

[Career Guide](#)[Project Ideas](#)[Interview Questions](#)[Great Learning](#) > [Blog](#) > [IT/Software Development](#) >

Table of Contents

[C BASIC INTERVIEW QUESTIONS](#)[C Interview Questions for Freshers](#)[C Technical Interview Questions](#)[C data structure Interview Questions](#)[C Programming Interview Questions](#)[FAQs](#)

100+ C Interview Questions and Answers for 2025

By [Tanuja Bahirat](#) / Updated on Nov 18, 2024

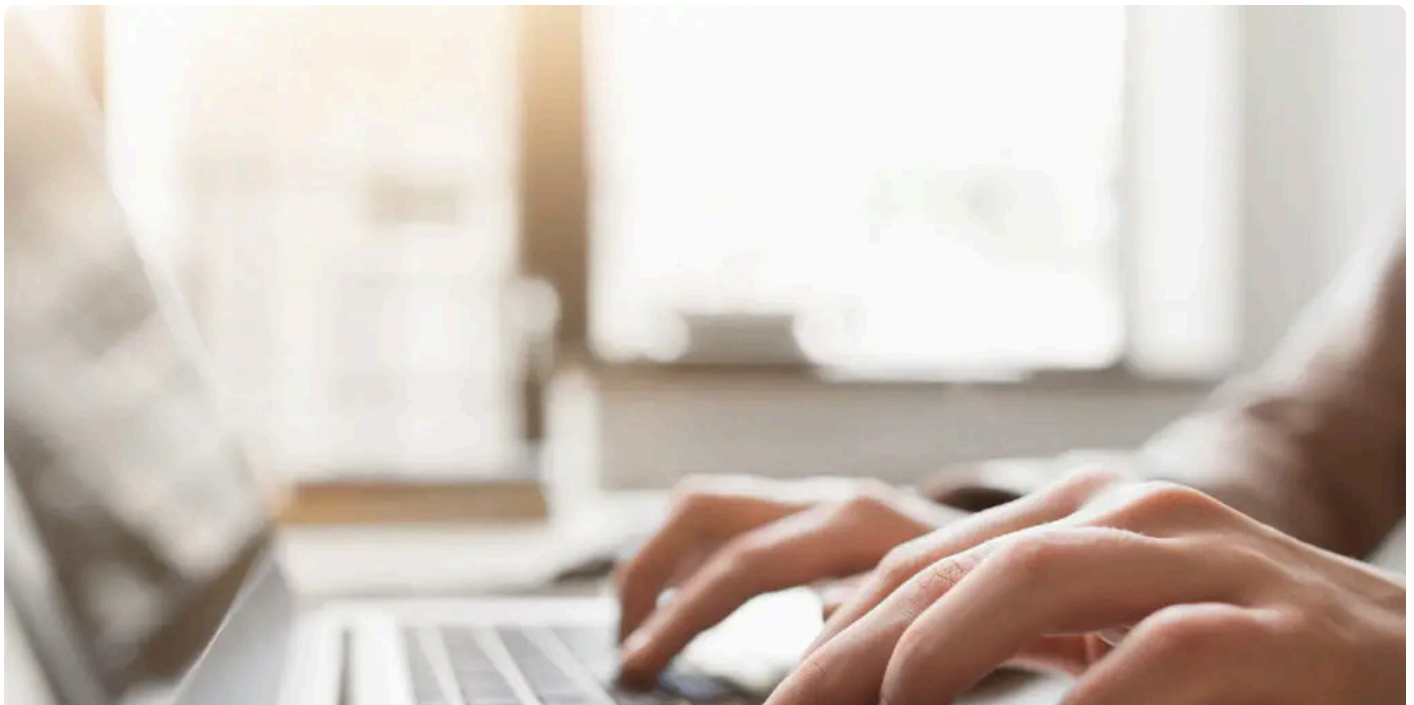


Table of contents



C programming is a general-purpose language computing programming language and is essential if you want to work in the software development domain. The first set of questions and answers are curated for freshers, but the blog will cover C Interview Questions for experienced, technical interview questions, C data structure interview questions and more. These questions cover all the basics of C and will help showcase your expertise in the subject. The questions are divide into groups as follows.

- [C Basic Interview Questions](#)
- [C Interview Questions for Freshers](#)
- [C Technical Interview Questions](#)
- [C data structure Interview Questions](#)
- [C Programming Interview Questions](#)
- [FAQs](#)

C BASIC INTERVIEW QUESTIONS

1. Can you tell me how to check whether a linked list is circular?

First, to find whether the linked list is circular or not, we need to store the header node into some other variable and then transverse the list. A linked list is not circular, if we get null at the next part of any node and if the next node is the same as the stored node, then it is a circular linked list. An empty linked list is also considered a circular linked list.

Algorithm to find whether the given linked list is circular.

An algorithm to determine whether the linked list is circular or not is a very simple way.

Algorithm: –

1. Traverse the linked list.
2. Check if the node is pointing to the head or not.
3. If yes then the linked list is circular.

2. What is the use of a semicolon (;) at the end of every program statement?

A semicolon (;) is used to mark the end of a statement and begin a new statement in the C language. It ensures proper termination of the statement and removes confusion while looking at the code. They are not used between control flow statements but are used to separate conditions in the loop. They are end statements in C programming.

C interview Questions PDF

3. Differentiate Source Codes from Object Codes

Source Code: –

1. Source code is in the form of Text.
2. Created by the Programmer.
3. Human readable.
4. Can be changed over time.
5. Not system specific
6. Input to the compiler
7. Not CPU executable

Object Codes: –

1. Object code is in the form of Binary Numbers.
2. Created by the Compiler.
3. Machine Readable.

4. Needs to compile the Source Code each time a change is to be made.
5. System specific
6. Output of the compiler
7. CPU executable

4. What are header files and what are their uses in C programming?

A header file in C contains predefined standard library functions and declarations to be shared between several source files.

file with extension .h. There are two types of Header files: – Files that the programmer writes and the files that come with your compiler. Header files are library files of your C program. Including a header file means using the content of the header file in your source program. The basic syntax of using header files is: – `#include <file>` or `#include "file"`.

Uses of header files: – Header files contain a set of functions that are used to write a program in C.

5. When is the "void" keyword used in a function?

In programming, when you declare functions, you first decide whether that function will declare a value or not. "void" when placed at the leftmost part of the function header, the function will not return a value. The function will only display some outputs on the screen. And in the second case, after the function executes when the return value is expected then only the data type of the return value is placed instead of "void".

6. What is dynamic data structure?

Data Structures are of two types. Static Data Structure and Dynamic Data Structure. While performing operations, the size of the structure is not fixed and can vary according to the requirement in the dynamic data structure. This data structure is designed to facilitate the change of data structures in the run time. In DDS, the data in memory has the flexibility to grow or shrink in size. This data structure can "overflow" if it exceeds its limit.

Add Two Numbers Without Using the Addition Operator.

