Interview Questions in C Programming

1. What is C programming?

C is a general-purpose, procedural programming language developed by Dennis Ritchie at Bell Laboratories in the early 1970s. It is designed to provide low-level access to memory while still being easy to write and understand. C is widely used for developing operating systems, compilers, device drivers, embedded systems, and system software. Because of its efficiency, portability, and direct interaction with hardware, C is often called the mother of many modern programming languages.

2. What are the main features of C?

The important features of C programming include:

Simple and efficient: Uses a small set of keywords and executes programs quickly.

Structured language: Programs are divided into functions, making code easy to understand and maintain.

Portable: A C program written on one system can be run on another with little or no modification.

Pointer support: Enables direct memory access, which is useful in system-level programming.

Rich library functions: Provides many built-in functions for file handling, string operations, and memory management.

Fast execution: C programs are compiled, resulting in high performance.

3. What is the difference between printf() and scanf()?

printf() and scanf() are standard input/output functions in C.

printf() is used to display output on the screen. It sends formatted data to the output device (monitor).

scanf() is used to read input from the user. It stores the entered data into variables using their addresses.
In short, printf() is an output function, while scanf() is an input function.

4. What is a pointer?

A pointer is a variable that stores the memory address of another variable. Instead of holding a value directly, it points to the location where the value is stored. Pointers are important in C because they allow dynamic memory allocation, passing arguments by reference, working with arrays and strings, and efficient memory management. Pointers make C powerful but require careful handling to avoid errors.

5. What is the difference between call by value and call by reference?

Call by value: A copy of the actual variable is passed to the function. Any changes made inside the function do not affect the original variable.

Call by reference: The address of the actual variable is passed to the function using pointers. Changes made inside the function directly affect the original variable.
Call by value is safer, while call by reference is more efficient when large data needs to be modified.

6. What is an array?

An array is a data structure that stores a fixed number of elements of the same data type in contiguous memory locations. Each element can be accessed using an index starting from zero. Arrays are useful for storing and processing large amounts of data efficiently, such as lists of numbers, marks of students, or characters in a string.

7. What is the use of the sizeof() operator?

The sizeof() operator is used to determine the size of a data type or variable in bytes. It is evaluated at compile time and helps in understanding memory requirements. It is commonly used in dynamic memory allocation, array size calculations, and ensuring portability across different systems.

8. What is the difference between malloc() and calloc()?

Both malloc() and calloc() are used for dynamic memory allocation.

malloc() allocates a single block of memory of the specified size but does not initialize it.

calloc() allocates memory for multiple elements and initializes all allocated memory to zero.
calloc() is slower than malloc() because it initializes memory, but it is safer to use.

9. What is recursion?

Recursion is a programming technique in which a function calls itself to solve a problem. A recursive function must have a base condition to stop execution; otherwise, it leads to infinite recursion. Recursion is commonly used in problems that can be divided into smaller subproblems, such as factorial calculation, Fibonacci series, and tree traversal.

10. What is the difference between break and continue?

break is used to terminate a loop or switch statement completely and transfer control to the statement following the loop.

continue is used to skip the remaining statements of the current iteration and move to the next iteration of the loop.
Both are control statements used to alter the normal flow of loops.