}
   return 0;
}
//Output
//Hello, World!
```

16. In C, write a program that swaps two variables without using a third variable to do so.

```
#include<stdio.h>
int main()
{
  int a,b;
  scanf("%d %d",&a,&b);
  printf("Before swapping: a=%d & b=%d.\n",a,b);
  a=a+b;
  b=a-b;
  a=a-b;
  printf("After swapping: a=%d & b=%d.\n",a,b);
  return 0;
}
//Output
//Before swapping: a=21 & b=34.
//Before swapping: a=34 & b=21.
```

17. Define structure in C.

A structure is a user-defined data type that is used for storing the elements of multiple data types in a single unit. In C, a structure is defined using the struct keyword.

18. Define union in C.

A union in C is a user-defined data type that allows various types of data to be stored in a single unit. The memory occupied by the union is the size of the largest member only which is shared by all other members of the union. In C, we define union using the union keyword.

19. Name the data types used in C programming languages.

- int
- float
- char
- double
- void

20. Mention the difference between pre-increment & post-increment operators.

In the pre-increment operator, a variable is first incremented by 1 and then gets stored or assigned in the same variable.
In the post-increment operator, a variable is first stored and then gets incremented by 1.

21. In the C programming language, what does pointer to pointer mean?

A pointer to pointer, as its name indicates, is a variable that holds or includes the address of another pointer variable in its memory.
For example,

```c
#include <stdio.h>
int main() {
  int x=30;
  int *p=&x;
  int **ptr=&p;
  printf("x=%d\n",x);
  printf("*p=%d\n",*p);
  printf("**ptr=%d\n",**ptr);
  return 0;
}
//Output
//x=30
//*p=30
//**ptr=30
```

22. Define header file in C.

A header file in C is a file having “.h” file extensions. It contains a set of predefined function declarations, keywords, and macros definitions. We use “#include” to include header files in our program so that we can use predefined functions and macros.

23. What happens if a header file is included twice in a C program?

If a header file is included twice in a C program, the second one is ignored.

24. Determine the difference between the header file enclosed in angular brackets and double quotes “”.

When a header file is included with >, the compiler only looks for it in the built-in include path. When a header file is included with ” “, the compiler looks for it first in the current working directory, then in the built-in include path if it isn’t found.

25. What do you mean when you say while(0) and while(1)?

while(0) is equivalent to while(false) so the loop will never be executed whereas while(1) is equivalent to while(true) so it will result in an infinite loop.

26. What do you understand by command line arguments in C?

The parameters given to the main() method of a C program from the command line during execution are known as command-line arguments.

Syntax

```
int main(int argc, char *argv[]) {}
```

argc denotes number of arguments and argv denotes a pointer array holding arguments passed.

27. In C, explain the differences between struct and union.

Struct	Union
Structures are defined with the struct keyword in C.	A union in C is defined using the union keyword.
The memory space occupied by the structure is equal to the sum of all the members' size i.e., each member of the structure gets a separate memory.	The memory space occupied by the union is equal to the size of the largest member i.e., a single memory block is shared by all the members of the union.

28. Define typedef keyword in C.

In the C programming language, typedef is a keyword that is used to define an alias or alternate name for an existing type.

Syntax

```
typedef data-type variable_name;
```

For example,

```
#include<stdio.h>
int main() {
  typedef int variable;
  variable x=3;
  printf("%d",x);
  return 0;
}
```

29. In C, write a program that adds two integers without using the addition operator (+).

```
#include<stdio.h>
int main() {
  int a=5,b=6;
  // Method-1
  printf("%d\n",a-(-b));
  // Method-2
  printf("%d",abs(-a-b));
    return 0;
}
```

30. Explain why a++ executes faster than a+1.

If you want to increase the value of a++, you only need one machine instruction, such as INR. If you want to increase it by a+1, you need more than one instruction.

31. How can you tell the difference between '=' and '=='?

'=' is an assignment operator which is used to assign values to variables whereas '==' is a comparison operator which is used to compare two values.

32. What do you understand by enum in C?

An enum is a user-defined data type in C created using the enum keyword. It makes the program more readable and understandable by assigning names to integer constants.
For example,