To add two numbers without using the Addition Operator: –

1. Take three variables num1, num2, and x.
2. Take input from the user.
3. Store it in num1 and num2.
4. Increase num1 using the 'for' loop.
5. Till the count of the second number for loop will call and, in each call, we will increase the value of num1 by 1.
6. After addition, the result will be stored in num1.

Using this method, we can calculate the sum of two numbers.

Subtract Two Number Without Using Subtraction Operator

To subtract two numbers without using the Subtraction Operator: –

1. Take four variables num1, num2, complement, result.

2. Take input from the user
3. Store it in num1 and num2.
4. Take two's complement of a num2 number.
5. Add num1 number with two's complement.
6. Finally, the difference will be stored in the result.

Using this method, we can easily perform subtraction of two numbers

Multiply an Integer Number by 2 without Using a Multiplication Operator.

We can use multiple approaches to multiply an integer number by 2 without using the Multiplication operator.

1. We can easily calculate the product of two numbers using the recursion method and without using the * multiplication operator.
2. Also, we can get the product of two numbers without using the * the multiplication operator is using an iterative method.

Check whether the number is EVEN or ODD, without using any arithmetic or relational operators

Even or Odd operation can be performed using the following 3 methods: –

Method 1: Using the bitwise (&) operator.

Method 2: Multiplying and dividing the number by 2.

First, divide the number by 2 and multiply it by 2. If the result is the same as that of the input, then it is an even number otherwise it is an odd number.

Method 3: Switching temporary variable n times with the initial value of the variable being

true. If the flag variable gets its original value (which is true) back, then n is even.

Else, it is false.

Check for Balanced Parentheses using Stack

1. First we need to declare a character stack S.
2. Then traverse the expression string exp.
 - a) If the character is a starting bracket '(' or '{' or '[' then push it to stack.
 - b) If the character is a closing bracket ')' or '}' or ']' then pop from the stack.
3. If the popped character is the matching starting bracket, then fine else brackets are not balanced.
4. If there is some starting bracket left after complete traversal in the stack, then it's "NOT BALANCED".

C Interview Questions for Freshers

1. What is function in C?

A function can be defined as a block of statements which are written to perform a particular task, improve code readability and reusability. The functions can be of two types: Predefined functions and User-defined functions.

Predefined functions are in-built functions which are already present in C library. For example, **printf()**

A user defined function is declared and defined by the programmer for customized functions. The syntax for the same is:

```
1  return_type_of_function <function name> (argument list)
2  {
3      Statements
4  }
```

2. What is recursion in C?

When a function calls itself in its definition then the process is called as Recursion. For example,

```
1  int addition (int n)
2  {
3      if (n <= 1) // base case
4          return 1;
5      else
6          return n + addition(n-1);
7  }
```

3. What is structure in C?

A structure in C is a data type that allows you to list a set of variables within it that can only be accessed by this data type's object. Structure allows us to emulate data with varied elements.

4. How to declare string in C?

The syntax of declaring a string in C is as follows:

```
1 | char <string name> [size of the string];
```

For example: `char dog_breed [100];`

5. What is string in C?

String is an array of characters whose index starts from 0 and it is terminated with a special character `"`.

6. What is segmentation fault in C?

A segmentation fault in c is a fault that happens when someone tries to access a forbidden memory location or makes use of a technique that is forbidden in the system on which they are working on.

7. What is variable in C?

A variable in c is created in c to store data by reserving memory in the system. You can have different types of variables that store different types of data. for eg. You can store integer data in 'int' variable or characters in 'char' variables.

8. What is variable in C?

A variable in c is created in c to store data by reserving memory in the system. You can have different types of variables that store different types of data. for eg. You can store integer data in 'int' variable or characters in 'char' variables.

9. How to take string input in C?

There are multiple ways to take a string input in C, such as:

– gets:

```
1 | char yourname[20];  
2 |     printf("Enter your name:");  
3 |     gets(yourname);
```

– Using %s in scanf:

```
1 | char yourname[20];  
2 |     printf("Enter your name:");  
3 |     scanf("%s", yourname);
```

10. What is macro in C?

Macro in C is basically used to replace a block of code in C. You do this by defining it in the start of the program as such:

```
1 | #define <name of the macro> <code to be replaces>
```

Eg. #define k 1.44

Here, whenever you use the macro 'k' in the code the compiler will replace it with the value '1.44'.

11. What is union in C?

A union in C is a data type that allows you to list & define a set of variables within it that can only be accessed by this data type's object. A union is actually very similar to a structure with the one of the difference being that all the elements defined within a union share a memory location whereas in a structure the elements do not share a memory location.

```
1 | union fun{  
2 |  
3 |     int a;  
4 |     char b[10];  
5 |  
6 | } funobj;
```

12. How to find prime numbers in C?

A natural number greater than 1 and not a product of two smaller natural numbers other than itself and 1 is called Prime number. For example, 5 (factors of 5 is 1 and 5) is a prime number. The C code to find if the entered number is prime or not is as follows.

```
1  #include <stdio.h>
2  int main() {
3      int num, j, f = 0;
4      printf("Enter a positive integer");
5      scanf("%d", &num);
6
7      for (j = 2; j <= num / 2; ++j) {
8          if (num % j == 0) {
9              f = 1;
10             break;
11         }
12     }
13     if (num == 1) {
14         printf("1 is neither prime nor composite.");
15     }
16     else {
17         if (f == 0)
18             printf("%d prime number", num);
19         else
20             printf("%d not prime number", num);
21     }
22
23     return 0;
24 }
```

13. What is Armstrong number in C?

If the integer is in the form:

$abc = a*a*a + b*b*b + c*c*c$, it is called the Armstrong number.

The code to find whether the entered number is Armstrong or not is as follows.

```
1  #include <math.h>
2  #include <stdio.h>
3
4  void main() {
```

```
5  int num, orig_num, rem, i = 0;
6  float res = 0.0;
7
8  printf("Enter an integer: ");
9  scanf("%d", &num);
10
11  orig_num= num;
12
13  for (orig_num = num; orig_num!= 0; ++i) {
14      orig_num/= 10;
15  }
16
17  for (orig_num = num; orig_num!= 0; orig_num/= 10) {
18      rem = orig_num% 10;
19      res += pow(rem, i);
20  }
21
22  if ((int)res == num)
23      printf("%d is an Armstrong number.", num);
24  else
25      printf("%d is not an Armstrong number.", num);
26 }
```

14. What are pointers in C?

Pointers in C are a data type that store the memory address of another variable.
For eg.

```
1  char *str = "Hi, How are you?";
2      Here the pointer variable *str points to the string '
3
4      or
5
6      int age;
7      int *int_value;
8
9      *int_value = &age;
10
11     printf("Enter your age please:");
12     scanf("%d", age);
13
14     printf("\n Your age is:%d", *int_value);
15
16     // this will print your age as the variable is pointi
```

15. How to round off numbers in C?

The process of making a number simpler to use but keeping the new value close to the old one is called rounding off. For example, 1.76521 round off to 1.7 or 1.76. In C to do the same we have round() function. The usage of the function is depicted in the code below.

```
1  #include <stdio.h>
2  #include <math.h>
3  void main()
4  {
5      float n, round_n;
6
7      printf("Enter a number ");
8      scanf("%f", &n);
9
10     round_n = round(n);
11
12     printf("\n The Round Value of %.2f = %.4f n", n, round_r
13 }
```

16. What is preprocessor in C?

A preprocessor in c is basically used to replace code with values you assign. And this process happens at the start of the program execution hence the name pre-processing.

