



Most Trending C Programming Interview Question and Answer

in 2024



Most Trending C Interview Questions and Answers in 2024

Q.1 What is the difference between `typedef` and `#define`?

Ans. `typedef` helps you assign an alternative name for an existing data type. It does not introduce a new data type. We use it mainly while dealing with structures, unions, and enumerations to increase the readability and appeal of the code by reducing its complexity.

The syntax of `typedef` is:

```
typedef data_type new_data_type_name;
```

For instance,

```
typedef unsigned long int Big_Positive_Integer;
```

`#define`, on the other hand, is a preprocessor directive and it works by replacing the value before compiling the code. Since it simply gives an alias name for the macro, we do not have to specify the data type of variable we are creating.

The syntax of `#define` is:

```
#define VALUE_NAME value;
```

For instance,

```
#define MAXIMUM 10
```

Q.2 What is the difference between `long` and `double` data types?

Ans. Although, both are data types that have a seemingly similar purpose, but are different.

Double is used for storing floating-point values up to a higher precision than float, whereas long is used only for storing large integer value up to a higher precision than int.

Q.3 What is the advantage of using macros over functions?

Ans. We generally prefer macros before functions as macros are pre-processed. In simple words, macros in C are processed before the compilation of the program.

On the other hand, functions are not preprocessed but simply compiled.

The preprocessing of macros gives it an upper hand over functions.

Q.4 Is it possible to give arguments in the main() function?

Ans. Yes, it is possible to give arguments to the main function. This concept is called command line arguments.

Command line arguments in C are nothing but simply arguments that are specified after the name of the program in the system's command line, and these argument values are passed on to your program during program execution. Only two arguments can be passed to the main function called argc and argv.

Q.05 What is the use of a NULL pointer?

Ans. NULL pointers are used for the initialization of pointers, or to represent the end of a linked list of unknown length. It can also be used to indicate an error while returning a pointer from a function.

Q.06 What is the difference between lvalue and rvalue?

Ans. We use 'lvalue' to refer to the objects that appear at the left-hand side or right-hand side of the identifier. In contrast to this, we use 'rvalue' to refer to the data value that is stored at an address in memory. It is important to note that rvalue is an expression that cannot have a value assigned to it which means the rvalue appears on right but not on the left-hand side of an assignment operator "=".

Q.07 What is the use of new and delete operators in C?

Ans. The new operator in C is used for memory allocation. It indicates to return a pointer to the beginning of the new block of memory allocated.

For instance,

```
int *pointer;  
pointer = new int[10];
```

Here, 10 new blocks of memory are allocated with the help of the new keyword.

The delete operator is used for memory deallocation. We use it when we no longer require the variable to be stored in computer memory. It is responsible to free the memory so that it becomes available for other purposes.

For instance,

```
delete pointer;
```

Q.8 Out of the following data types, which are considered as primitive?

float
long
arrays
double
enumeration

Ans. The primitive data types are the fundamental data types available in the C programming language. Basically, they are data types that are not derived from any other data type. The primitive data types from the above list are a float, long and double.

Q.09 What are the conditions of binary search?

Ans. There are basically 2 conditions that need to be satisfied before we perform a binary search:

The array must be sorted either in ascending or descending order.

The lower and upper bound and the sort order of the array should be known.

Q.10 What could be wrong with your code if the function malloc() is declared as undefined by the C compiler?

Ans. You might have probably not included the header file <stdlib.h> while using malloc().

Q.11 What would be the output of the following code according to a 16-bit compiler?

1. #include <stdio.h>
2. int main()
3. {

```
4. int number = -10;  
5. printf("%d\n", sizeof(number));  
6. }
```

Ans. The output would be 2.

According to a 16-bit compiler, the memory occupied by the int data type is 2 bytes whereas the memory occupied by a 32-bit and 64-bit compiler is 4 bytes for the int data type.

Q.12 What are the different ways to sort numbers in C?

Ans. There are 2 ways to sort numbers in C:

1. By implementing user-defined sorting algorithms like the bubble sort, selection sort or insertion sort.
2. An inbuilt function in C is available to sort numbers called the qsort() function

Q.13 What is the difference between #define and const?

Ans. #define is a preprocessor directive that we use in order to define a macro containing some value or expression. This value or expression is substituted during program compilation.

Evidently, #define does not occupy computer memory, unlike variables.

For instance,

1. #include<stdio.h>
2. #define LIMIT 5;
3. int main()
4. {
5. int array[LIMIT];

7. }

We use the `const` keyword in the declaration of a variable. Unlike `#define`, it occupies computer memory to store a particular constant value. Once the value has been initialized, it cannot be further modified.

For instance,

```
const float pi = 3.14;
```

Q.14 Which header file would you require to access the following inbuilt functions:

isupper()

fgets()

strrev()

fabs()

Ans.

<ctype.h>

<stdio.h>

<string.h>

<math.h>

Q.15 Where is the deletion of elements done in a queue?

Ans. The deletion (dequeue) operation is performed from the front of the queue as it follows the FIFO (First In First Out) rule.

Q.16 What is the difference between `islower()` and `tolower()` function in C?

Ans. The `islower()` functions checks if the character or string has all its characters in lower case or not and returns an

integral value depending upon the validation of the condition. The `tolower()` function is used to convert a character or a string to lowercase form.

Q.17 What does the '\0' character indicate in a string?

Ans. The '\0' is a NULL character that indicates the termination of a string. It is automatically appended at the end of the string.

Q. 18 What is the maximum number of characters you can use to name a variable?

Ans. You cannot use more than 32 characters while naming a variable in C provided it follows the rules of naming an identifier.

Q.19 What is the use of comments in C?

Ans. Comments in C are a great way to express the logic behind certain statements useful for the programmer to bookmark the logic used by him.

Q.20 What is the difference between 'g' and "g" in C?

Ans. In C, 'g' is treated as a char data type as it is enclosed within single quotes. "g" is treated as a string data type (array of characters) which is terminated by a '\0' character as it is enclosed within double-quotes.

Q. 21 Is it possible to convert a queue into a stack? If yes, then how?

Ans. It is possible to convert a queue into a stack. 2 stacks would be required for the creation of a queue. It can be done in 2 ways:

1. By making the enqueue operation costly.

2. By making the dequeue operation costly.

Q. 22 Describe function pointers and how callbacks in C may be implemented using them.

Ans. Function pointers are pointers that direct users to a function's address. They may be used to construct callbacks, in which one function is called inside of another and received as an argument. This enables more extension and flexibility in the programming. Asynchronous programming and event handling both frequently employ callbacks. You may dynamically alter a function's behaviour to meet various needs by giving function pointers as arguments.

Q. 23 What distinguishes the C “malloc” and “calloc” functions, and when would one be preferable to the other?

Ans. In C, memory is dynamically allocated using the “malloc” and “calloc” functions. The primary distinction between the two is that “malloc” allocates memory that has not yet been initialised whereas “calloc” does the opposite. When you need to quickly allocate memory but don't require it to be initialised, use the “malloc” command. When the memory that has been allocated has to be initialised to zero or a certain value, use the “calloc” command.

Q. 24 Describe the distinction between “int* const ptr” and “const int* ptr” in C.

Ans. As a pointer to a constant integer, “const int* ptr” specifies that the value indicated by ptr cannot be changed. On the other hand, “int* const ptr” defines a constant pointer to an integer, meaning the value it points to can be changed but the pointer itself cannot be changed to reference a new memory address.