By OnlineInterviewQuestions.com

C Programming Interview Questions

C is the general and basic **programming language** that will create a base for other programming languages. C programming language was designed by **Dennis Ritchie** in Bells Lab. And it appeared around 46 years ago which is in 1972 and it was stably established on 11 December 2011. It is a crucial language of computer and it is coded in assembly language and it can run on from supercomputers to the embedded systems. ANSI C (American National Standards Institute) has standardized the C programming language since 1989 and even by the International Organization for Standardization (ISO).

Read Top C language interview questions and answers

Q1. What are static variables in C ?

C defines another class of variables called static variables. Static variables are of two types:

- **Static variables** that are declared within a function (function static variables). These variables retain their values from the previous call, i.e., the values which they had before returning from the function.
- **File static variables.** These variables are declared outside any function using the keyword static. File static variables are accessible only in the file in which they are declared. This case arises when multiple files are used to generate one executable code.

```
#include <stdio.h>
void PrintCount()
{
 static int Count = 1;
 //Count is initialized only on the first call
 printf( "Count = %d\n", Count);
 Count = Count + 1;
 //The incremented value of Count is retained
}
int main()
{
 PrintCount();
 PrintCount();
 PrintCount();
 return 0;
}OutPut:
Count = 1
Count = 2
Count = 3
```

The output of the program is a sequence of numbers starting with 1, rather than a string of 1?s. The initialization

of static variable Countis performed only at the first instance of the function call. In successive calls to the function, the variable count retains its previous value. However, these static variables are not accessible from other parts of the program.

Q2. What is a scope resolution operator in C ?

The scope resolution operator **(::)** is useful in C programming when both global and local variables have the same name, a statement using that name will access the local variable (more precisely, it will access the variable of that name in the innermost block containing that statement).
The scope resolution operator represented as **::** (a double colon) can be used to select the global variable explicitly.

Consider the example below.

```
#include<stdio.h>
int x = 10;
int main()
{
  int x = 20;
  printf("%d\n",x);
  printf("%d\n", ::x);
  return 0;
}OutPut:
20
10
```

Q3. What is register variable in C language ?

C language allows the use of the prefix register in primitive variable declarations. Such variables are called register variables and are stored in the registers of the microprocessor. The number of variables which can be declared register are limited. If more variables are declared register variables, they are treated as auto variables. A program that uses register variables executes faster as compared to a similar program without register variables. It is possible to find out the allocation of register variables only by executing and comparing the performance with respect to the time taken by the program (perceptible in large programs). It is the responsibility of the compiler to allow register variables.

In case the compiler is unable to do so, these variables are treated as auto variables. Loop indices, accessed more frequently, can be declared as register variables. For example, index is a register variable in the program given below.

//Illustration of register variables in C language

```
#include <stdio.h>
#include <string.h>
int main()
{
  char Name[30];
  register int Index;
```

```
  printf( "Enter a string: ");
  gets( Name );

  printf( "The reverse of the string is : ");
  for( Index = strlen( Name ) - 1; Index >= 0; Index-- )
   printf( "%c", Name[Index]);
  printf( "\n" );
  return 0;
}OutPut:
Enter a string: Apple is Red
The reverse of the string is : deR si elppA
```

Q4. What is a structure in C Language.How to initialise a structure in C?

A structure is a composite data type declaration that defines a physically grouped list of variables to be placed under one name in a block of memory, allowing the different variables to be accessed via a single pointer.

Defining a structure in C: In C language a structure is defined using struct keyword followed by variable or pointer name below is the basic syntax for declaring a structure in C language.

```
struct tag_name {
   type member1;
   type member2;
   /* declare as many members as desired, but the entire structure size must be
};
```

Q5. What is an auto variable in C ?

All variables declared within a function are auto by default. The extent of a variable is defined by its scope. Variables declared auto can only be accessed only within the function or the nested block within which they are declared. They are created when the block is entered into and destroyed when it is exited. i.e., memory is allocated automatically upon entry to a block and freed automatically upon exit from the block and note that default value is a garbage value.

Q6. What is the use of extern in C?