Eg:

```
1  #define k 1.44
2
3  #include <stdio.h>
```

17. What is identifier in C?

In identifiers are basically any name user defined name you give to any data type, label, variable or function. Eg. int hello; here hello is an identifier. But there are rules to how you can name an identifier, such as:

The first letter of the identifier has to be an alphabet or an underscore.
You can only use alphabets, underscore or digits to write an identifier.
There cannot be any whitespace within an identifier, otherwise you will get an error.

18. What is a function in C?

A function in c is a block of code that can be referenced from anywhere in the system and that serves a specific purpose. Eg:

```
1  int fun() {  
2      int a = 11;  
3      return 11;  
4  }  
5  
6  int main() {  
7  
8      int b = fun();  
9  }
```

19. What is enum in C?

Enum or Enumeration in C is data type that you can use to assign values to an integral constant. Eg. enum fruits{apple, banana, pineapple};

20. What is stack in C?

A stack in c is a data structure that can be used to store data. as the name suggests a stack allows you store data in the form of a stack – one top of another. You can only insert and remove data from one side that is the top of the stack. We can use operations such as pop(),push() with a stack.

21. What is loop in C?

A loop in C is a set of code that allows the user to repeatedly perform a certain action until the set condition is fulfilled.

22. What is eof in C?

Eof in C defines the end of file, i.e. where the file that you are reading ends.

23. Who invented C?

Dennis Ritchie invented C in 1972.

24. How to return an array in C?

```
1  "#include <stdio.h>
2
3  int fun() {
4      int int_value[2]={11,22};
5      return int_value;
6  }
7
8  int main() {
9      int a = 10;
10     int *ptr;
11     ptr = fun();
12
13 }
```

25. What is #include in C?

#include is a pre-processor statement which makes sure about the availability of all the declarative statements and related function calls.

26. What is file in C?

Files in C are a set of bytes that are reserved for storing related data. It's different than variables as it is used to store data permanently.

27. How to learn C?

To learn C, start with why you should learn C and the features of this language. Then make sure you have the environment set for the same. About programmi

start with basic syntax, data types, data structure and control structure. After learning basics, then jump to advance concepts.

28. How to call a function in C?

There are two ways in which C function can be called from a program.

1. Call by Value: In this method, the value of the actual parameter can not be modified by formal parameter as both the parameters have different memory allocated to them.
2. Call by Reference: In this method, the value of the actual parameter will be modified by formal parameter as both the parameters have the same memory allocated to them.

```
1  #include<stdio.h>
2  // call by reference
3  void swap_ref(int *a, int *b);
4  //call by value
5  void swap_val(int a, int b);
6  int main()
7  {
8      int s = 2, n = 4;
9      // calling swap function by reference
10     printf("values before swap m = %d n and n = %d", s, n);
11     swap_ref(&m, &n);
12     // calling swap function by value
13     printf(" values before swap  m = %d nand n = %d", s, n)
14     swap_val(s, n);
15 }
16
17 void swap_ref(int *a, int *b)
18 {
19     int t;
20     tmp = *a;
21     *a = *b;
22     *b = t;
23     printf("values after swap a = %d and b = %d", *a, *b);
24 }
25
26 void swap_val(int a, int b)
```

```
27 {  
28     int t;  
29     tmp = a;  
30     a = b;  
31     b = t;  
32     printf(" values after swap m = %d and n = %d", a, b);  
33 }
```

29. What is static variable in C?

The variable that is defined only once and the compiler retains its value even after they exit the scope are called Static Variables. They are declared using static keyword. The syntax is:

static data_type variable_name = variable_value;

For example: static int c=9;

30. Who invented C language?

Dennis M. Ritchie at Bell Laboratories (formerly AT&T Bell Laboratories) in 1972 invented the C programming language.

31. How to check if a number is a perfect square in C?

You can check if a number is a perfect square by two ways.



1. Using functions from math.h library.
2. Using for and if loop

The sample code using math.h is as follows:

```
1 #include <stdio.h>  
2 #include <math.h>  
3 void main()  
4 {  
5     int n,i;
```

```
6   float f;
7   printf("Enter a number");
8   scanf("%d", &n);
9   f=sqrt((double)n);
10  i=f;
11  if(i==f)
12  printf("%d is a perfect square",n);
13  else
14  printf("%d is not a perfect square",n);
15  }
```

16

17 The sample code **using** loops is as follows:

18

```
19 #include<stdio.h>
20 int main()
21 {
22     int i, n;
23     printf("Enter a number");
24     scanf("%d", &n);
25     for(i = 0; i <= n; i++)
26     {
27         if(n == i*i)
28         {
29             printf("%d is a perfect square", n);
30         }
31     }
32     printf("%d is not a perfect square", n);
33 }
34 }
```

32. How to find power of a number in C?

The power of a number can be calculated by using pow() function of math.h library. For example:

```
1   #include <math.h>
2   #include <stdio.h>
3
4   void main() {
5       double x, y, res;
6       printf("Enter a base number");
7       scanf("%lf", &x);
8       printf("Enter an exponent");
9       scanf("%lf", &y);
10
11     res = pow(x, y);
```

```
12
13     printf("%.2lf", res);
14 }
15
16 This can be done using a while loop also.
17 #include <stdio.h>
18 void main() {
19     int x, y;
20     long long res = 1;
21     printf("Enter a base number: ");
22     scanf("%d", &x);
23     printf("Enter an exponent: ");
24     scanf("%d", &y);
25
26     while (y != 0) {
27         res *= x;
28         --y;
29     }
30     printf("%lld", res);
31 }
```

33. What is compiler in C?

Programmers write code in high level languages and a special program called compiler converts that high level code in machine language.

34. What are identifiers in C?

Identifier refers to unique names given to entities such as variables, functions, structures etc.

- Rules for naming identifier:
- Can have letters, digits, and underscores
- First letter should either alphabet or underscore
- Cannot use keywords as identifiers

35. How to use switch case in C?

A switch statement allows a variable to be tested against different values. Each of these values is called a case. Syntax for the same is:

```
1  switch(expression)
2  {
3      case constant_1 :
4          // Code to be executed if expression == constant_1
5          break;
6
7      case constant_2 :
8          // Code to be executed if expression == constant_2;
9          break;
10
11     default : // the default case is optional
12         // Code to be executed if none of the cases match.
13 }
```

36. How to convert decimal to binary in C?

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  void main()
4  {
5      int i[10],num,j;
6      printf("Enter the number");
7      scanf("%d",&num);
8      for(j=0;n>0;j++)
9      {
10         i[j]=num%2;
11         num=num/2;
12     }
13     printf("Binary is");
14     for(j=j-1;j>=0;j--)
15     {
16         printf("%d",i[j]);
17     }
18
19 }
20 "
```

37. What is linked list in C?

A linked list is a data structure in which nodes are dynamically allocated and arranged where each node contains one value and one pointer. Syntax is as follows:

```

1  struct node {
2      int d;
3      struct node *n;
4  }

```

38. In C, if you pass an array as an argument to a function, what actually gets passed?

When array is passed as argument, then the base address of the array is passed to the function.

39. How to use gets in C?

Gets is a function that reads a line from stdin and stores it in string.