```
#include<stdio.h>
enum days{
  True=1,False=0
};
int main() {
  printf("%d %d",True,False);
  return 0;
}
```

33. Is there a difference between the getch() and getche() functions in the C programming language?

Yes, both getch() and getche() are different functions in C. getch() takes an input character from the keyboard but doesn't print it on the screen whereas also takes an input character from the keyboard but prints it on the screen.

34. Is it possible to make your header files?

Yes, we can make our header files in C, and by using #include directive we can make them included in our program.

35. Define macros.

Macros are code segments that are replaced by the macro value. It is defined by the #define directive.

36. In C, is it possible to take a variable number of arguments in a function?

Yes, a function variable number of arguments by using ellipsis operator (…).

37. What is dynamic memory allocation in C?

Dynamic memory allocation in C is defined as the allocation of memory at run time or during the execution of the program. Thus, memory size can be increased or decreased according to our requirements. We use calloc() and malloc() to allocate memory dynamically and free() to deallocate the allocated memory.

38. Define wild pointers in C.

A wild pointer is a pointer that is not initialized with any value.
For example, int *ptr;

39. What do you know about far pointers and near pointers?

A far pointer is a pointer that is used to store 32-bit addresses and can access 2^32 or 4096 MB of a data segment.

In a 16-bit computer, a near-pointer is used to store 16-bit addresses within the current segment. They can access 2^16 i.e., 64KB of a data segment.

40. How many types of phases are there in recursion?

There are 2 phases of recursion in C:

1. Winding phase
2. Unwinding phase

41. Is it possible to call main() function recursively?

Yes, we may recursively call the main() function.

42. Is there a limit to how long an identifier may be in C?

Yes, the maximum length of identifier can be upto 31 characters.

43. What is the purpose of the pointer in the C programming language?

- Dynamic memory allocation
- Used in data structures such as linked lists, graphs, and trees.
- Call by reference.
- Array items can be accessed by using pointers.

44. What do you understand by the scope of a variable in a C program?

The scope of a variable is defined as a section or region of the program where a variable can be easily accessed.

45. Mention some of the differences between C++ and C.

C++	C
It is both a procedural and object-oriented programming language.	It is a structured or procedural programming language.
For dynamic memory allocation, we use new and delete operators.	We use malloc(), calloc() & free() functions for dynamic memory allocation.
It supports object-oriented features such as classes, objects, polymorphism, encapsulation, inheritance.	It does not support object-oriented programming languages
It follows a bottom-up approach.	It follows a top-down approach.
It was developed by **Bjarne Stroustrup**.	It was developed by **Dennis Ritchie**.

46. What is a self-referential structure in C?

A self-referential structure in C is a structure having one or more pointers as a member pointing or referring to the structure of the same type. It is used in creating data structures such as linked lists, trees, and graphs.

For example,

```
  struct Node{
  int data;
  struct Node *link;
};
```

47. What is a static function in C?

A static function in C is a function defined using the static keyword in C. It is created when we want to restrict the use of a function within a file where it is declared.

Syntax

```
static return-type function_name(){
}
```

For example,

```
static void func()
{
  printf("Static function in C.");
}
```

48. How do variable declaration and definition differ from one other?

When a variable is declared, it informs the compiler of its data type and size, but when you define a variable in your code, it allocates memory to the variable.

We can declare a variable many times in a variable in a program however, a variable may only be defined once.

49. Write a program in C to print backslash (\) characters.

```c
int main()
{
  printf("\\");
  return 0;
}
```

50. Why do we use the arrow operator (->) in C?

We use the arrow operator (->) when a structure or union pointer variable is used to access structure or union members.

51. When we cannot use & (address-of) operator in C?

We cannot use & (address-of) operator with constants and variables declared using register storage class.

52. Is there any error in the following C program?

```c
int main()
{
  printf("Hello, ") , printf("World!");
  return 0;
}
```

No, there will be no error in the given program because the comma (,) operator is used for separating two or more expressions or variables. Hence, the output will be "Hello, World!".

53. Name the linker's output file.

The linker generates an executable output file.

54. Is it possible to do nested comments in C?

No, we cannot do this.

55. What are the various modes of file in C?

- r – The file has been opened to allow for reading.
- w – Opens a file so that it can be written into.
- a – Open for appending content at the end of the file.
- r+ – You can read and write to the file that's been opened.
- w+ – You can read and write to the file that's been opened.
- a+ – Opens a file for appending content to the end of the file.

56. What is the purpose of using the realloc() function in C?

realloc() function is used for resizing the memory block that is allocated dynamically.
Syntax

```
realloc(ptr, size)
```

The first argument, ptr, is a pointer pointing to the memory block allocated using either malloc() or calloc() and the second argument is the new size of a memory block.

57. Name the functions used in C file handling.

- fopen()
- fprintf()
- fscanf()
- fputc()
- fseek()
- fgetc()
- fclose()
- ftell()

58. Write a C program to print a string without double-quotes.

```
#include<stdio.h>
#define print_string(string) #string

int main()
{
  printf(print_string(UseMyNotes));
  return 0;
}
//Output
//UseMyNotes
```

59. When using C, how can a negative integer be stored?

- Given number excluding the negative sign is converted into binary equivalent.
- Now, we calculate the 2’s complement of a number.

60. Write a program in C to print the fibonacci series.

```
#include <stdio.h>
int main()
{
   int i, first=0, second=1, third, n;
   scanf("%d", &n);
   printf("Fibonacci Series\n");
   printf("%d %d", first, second);
   for(i=2;i<n;i++){
     third=first+second;
     printf(" %d", third);
     first=second;
     second=third;
  }
return 0;
}
//Output
//8
```

```
//Fibonacci Series
//0 1 1 2 3 5 8 13
```

I hope this module will be very beneficial for your interview preparation. Many other questions related to C are asked in the interview, but it cannot be discussed in a single article. For learning programming languages like C, Java you can study from our website, until then stay tuned with us for more informative and exciting modules like this.