Global Static variables are global to the file in which they are defined. They are used when the same global variable is referenced in each of the files and these variables must be independent of each other across the files. The use of global variables is not recommended, as function independence is one of the basic idea of modular programming. Global variables should be used only when it is inevitable (i.e., when all the functions need a particular variable).

When program spans across different files and we want to have a global variable accessible to all functions in these files, the keyword extern should be used before the data type name in the declarations in all these files

where it is accessed except one. The linker requires that only one of these files have the definition of the identifier. Global variables definitions can occur in one file only.

Consider the following program example to understand better the use of keyword extern,

```
/* file sub_file.c Defines the function sub_func */
extern int globalVar;

void inc_global()
{
  globalVar++;
}
```

The code of the second file, mainfile.c is given below,

```
/*file mainfile.c Defines the function main and uses the function
inc_global. Compile with cc mainfile.c sub_file.c */
int globalVar;

void main()
{
 /*Assign something to globalVar*/
 globalVar = 10;

printf( " globalVar before calling = %d\n", globalVar);
inc_global();
printf( " globalVar after calling = %d\n", globalVar);
}
```

Observe the integer **globalVar**has been declared in the sub_file.c as **extern globalVar;**

Upon encountering the above statement, the compiler knows that an integer**globalVar**exists. It goes ahead to produce the object file. In the file mainfile.c, the actual definition of **globalVar** occurs with the statement. **int globalVar;**

Note that no special keywords are necessary to say that **globalVar**spans across files. This is because a declaration such as the one shown above tells the compiler that the integer**globalVar**is exported and can be accessed in different files.

The linker is then invoked to link the two object together. When the compiler produces the object file of sub_file.c, it does not know the exact location of the variable **globalVar**. So, it includes the information in the locations in the object file where the global variable is accessed. In the object file of mainfile.c, the compiler will store this information (that a global integer called **globalVar**exists, that can be used across files). When the linker links these two object files together, it knows the location of**globalVar**from the object file of mainfile.c. It uses this information to complete the object file of sub_file.c, where the compiler had encoded the information where **globalVar**is accessed in sub_file.c

Q7. What is the difference between #include <header file> and #include “header file” ?

- #include “myheaderfile.h”
- #include <headerfile.h>

The difference between these two is the location the compiler searches for the header file to be included. If the file name is enclosed in quotes, the compiler searches in the same directory in which the file is included. This method is normally used to include programmer defined headers. If the file name is enclosed in brackets which are used for standard library headers; the search is performed in an implementation dependent manner, normally through pre-designated directories. These both file inclusions one using angle brackets and the other using quotes in an include statement can be used in C and C++.

It does:

“path/myheaderfile.h” is short for ./path/myheaderfile.h

is short for /path/headerfile.h

Q8. What are Derived data types in C ?

Derived data types are object types which are aggregates of one or more types of basic data types. below are the list of derived datatype in C Language.

- Pointer types
- Array types
- Structure types
- Union types
- Function types

Q9. Explain C preprocessor ?

The C preprocessor or cpp is the macro preprocessor for the C and C++ computer programming languages. The preprocessor provides the ability for the inclusion of header files, macro expansions, conditional compilation, and line control.

Q10. What is recursion in C ?

Recursion: A function, which calls itself, recursion is simple to write in program but takes more memory space and time to execute.

Advantages of using recursion:-

- Avoid unnecessary calling of functions
- Substitute for iteration
- Useful when applying the same solution

Factorial program in c using Recursion

```
#include
int factorial(unsigned int i) {
   if(i <= 1) {
      return 1;
   }
   return i * factorial(i - 1);
}
int  main() {
   int i = 15;
   printf("Factorial of %d is %d\n", i, factorial(i));
   return 0;
}
```

Q11. Explain Enumerated types in C language?

Enumerated types are used to define variables that can only assign certain discrete integer values throughout the program.

Enumeration variables are variables that can assume values symbolically

Declaration and usage of an Enumerated variable.

```
enum boolean{
false;
true;
};
enum boolean
```

Q12. Differentiate call by value and call by reference ?

Call by value:
A process in which the values of the actual parameters sent by the calling function are copied to the formal parameters of the called function.

Call by reference:
A process in which the parameters of a calling function are passed to the parameters of the called function using an address.