```

1  #include <stdio.h>
2
3  void main () {
4      char s[10];
5
6      printf("Enter a string");
7      gets(s);
8
9      printf("%s", s);
10 }

```

40. What is extern in C?

Extern often referred to as global variables are declared outside the function but are available globally throughout the function execution. The value of this variable can be changed/modified and the default value is zero. These are declared using the "extern" keyword.

41. How to write a C program?

First the appropriate libraries are included using pre-processor. For example, `#include<stdio.h>`. After this, `main()` function is written which is the entry point of

every program. Inside the main() function code blocks are written using various functions and variables.

Sample Hello World code is as follows:

```
1  #include <stdio.h>
2  void main()
3  {
4      printf("Hello C Language");
5  }
```

42. What is token in C?

Smallest elements of a program are called tokens such as keywords, identifiers, strings etc.

43. What is volatile in C?

Volatile is a keyword in C which when applied to a variable at the time of declaration tells the compiler that the value of that variable will change at any time. Syntax for declaring a variable as volatile is as follows:

```
volatile uint8_t a;
```

44. What is argument in C?

In C99 Standard, an argument is defined as follows. "An argument is an expression in the comma-separated list bounded by the parentheses in a function call expression, or a sequence of preprocessing tokens in the comma-separated list bounded by the parentheses in a function-like macro invocation." To explain it through code. Suppose a function:

```
1  sub(int a, int b)
2  {
3  ...
```

```
4 | }
```

And while calling the function- `sub(5,4)` – here 5 and 4 are the arguments of the function `sub()`.

45. How to solve increment and decrement operators in C?

Increment (`++`) and decrement (`--`) are unary operators in C. These operators increase and decrease their values by 1.

`++a` is same as `a=a+1`

`--a` is same as `a=a-1`

```
a = 5;
```

```
b = 8;
```

```
c = ++a + b++;
```

```
c=++a + 8;
```

```
c=6+8= 14
```

46. What is an algorithm in C?

A sequential and well defined instructions for problem solving is called an algorithm. For example, a sample algorithm for subtracting two numbers.

Step 1: Start

Step 2: Declare variables `a`, `b` and `sub`.

Step 3: Read values `a` and `b`.

Step 4: Add `a` and `b` and assign the result to `sub`. `sub←a – b`

Step 5: Display `sub` Step 6: Stop

47. What is the use of C language?

The benefits of C language:

- C has rich library with a large number of inbuilt functions.
- C supports dynamic memory allocation.
- C facilitates fast computation in programs.
- C is highly portable. – C is case sensitive language.
- C is used for scripting applications.
- C allows free movement of data across the functions.

Due these benefits of C, it can be used/it has the following applications:

- Database systems
- Graphics packages
- Word processors
- Operating system development
- Compilers and Assemblers
- Interpreters

48. How to store values in array in C?

Sample codes for storing values in an array.

```
1  #include <stdio.h>
2
3  int main() {
4      int arr[5];
5
6      printf("Enter 5 numbers");
7
8      for(int i = 0; i < 5; ++i) {
9          scanf("%d", &arr[i]);
10     }
11 }
```

49. Why C is called middle level language?

C is called middle level language as some of its features resemble high level language and some resemble low level language. For example, certain instructions in C are like a normal algebraic expressions along with certain English keyword

like if, else, for etc. which resembles a high level language. On the other hand C permits the manipulation of individual bits which resembles low level language.

50. What is conditional operator in C?

Conditional operator also known as ternary operator is used for decision making and it uses two symbols- ? and :. The syntax for conditional operator is as follows:

Expression1? expression2: expression3;

Code for the conditional operator is as follows:

```
1 | #include <stdio.h>
2 | int main()
3 | {
4 |     int a;
5 |     printf("Enter a number");
6 |     scanf("%d",&a);
7 |     (a>=20)? (printf("Hi")) : (printf("Hey"));
8 |     return 0;
9 | }
```

51. Why we use conio.h in C?

Conio is a header file which stands for console input output. It contains some functions and methods for formatting the output and getting input in the console which is why it is used.

52. What is null in C?

Null is macro defined in C which has a value equal to zero. It is defined as follows:

```
1 | #define NULL (void *) 0;
```

C Technical Interview Questions

53. What is pointer in C?

Every variable is stored at an address in C and pointers are special variables which store addresses. You can declare pointers in 3 different ways.

```
1  int* ptr;  
2  int *ptr;  
3  int * ptr;  
4  
5  A simple example of pointer is :  
6  
7  int *ptr;  
8  int n;  
9  n=1;  
10 ptr=&n;
```

54. How to print double in C?

Double in C is initialized using the syntax below:

```
double d;
```

To print, double is printed using %lf

```
printf("%lf",d);
```

55. Which data type has more precision in C?

Long double data type has more precision (upto 19 decimal places).

C data structure Interview Questions

56. What is array in C?

Array can be defined as a collection of the same type of data which can be accessed using the same name. Array can be defined in different dimensions for example, one dimensional array, two dimensional array etc. 1D array is like a list, 2D array is like a table with rows and columns.

57. What is an array in C?

Array can be defined as a collection of the same type of data which can be accessed using the same name. Array can be defined in different dimensions for example, one dimensional array, two dimensional array etc. 1D array is like a list, 2D array is like a table with rows and columns.

58. How to pass 2d array to function in C?

Passing a 2D array to a function is similar to passing 1D array. A sample code depicting the same is as follows:

```
1  #include <stdio.h>
2
3  void print_array(int arr[2][2])
4  {
5      printf("Displaying Arrayn");
6      for (int i = 0; i < 2; ++i) {
7          for (int j = 0; j < 2; ++j) {
8              printf("%dn", arr[i][j]);
9          }
10     }
11 }
12
13 int main()
14 {
15     int arr[2][2];
16     printf("Enter the array n");
17     for (int i = 0; i < 2; ++i)
18         for (int j = 0; j < 2; ++j)
19             scanf("%d", &arr[i][j]);
20
21     print_array(arr);
22     return 0;
23 }
```

C Programming Interview Questions

59. How to reverse a string in C?

To reverse a string in C, we can make use of 2 variables one starting from the 0th index and one from the last index and making use of a temporary variable reversing the string. The code for the same is as follows:

```
1  #include<stdio.h>
2  #include<conio.h>
3  void main()
4  {
5      char str_1[10], s[20];
6      int k, j;
7      k=j=0;
8      print("Enter the string to be revered");
9      scanf("%s", str_1);
10     //making the variable k start from the end
11     while (str[k] != '\0')
12         k++;
13     k--; //removing the special character from the count
```

```
14     while (j < k) {
15         s= str_1[j];
16         str[j] = str[k];
17         str[k] = s;
18         j++;
19         k--;
20     }
21 }
```

60. How to compare two strings in C?

We can compare two strings using the in-built function `strcmp()` which returns 0 if the two strings are equal or identical, negative integer when the ASCII value of first unmatched character is less than the second, and positive integer when the ASCII value of the first unmatched character is greater than the second. Let's understand through code.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void main()
5  {
6      char first_str[] = "ab", second_str[] = "ab", third_str[] = "Ab";
7      int status;
8
9      status= strcmp(str1, str2);
10     printf("Comparing first and second string %dn", status)
11
12     status= strcmp(str1, str3);
13     printf("Comparing first and third string %dn", status);
14
15 }
```

The result of comparing the first and second string will be 0 as both the strings are identical. However, the result of comparing the first and third strings will be 32 as the ASCII code of b is 98 and B is 66.

61. How to run C program in cmd?

The steps to run your c code in command prompt is as follows:

Step-1: Install C compiler (for example gcc)

Step-2: Write the c code and save it as .c file. For example, the name of your file is family.c and it is stored in Desktop.

Step-3: Open Command Prompt clicking start button → All Apps → Windows System folder → Click Command Prompt.

Step-4: Change the directory in command prompt to where you have saved the code file. For example, cd Desktop

Step-5: Compile the code file using the compiler you have installed. Use the following syntax gcc -o <name_of_executable> <name_of_source_code> For example, gcc -o fam family.c

Step-6: Now run the executable file (for example, fam.exe) by going to the directory of the executable file in command prompt (usually the same as code file until specified otherwise), then type the name of the executable file without the extension

62. How to concatenate two strings in C?

Two strings in C can be concatenated by taking a third string and appending the two strings into third. The code for the same is as follows:

```
1  #include<stdio.h>
2  #include<conio.h>
3  void main()
4  {
5  char str_1[10], str_2[10], temp[20];
6  int k, j;
7  print("Enter first string to be concatenated");
8  scanf("%s", str_1);
9  printf("\nEnter second string to be concatenated");
10 scanf("%s", str_2);
11 for(j=0; str_1[j]!='\0'; j++)
12 {
13 str_3[j]=str_1[j];
14 }
15 str_3[j]='\0';
16 for(k=0; k<=j; k++)
17 {
18 str_3[j+k]=str_2[k];
19 }
```

```
20 | printf("The Concatenated string is %s",str_3);  
21 | }
```

63. What is data type in C?

Data types in c are used to tell the c compiler what type of data is going to be stored in a variable or what type of a function it is going to be.

Eg.

– for variables:

Let's say there is a variable 'a' and we want to store integers in it, so we will make use of the declaration type 'int' to tell the compiler that 'a'.

```
1 | int a;
```

here 'int' is the data type.

– for functions:

```
1 | int ourFunction()  
2 | {  
3 |     return 11;  
4 | }
```

Here we are returning an integer value '11' and for that purpose we have assigned the data type of this function.

64. How to print a string in C?

The syntax to print the string is as follows:

```
printf( "%s", <string name>);
```

For example: printf ("%s", dog_breed);

65. How to initialize array in C?

Array is initialized in two ways in C. First if the array elements are user input:

```
1  #include <stdio.h>
2  void main ()
3  {
4
5      int arr[ 10 ];
6      int i,j;
7
8      for ( i = 0; i < 10; i++ ) {
9          n[ i ] = i + 100;
10     }
11 }
12
13 Second if the array is programmer defined.
14 #include <stdio.h>
15 void main ()
16 {
17     int arr[] = [1,2,3,4,5]
18     for (int i=0;i<5;i++)
19         printf("%d", arr[i]);
20 }
```

66. How to declare a string in C?

The syntax of declaring a string in C is as follows: `char <string name> [size of the string];` For example: `char dog_breed [100];`

67. How to print string in C?

You can do it in two ways:

Using scanf:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char str_val[50];
6      printf("Please input a string value:");
7      scanf("%s",str_val);
```

```
8 |         return 0;  
9 |     }
```

or Using gets():

```
1 | #include <stdio.h>  
2 |  
3 | int main()  
4 | {  
5 |     char str_val[50];  
6 |     printf("Please input a string value:");  
7 |     gets(str_val);  
8 |     return 0;  
9 | }
```

68. How to reverse a number in C?

We can reverse a number using a temporary variable and performing modulus and division operations on the number. The code for reversing a number is as follows:

```
1 | #include <stdio.h>  
2 | int main()  
3 | {  
4 |     int num, rev = 0, temp;  
5 |     printf("Enter a number");  
6 |     scanf("%d", &num);  
7 |     while (num != 0) {  
8 |         temp = num % 10;  
9 |         rev = rev * 10 + temp;  
10 |        num /= 10;  
11 |    }  
12 |    printf("Reversed number = %d", rev);  
13 |    return 0;  
14 | }
```

69. How to return array in C?

The most commonly used way to return an array is by using a pointer. A sample code for returning an array using pointer is mentioned below.

```
1 | #include <stdio.h>  
2 |
```

```
3  int* dummy(int *d)
4  {
5      d[0] = 1;
6      d[1] = 2;
7
8      return d;
9  }
10
11 int main()
12 {
13     int d[2];
14     int* arr= dummy(d);
15     printf("%d %d", arr[0], arr[1]);
16     return 0;
17 }
```

70. What is dangling pointer in C?

If a programmer fails to initialize a pointer with a valid address, then the pointer is known as a dangling pointer in C. Dangling pointers point to a memory location that has been deleted.

71. How to initialize string in C?

There are multiple ways to do so:

Character array Initialization:

– Even here we have several ways:

```
1  char str_value[20] = {'H','e','l','l','o','-','W','o','r','l','d','\0'};
2  char str_value[] = {'H','e','l','l','o','-','W','o','r','l','d','\0'};
3  char str_value[20] = ("Hello-World");
4  char str_value[20] = ("Hello-World");
```

or you can define a macro and use it with the array

```
1  #define value 45
2  char str_value[value] = ("Hello-World");
```

Character pointer initialization:

```
1 | char *str_value = "Hello-World";
```

Here Hello-World value is read-only and cannot be edited.

72. What is malloc in C?

Malloc is used for dynamic memory allocation. Malloc, refers to memory allocation, which reserves a block of memory for specific bytes. It initializes each block with default garbage value and returns a pointer of type void. The syntax for malloc is as follows:

```
1 | ptr_name = (cast_type*) malloc(byte_size)
```

Example: ptr = (int*) malloc(10 * sizeof(int));

The size of int is 4 bytes so this statement will allocate 40 bytes of memory. The pointer ptr will hold the address of the first byte in the allocated memory.

73. What is typedef in C?

Typedef is a keyword that assigns alternative names to existing data types. For example, for variables with data type unsigned long or for structures.