Q13. List some basic data types in C ?

In Programming Languages data types are used to define a variable before its use.

Below are list of some basic data types in C language.

- **Integer:** used to define integer numbers, denoted as int
- **Floating point:** decimal
- **Character:** defines character
- **Void:** void type has no value and only one operation: assignment. Plays a role of generic data type.

Q14. What is typecasting?

Typecasting is a way to convert a variable/constant from one type to another data type.

Q15. Explain about block scope in C ?

A block is defined as a sequence of statements enclosed between curly braces, { and }. In other words, a block is a compound statement. For example, a function body is a block, because it is simply a sequence of statements enclosed within curly braces. Blocks of statements in if statements and loops. All these blocks of statements actually follow the same rules.

Consider the example shown below, the variable j declared in both main and the user defined function other func. Accessing j uses the local declaration in the called function.

```
//Illustration of block scope
#include <stdio.h>
void OtherFunc( void )
{
 int i = 10;
 printf(" Inside the function OtherFunc, the value of i is %d\n", i);
}
int main()
{
  int i = 20;
  printf(" Inside the function main, the value of i is %d\n", i);
  OtherFunc();
  return 0;
}OutPut:

Inside the function main, the value of i is 20
Inside the function OtherFunc, the value of i is 10
```

The variables defined can be accessed only within the block in which they are declared. In cases of nested blocks, the variables declared in the outer blocks are accessible by statements in the inner blocks, and not vice-versa. These variables are called local variables because they are localized to the block. It helps to prevent the integrity of data (data of one function cannot be modified by another function, directly). It can be observed that if two variables of the same name are declared in many functions, they are distinct and unrelated variables. The scope of the variables in the function parameter list is also confined to the function, i.e., they are also local variables.

Q16. Explain continue keyword in C

Continue is a jump statement that transfers control back to the beginning of the loop, bypassing any statements that are not yet executed. It can be used only in an iteration statement.

Q17. Compare array data type to pointer data type

Array is a collection of variables of same type that are referred through a common name and a **pointer** is a variable that holds a memory address. pointers can point to array and array to pointers

Q18. What are bit fields in C ?

Bit fields are used to store multiple, logical, neighboring bits, where each of the sets of bits and single bits can be addressed.

Q19. What are different storage class specifiers in C

auto, register, static, extern are storage class specifiers in C

Q20. What is NULL pointer?

NULL is used to indicate that the pointer doesn't point to a valid location. Ideally, we should initialize pointers as NULL if we don't know their value at the time of declaration. Also, we should make a pointer NULL when memory pointed by it is deallocated in the middle of a program.

Q21. Explain main function in C ?

Main Function is the function where every C program begins executing. it will usually call other functions to help perform its job, some that you wrote, and others from libraries that are provided for you.

Q22. What does printf does ?

The printf is a library function that prints output. requires the \n newline character, even separately like printf(“\n”);. The first argument is the string of characters to be printed, with each % indicating where one of the other arguments is to be substituted, and in what form it is to be printed.

Q23. List some applications of C programming language?

Application of C Programming Language

- To develop embedded software
- It is to create a computer application
- It is effective to create a compiler for various computer languages to convert them into low-level language that is the machine understandable language.
- It can be used to develop an Operating system and UNIX is one which is developed by the C .programming language.
- It is used for creating software for various applications and even hardware.

Q24. Write program to remove duplicate in an array ?

C program to remove duplicate programme:

```
#include <stdio.h>
int main(){
  int n, a[100], b[100], calc = 0, i, j;
  printf("Enter no. of elements in array\n");
  scanf("%d", &n);
   printf("Enter %d integers\n", n);
   for (i = 0; i < n; i++)
    scanf("%d", &a[i]);
   for (i = 0; i<n; i++) {
    for (j = 0; j < calc; j++) {
      if(a[i] == b[j])        break
;   }
    if (j== calc){
      b[count] = a[i];
      calc++;  }  }
   printf("Array obtained after removing duplicate elements:\n");
   for (i = 0; i < calc; i++)
    printf("%d\n", b[i]);
  return 0;}
```

Q25. Explain Pointers in C programming?