```
1 | typedef struct <structure name>  
2 | {  
3 |     variable1;  
4 |     variable2;  
5 |     variable3;  
6 | } struct_name;  
7 |  
8 | struct_name sample1;
```

74. How to find array length in C?

```
1 | int int_value[] = {22,11,33};
```

```
2  int size;  
3  size = sizeof(int_value)/sizeof(int_value);  
4  printf("%d",size);
```

75. How to declare array in C?

You can do this in multiple ways, such as:

```
1  int int_value[100];  
2  
3      or  
4  
5  int no = 100;  
6  int int_value[no];  
7  
8      or  
9  
10 int int_value[3]={11,22,33};  
11  
12      or  
13  
14 int int_value[]={11,22,33};
```

76. How to compile C program in cmd?

Open cmd

verify gcc installation using the command:

```
$ gcc -v
```

then go to your working directory or folder where your code is:

```
$ cd <folder_name>
```

then build the file containing your c code as such:

```
$ gcc main.c
```

then run the executable generated in your system:

```
$ main.exe
```

77. How to print a sentence in C?

```
1  int main() {  
2  int str_value[20];  
3  printf("Enter a string value:n");
```

```
4  scanf("%s",str_value);
5  printf("The string you entered is:%sn",str_value);
6  }
7
8  or
9
10 int main(){
11     int str_value[20];
12     printf("Enter a string value:n");
13     gets(str_value);
14     printf("The string you entered is:%sn",str_value);
15 }
```

78. How to print ascii value in C?

The code to print ASCII value is as follows:

```
1  #include <stdio.h>
2  void main() {
3      char a;
4      printf("Enter a character");
5      scanf("%c", &a);
6      printf("ASCII value of %c = %d", a, a);
7  }
```

79. What are storage classes in C?

A storage class represents information about variable such as variable scope, location, value, lifetime etc. There are four types of Storage classes.

1. Auto (default storage class)
2. Extern (global variable)
3. Static
4. Register

80. What are macros in C?

Defined by #define directive, a macro is defined as a segment of code which can be replaced by its initialized value. There are two types of macros.

1. Object-like: For example: #define M=1000
2. Function like: For example: #define Max(x,y) ((x)>(y)?(x):(y))

81. How to use stdin and stdout in C?

STDIN stands for Standard Input.

STDOUT stands for Standard Output.

There are several functions to both input and output.

scanf() [STDIN] and printf() [STDOUT]

getchar() [STDIN] and putchar() [STDOUT]

gets() [STDIN] and puts() [STDOUT]

```
1  #include <stdio.h>
2  int main( ) {
3
4      int a;
5      printf( "Enter a value");
6      a = getchar( );
7      printf( "You entered");
8      putchar( a );
9
10     char s[100];
11     printf( "Enter a value");
12     gets( s );
13     printf( "You entered");
14     puts( s );
15
16     printf( "Enter a value :");
17     scanf("%s %d", s, &a);
18     printf( "You entered %s %d ", s, a);
19
20     return 0;
21 }
```

82. What is garbage value in C?

In C, when a variable is defined but no value is initialized or assigned to it. Then that variable is assigned to any random computer's memory which is called Garbage value.

83. What is type casting in C?

Typecasting is a process to convert one data type to another. The syntax for type casting is as follows: (type_name) expression A sample code for the same is as follows:

```
1  #include <stdio.h>
2
3  void main() {
4
5      int a = 6, b= 5;
6      double c;
7
8      c= (double) a/ b;
9      printf("%f", c);
10 }
```

84. How to print pattern in C?

Suppose you have a half pattern mentioned below to print.

```
*
* *
* * *
```

The code for the same is:

```
1  #include <stdio.h>
2  int main() {
3      int i, j;
4      for (i = 1; i <= 3; ++i) {
5          for (j = 1; j <= i; ++j) {
6              printf("* ");
7          }
8          printf("\n");
9      }
10     return 0;
11 }
```

The * can be replaced by alphabet or number.

For full pattern, the code is:

```
*
* * *
* * * * *
```

```
1  #include <stdio.h>
2  int main() {
3      int i, s, j = 0;
4      for (i = 1; i <= 3; ++i, j = 0) {
5          for (s = 1; s <= 3 - i; ++s) {
6              printf(" ");
7          }
8          while (j != 2 * i - 1) {
9              printf("* ");
10             ++j;
11         }
12         printf("\n");
13     }
14     return 0;
15 }
```

85. How to convert uppercase to lowercase in C?

To convert a string from uppercase to lowercase, a predefined function `tolower()` can be used and also by looping it can be achieved. A sample code containing both are as follows:

```
1  #include<stdio.h>
2  #include<string.h>
3  int main()
4  {
5      char s[25];
6      int i;
7
8      //through looping
9      printf("Enter the string");
10     scanf("%s", s);
11
12     for(i=0; i<=strlen(s); i++) {
13         if(s[i]>=65 && s[i]<=90)
14             s[i]=s[i]+32;
15     }
```

```
16     printf("%s", s);
17
18     //using in-built function- toupper()
19     printf("Enter the string");
20     scanf("%s", s);
21     printf("%s", tolower(s));
22     return 0;
23 }
```

86. What is flowchart in C?

Graphical representation of an algorithm which as a tool for program planning for solving a problem is used is called flowchart. Symbols are used in flowchart for representing flow of information.

87. How to read a file in C?

The code to read a file in C as follows:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      char c, filename[25];
6      FILE *p;
7
8      printf("Enter name of a file");
9      gets(filename);
10
11     p = fopen(filename, "r"); // read mode
12
13     if (p == NULL)
14     {
15         perror("Error while opening the file.");
16         exit(EXIT_FAILURE);
17     }
18
19     while((c = fgetc(p)) != EOF) //printing the content of file
20         printf("%c", c);
21
22     fclose(p);
23     return 0;
24 }
```

88. What is perfect number in C?

Perfect number is a number which is equal to the sum of its divisor. For example, 6 is a perfect number because 6 divisors: 1,2,3 has sum also equal to 6.

Code to check whether a number is perfect or not is as follows:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, r, s = 0, j;
6
7      printf("Enter a Number");
8      scanf("%d", &n);
9      //for loop for identifying the perfect number
10     for (j = 1; j <= (n - 1); j++)
11     {
12         r = n% j;
13         if (r == 0)
14         {
15             s= s+ j;
16         }
17     }
18     if (s == n)
19         printf("%d is perfect number",n);
20     else
21         printf("%d is not a perfect number",n);
22     return 0;
23 }
```

89. How to convert lowercase to uppercase in C?

Using toupper() functions, one can convert lowercase to uppercase in c. Syntax for the same is as follows. int toupper(int ch);

90. How to stop infinite loop in C?

Break statements (along with a condition) can be used in C to stop infinite loop. Also, keyboard shortcuts such as CTRL-Break, Break, CTRL-C and CTRL-ESC-ESC can be used based on the compiler you are using. Sample code for break statement is as follows:

```
1  #include <stdio.h>
2
3  int main() {
4      int a = 1;
5
6      // infinite while loop
7      while (1) {
8          if (a > 10) // in case this if is not used along with
9              break;
10
11         printf("%d ", a);
12         a++;
13     }
14
15     return 0;
16 }
```

91. How to swap two numbers in C?

Swapping two numbers using a temporary variable.

```
1  #include<stdio.h>
2  int main() {
3      int a, b, temp;
4      scanf("%d", &a);
5      scanf("%d", &b);
6
7      temp = a;
8      a= b;
9      b= temp;
10
11     printf("%.d", a);
12     printf("%.d", b);
13     return 0;
14 }
```

92. What is %d in C?

%d means printing and scanning integer variables.

93. What is control statement in C?

A statement that determines whether other statements will be executed or not is called a control statement. There are two types of flow control in C.

Branching: it decides what action to take. For example, if and else statement.

Looping: it decides a number of times to take certain actions. For example, for, while and do while.

94. What is while loop in C?

A while loop allows the code written inside the code block to be executed a certain number of times depending upon a given Boolean condition.

The syntax of while loop in c language:

```
while(condition)
{
//code to be executed
}
```

95. What is static function in C?

Static in C is a keyword applied before a function to make it static. Syntax for static function is as follows:

```
static int fun(void)
{
printf("static function ");
}
```

96. What is pointer to pointer in C?

Pointers are used to store the address of variable. In Pointer to pointer, the first pointer stores the address of the variable and the second pointer is used to st

the address of the first pointer. Syntax for declaring pointer to pointer is as follows:

```
int **ptr;
```

97. What is stderr in C?

There are 3 I/O descriptors: stdin for standard input, stdout for standard output, stderr for error message output. stderr is used for mapping the terminal output and generates the error messages for the output devices. Sample code for the same is as follows:

```
1  #include <stdio.h>
2  int main()
3  {
4      fprintf(stderr, "Error");
5  }
```

98. How to declare Boolean in C?

Boolean is a data type containing two values 1 or 0.

Syntax for declaring Boolean in C `bool variable_name;`

Code for illustrating the same is as follows:

```
1  #include <stdio.h>
2  #include<stdbool.h>
3  int main()
4  {
5      bool x=false;
6      if(x==true)
7          printf("value of x is true");
8      else
9          printf("value of x is false");
10     return 0;
11 }
```

99. How to find the length of a number in C?

The code to calculate length of a number in C is as follows:

```
1  #include <stdio.h>
2
3  int num_length(long num)
4  {
5      int count = 0;
6      while (num != 0) {
7          num = num / 10;
8          ++count;
9      }
10     return count;
11 }
12
13 int main(void)
14 {
15     long long n = 7654321;
16     printf("%d", num_length(n));
17     return 0;
18 }
```

100. Write a program to reverse an integer in C.

```
#include <stdio.h>
int main() {
    int n, rev = 0, rem;
    printf("Enter an integer = ");
    scanf("%d", &n);
    while (n != 0)
    {
        rem = n % 10;
        rev = rev * 10 + rem;
        n /= 10;
    }
    printf("Reversed number = %d", rev);
    return 0;
}
```

101. Write a program in C to check whether an integer is an Armstrong number or not.

```
#include <stdio.h>
void main()
{
```

```

int num,x,sum=0,arm;
printf("Enter a number = ");
scanf("%d",&num);
for (arm=num;num!=0;num=num/10)
{
    x=num % 10;
    sum=sum+(x*x*x);
}
if(sum==arm)
    printf("%d Given is an Armstrong number.n",arm);
else
    printf("%d Given is not an Armstrong number.n",arm);
}

```

102. Write a program in C to check if a given number is prime or not.

```

#include<stdio.h>
int main()
{
    int m,i,n=0,rem=0;
    printf("Enter the number:-  ");
    scanf("%d",&m);
    n=m/2;
    for(i=2;i<=n;i++)
    {
        if(m%i==0)
        {
            printf("Number is not a prime number ");
            rem=1;
            break;
        }
    }
    if(rem==0)
        printf("Number is a prime number ");
    return 0;
}

```

104. Write a program in C to print the Fibonacci series using iteration.

```

#include <stdio.h>
int main()
{
    int x, i, m1 = 1, m2 = 2, m;
    printf("Enter the number of terms to generate the fibonacci series = n");
    scanf("%d", &x);
    printf("Fibonacci Series is = ");
    for (i = 1; i <= x; ++i)

```



```
{
    printf("%d, ", m1);
    m = m1 + m2;
    m1 = m2;
    m2 = m;
}
printf("n");
return 0;
}
```

105. Write a program in C to print the Fibonacci series using recursion.

```
#include<stdio.h>
int main()
{
    int x=0, y=1, i, n, sum=0;
    printf("Enter the number of terms= ");
    scanf("%d",&n);
    printf("Fibonacci Series is: ");
    for(i=0 ; i<n ; i++)
    {
        if(i <= 1)
        {
            sum=i;
        }
        else
        {
            sum=x + y;
            x=y;
            y=sum;
        }
        printf(" %d",sum)
    }
    return 0;
}
```

106. Write a program in C to check whether a number is a palindrome or not using iteration.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, reverse = 0, temp;
    printf("Enter the number = n");
    scanf("%d",&n);
    temp = n;
```

```
        while(temp!=0){
            reverse = reverse*10 + temp%10;
            temp=temp/10;
        }
        if(reverse == n)
        {
            printf("Given number is a palindrome.");
        }
        else{
            printf("Given number is not a palindrom.");
        }
        getch();
    }
```

107. Write a program in C to check whether a number is a palindrome or not using recursion.

```
#include <stdio.h>
#include <conio.h>
int reverse(int num);
int isPalindrome(int num);

int main()
{
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if(isPalindrome(num) == 1)
    {
        printf("Given number is a palindrome");
    }
    else
    {
        printf("Given number is not a palindrome number");
    }
    return 0;
}

int isPalindrome(int num)
{
    if(num == reverse(num))
    {
        return 1;
    }
    return 0;
}

int reverse(int num)
{
    int rem;
    static int sum=0;
```

```

    if (num!=0)
    {
        rem=num%10;
        sum=sum*10+rem;
        reverse (num/10);
    }
    else
        return sum;
return sum;
}

```

108. Write a program in C to find the greatest among three integers.

```

#include <stdio.h>
int main()
{
    int X, Y, Z;
    printf("Enter the numbers X, Y and Z = ");
    scanf("%d %d %d", &X, &Y, &Z);
    if (X >= Y && X >= Z)
        printf("%d is the largest number.", X);
    if (Y >= X && Y >= Z)
        printf("%d is the largest number.", Y);
    if (Z >= X && Z >= Y)
        printf("%d is the largest number.", Z);
    return 0;
}

```

109. Write a program in C to check if a number is binary.

```

#include<stdio.h>
int main()
{
    int j,num;
    printf("Enter a number= ");
    scanf("%d",&num);
    while(num>0)
    {
        j=num%10;
        if( j!=0 && j!=1 )
        {
            printf("Number is not binary");
            break;
        }
        num=num/10;
        if (num==0)
        {
            printf("Number is binary");
        }
    }
}

```

```

    }
}
return 0;
}

```

110. Write a program in C to find the sum of digits of a number using recursion.

```

#include<stdio.h>

int Sum(int);
int main()
{
    int x,sum;
    printf("Enter a Number = ");
    scanf("%d",&x);

    sum = Sum(x);

    printf("The sum of Digits of Given Number is = %d",sum);
    return 0;
}

int Sum(int x)
{
    static int sum =0,r;

    if(x!=0)
    {
        r=x%10;
        sum=sum+r;
        Sum(x/10);
    }

    return sum;
}

```

FAQs

Q1 What are the basics of C?