Pointers are variables that are used to store addresses. The concept of the pointer is considered to be one of the difficult part of learning the C and C++ programming languages. There are several easy ways to write programs

without pointers, but in case of dynamic memory allocation, the knowledge of pointers is a must.

Knowing about memory locations and addresses defined will enable you with the ideas of how every variable function in a program.

Q26. How can you find the exact size of a data type in C?

One can determine the exact size of a data type by using the sizeof operator. The storage size of the data type is obtained in bytes by using the syntax: sizeof**(data_type)**.

Q27. If the size of int data type is two bytes, what is the range of signed int data type?

The range of signed int data type if from **-32768** to **32767**

Q28. What do you understand by normalization of pointers?

Normalization is the process by which an address is converted to a form such that if two non-normalized pointers point to the same address, they both are converted to normalized form, thereby having a specific address

Q29. What is the best way to store flag values in a program?

Flag values are used to make decisions between two or more options during the execution of a program. Generally, flag values are small (often two) and it tries to save space by not storing each flag according to its own data type.

The best way to store flag values is to keep each of the values in their own integer variable. If there are large number of flags, we can create an array of characters or integers. We can also store flag values by using low-order bits more efficiently.

Q30. Explain bit masking in C?

Bit masking refers to selecting a particular set of bits from the byte(s) having many values of bits. Bit masking is used to examine the bit values and can be done by 'AND' operation of byte, bitwise.

Q31. Is there any demerits of using pointer?

Yes. As pointers have access to a particular memory location, the security level decreases and restricted

memory areas can be accessed. Other demerits include memory holes, process and memory panics, etc.

Q32. Is there any data type in C with variable size?

Yes, Struct is one of the data type in C that have variable size. It is because the size of the structure depends on the fields which can be variable as set by the user.

Q33. What is a void pointer?

Void **pointer** is a generic pointer in programming. If the pointer type is unknown, we make use of the void pointer.

Q34. When can you use a pointer with a function?

A pointer can be used with a function-

- When an address is to be passed to a function
- When an array elements are to be accessed through a function. Passing base address will give access to the whole array.

Q35. When can a far pointer be used?

Sometimes the task we are required to do might not fit in the allocated data and code segments. Far pointers help to access rest of the memory inside a program. Far pointers are the information present outside the data segment (generally 64 kb). Such pointers are used when we need to access an address outside of the current segment.

Q36. What is difference between structure and union?

The differences between structure and union are as follows:

Structure	Union
The keyword struct is used to declare a structure.	The keyword union is used to declare a union.
Individual members can be accessed at a time.	Only one member accessed at a time.
Several members of a structure can initialize at once.	Only the first member of a union can be initialized.
Altering the value of a member will not affect other members of the structure.	Altering the value of any of the member will alter the other member values.
Each member within a structure is assigned a unique storage area of location.	Memory allocated is shared by the individual members of the union.

Q37. What is the differences between constant pointer and constant variable?

The differences between a constant pointer and the constant variable are as follows:

Constant Pointers :	Constant variable
Constant Pointer is the one who cannot change the address they are pointing to. This means that suppose there is a pointer that points to a variable (or stores the address of that variable).	The constant variable is a value that can not be altered throughout the program. A storage location paired with an associated symbolic name has a value. It is similar to a variable but cannot be modified once defined. It is sometimes known as a controlled variable. It is used in every scientific experiment that is helpful to understand the conclusions of an experiment.
A constant pointer is declared as follows: 'int *const ptr' The location of the 'const' keyword makes the pointer 'ptr' a constant pointer.	A Variable can be declared as constant with the use of the “const” keyword before the data type of the variable. It can be initialized only once. The default value of constant variables is zero. A Constant variable is declared as follows: const int a;

Q38. What is a palindrome number?

Q39. What is AVL tree?

AVL tree is a binary search tree where the node difference of heights in either the right and left subtrees is less than or equal to one. Three developers by name Adelson, Velskii and Landi developed the technique for balancing the binary trees height. From the first letter of their names, (AVL) came to be. AVL tree has advantages such as low time complexity in inserting and deleting operations.

Q40. What is array in C?

An array is a collection of the same data items, which are stored in a contagious memory location.