Below is the syntax and a few commands to write a basic c program.

Code:

```
#include <stdio.h>
int main()
{
    /*C basic program */
    printf("Hello!");
    getch();
    return 0;
}
```

Output:

Hello!

Command	Explanation
#include<stdio.h>	Preprocessor command that includes standard input output header file (stdio.h) from the C library before the C program is compiled.
int main()	Main function from where the execution of C programs begin.
{	Beginning of the main function is indicated.
/*Comment*/	Text within "/* */" is not considered for compilation or execution.
printf("Great Learning");	To print the output on the screen, print command is used.
getch();	Command that waits for any character input from the user.

<code>return 0;</code>	It terminates the main function and returns 0.
<code>}</code>	Marks the end of the main function.

Q2 What is C mainly used for?

C is a general-purpose programming language that can be easily read, edited, and understood. It is the language in which operating systems for desktop as well as mobile phones are developed. Flexibility is a property of C that makes it popular for embedded systems programming. Even if you use C language for the embedded systems, you don't have to pay anything for it as C is a free language.

Q3 Why is C dangerous?

C is unsafe as executing an erroneous operation can cause the entire program to become meaningless. In such a case, the output would become unpredictable as erroneous operations are said to have undefined behavior. Also, C programming allows arbitrary pointer arithmetic with pointers implemented as direct memory addresses having no provision for the bound checking, making C potentially memory-unsafe.

Q4 What is the main() in C?

Every program in C must have a `main()` function as it is the entry point for any C program. The first function in your program to be executed at the beginning of execution is the `main()` function, as the execution control goes directly to it.

Q5 What are keywords in C?

Keywords have specific meanings associated with them to the compiler. They have been already defined (pre-defined) and reserved. They cannot be used as identifiers. The 32 keywords in C are:

1. auto
2. break
3. case
4. char
5. const
6. continue
7. default
8. do
9. double
10. else
11. enum
12. extern
13. float
14. for
15. goto
16. if
17. int
18. long

19. register
20. return
21. short
22. signed
23. sizeof
24. static
25. struct
26. switch
27. typedef
28. union
29. unsigned
30. void
31. volatile
32. while

Q6 What is c and its features?

C is a widely used, easily read, edited, and understood programming language. Features of C programming language are:

1. Simple
2. Procedural Language
3. Fast and Efficient
4. Modularity

5. Statically Type
6. General-Purpose Language
7. Rich Libraries
8. Middle-Level Language
9. Machine Independent or Portable
10. Easily Extensible

Q7 Is C still used in 2021?

C is a middle-level language; hence it combines features of both the high-level and the low-level languages. It can be used for scripting for drivers & kernels (for low-level programming) & also for the scripting for software applications (for high-level language). C is a language very close to the hardware and is very useful for coding in situations where the speed of execution is critical or only limited resources are available.

Q8 Why is C so popular?

C is known as the mother of all programming languages. It is still widely used and popular due to its features and flexible memory management. For system-level programming, C is considered the best option. Operating system kernels are written in C, so even the smartphones that you have been using every day are running on a C kernel.

Q9 Which is better Python or C?

Both C & Python are useful languages to develop different applications. C is mainly used for the development related to hardware like network drivers & operating systems, whereas Python is used for natural language processing, machine learning, web development, etc. It is not enough for one to master a single programming language in today's competitive scenario; therefore, learning both will be beneficial to stand strong in the present competitive market.

This brings us to end of the blog on C Interview Questions. We hope you are now well-equipped with the kind of questions that may be asked during an Interview. Wondering where to learn the highly coveted in demand skills for free? Check out the courses on [Great Learning Academy](#).

Sharing is caring:



Tanuja Bahirat

Tanuja is a content writer who enjoys spending time in nature, watching football, and journaling. She loves attending music festivals and reading. In her current journey, she writes about recent advancements in technology and its impact on the world.

Recommended for you

Top Skills required to become a DevOps Engineer

20 Best Programming Languages to Learn in 2025

Data Structures using C | What are the Data Structure in C and How it works?

The Latest Technology Trends of 2024

Top Python Books You Need to Read

Top DevOps Interview Questions and Answers

Free Courses

[Free Artificial Intelligence Course With Certificate](#)

[Free Prompt Engineering Course With Certificate](#)

[Python for Machine Learning Free Course](#)

[Data Science Foundations Free Course](#)

[Deep Learning with Python Free Course](#)

[Introduction to Cyber Security Free Course](#)

[Free Digital Marketing Course](#)

[Java Programming Free Course](#)

[View More →](#)

Study Abroad

[Study Abroad Programs](#)

[Study in USA](#)

[Study in Germany](#)

Popular Courses

[PGP In Data Science and Business Analytics](#)

[PGP In Artificial Intelligence And Machine Learning](#)

[PGP In Management](#)

[PGP In Cloud Computing](#)

[Software Engineering Course](#)

[PGP In Digital Marketing](#)

[Data Science Course Eligibility and Requirements](#)

[View More →](#)

Blog Categories

[Data Science Articles](#)

[Artificial Intelligence Articles](#)

[Cloud Computing Articles](#)

[Cyber Security Articles](#)

[IT/Software Guides](#)

[Career Development Articles](#)

[Research and Studies](#)

Premium Courses

[Master Generative AI](#)

[Master Python programming](#)

[Master Data Analytics in SQL](#)

[Master Data Analytics in Excel](#)

[Master Data Analytics in SQL & Excel](#)

[Master Data Science & Machine Learning in Python](#)

International Courses

[AI in Healthcare Program](#)

[Certificate Program in Applied Generative AI](#)

[Certificate Program in AI Business Strategy](#)

[MIT Applied Data Science Program](#)

[MIT Data Science and Machine Learning Course](#)

[Full Stack Developer Program](#)

[Artificial Intelligence Program for Leaders](#)

[Post Graduate Program in Data Science and Business Analytics](#)



[About Us](#)

[Contact Us](#)

[Privacy Policy](#)

[Terms of Use](#)

[Great Learning Careers](#)

© 2013 - 2025 Great Learning Education Services Private Limited (Formerly known as Great Lakes E-Learning Services Private Limited).All rights reserved

Get our android app

Get our ios app