C LANGUAGE INTERVIEW QUESTIONS

1. Why is C called a mid-level programming language?

Ans. Due to its ability to support both low-level and high-level features, C is considered a middle-level language. It is both an assembly-level language, i.e. a low-level language, and a higher-level language. Programs that are written in C are converted into assembly code, and they support pointer arithmetic (low-level) while being machine-independent (high-level). Therefore, C is often referred to as a middle-level language. C can be used to write operating systems and menu-driven consumer billing systems.

2. What are basic data types supported in the C Programming Language?

Ans. Each variable in C has an associated data type. Each data type requires different amounts of memory and has some specific operations which can be performed over it. It specifies the type of data that the variable can store like integer, character, floating, double, etc. In C data types are broadly classified into 4 categories:

- **Primitive data types:** Primitive data types can be further classified into integer, and floating data types.
- **Void Types:** Void data types come under primitive data types. Void data types provide no result to their caller and have no value associated with them.
- **User Defined data types:** These data types are defined by the user to make the program more readable.
- **Derived data types:** Data types that are derived from primitive or built-in data types.

3. What are tokens in C?

Ans. Tokens are identifiers or the smallest single unit in a program that is meaningful to the compiler. In C we have the following tokens:

Keywords: Predefined or reserved words in the C programming language. Every keyword is meant to perform a specific task in a program. C Programming language supports 32 keywords.

Identifiers: Identifiers are user-defined names that consist of an arbitrarily long sequence of digits or letters with either a letter or the underscore (_) as a first Character. Identifier names can’t be equal to any reserved keywords in the C programming language. There are a set of rules which a programmer must follow in order to name an identifier in C.

Constants: Constants are normal variables that cannot be modified in the program once they are defined. Constants refer to a fixed value. They are also referred to as literals.

Strings: Strings in C are an array of characters that end with a null character (‘\0). Null character indicates the end of the string;

Operators: Symbols that trigger an action when they are applied to any variable or any other object. Unary, Binary, and ternary operators are used in the C Programming language.

4. What do you mean by the scope of the variable?

Ans. Scope in a programming language is the block or a region where a defined variable will have its existence and beyond that region, the variable is automatically destroyed. Every variable has its defined scope. In simple terms, the scope of a variable is equal to its life in the program. The variable can be declared in three places These are:

Local Variables: Inside a given function or a block

Global Variables: Out of all functions globally inside the program.

Formal Parameters: In-function parameters only.

5. What is the use of static variables in C?

Ans. Static variables in the C programming language are used to preserve the data values between function calls even after they are out of their scope. Static variables preserve their values in their scope and they can be used again in the program without initializing again. Static variables have an initial value assigned to 0 without initialization.

6. What is recursion in C?

Ans. Recursion is the process of making the function call itself directly or indirectly. A recursive function solves a particular problem by calling a copy of itself and solving smaller subproblems that sum up the original problems. Recursion helps to reduce the length of code and make it more understandable. The recursive function uses a LIFO (Last In First Out) structure like a stack. Every recursive call in the program requires extra space in the stack memory.

7. What is the difference between the local and global variables in C?

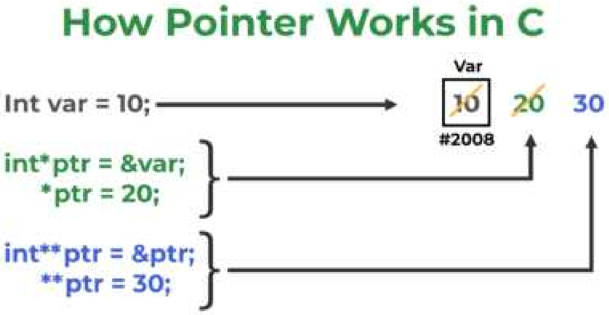
Ans. Local variables are declared inside a block or function but global variables are declared outside the block or function to be accessed globally.

Local Variables	Global Variables
Declared inside a block or a function.	Variables that are declared outside the block or a function.
By default, variables store a garbage value.	By default value of the global value is zero.
The life of the local variables is destroyed after the block or a function.	The life of the global variable exists until the program is executed.
Variables are stored inside the stack unless they are specified by the programmer.	The storage location of the global variable is decided by the compiler.
To access the local variables in other functions parameter passing is required.	No parameter passing is required. They are globally visible throughout the program.

8. What are pointers and their uses?

Ans. Pointers are used to store the address of the variable or a memory location. Pointer can also be used to refer to another pointer function. The main purpose of the pointer is to save memory space and increase execution time. Uses of pointers are:

- To pass arguments by reference
- For accessing array elements
- To return multiple values
- Dynamic memory allocation
- To implement data structures
- To do system-level programming where memory addresses are useful



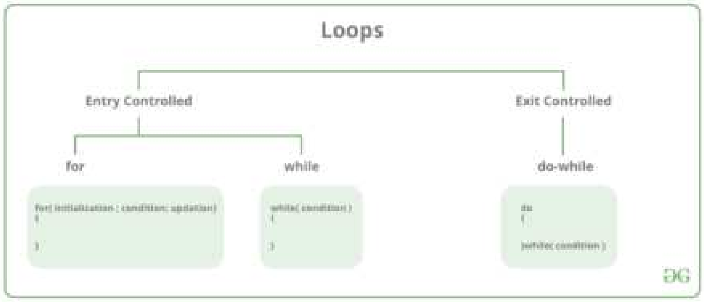
```
Int var = 10;
int*ptr = &var;
  *ptr = 20;
int**ptr = &ptr;
  **ptr = 30;
```

9. What is typedef in C?

Ans. Whenever a variable is defined some amount of memory is created in the heap. If the programmer forgets to delete the memory. This undeleted memory in the heap is called a memory leak. The Performance of the program is reduced since the amount of available memory was reduced. To avoid memory leaks, memory allocated on the heap should always be cleared when it is no longer needed.

10. What are loops and how can we create an infinite loop in C?

Ans. Loops are used to execute a block of statements repeatedly. The statement which is to be repeated will be executed n times inside the loop until the given condition is reached. There are two types of loops Entry controlled and Exit-controlled loops in the C programming language. An Infinite loop is a piece of code that lacks a functional exit. So, it repeats indefinitely. There can be only two things when there is an infinite loop in the program. One it was designed to loop endlessly until the condition is met within the loop. Another can be wrong or unsatisfied break conditions in the program.



```
for( initialization ; condition; updation)
{
}
```

```
while( condition )
{
}
```

```
do
{
}while( condition )
```

11. What are header files and their uses?

Ans. C language has numerous libraries which contain predefined functions to make programming easier. Header files contain predefined standard library functions. All header files must have a '.h' extension. Header files contain function definitions, data type definitions, and macros which can be imported with the help of the preprocessor directive '#include'. Preprocessor directives instruct the compiler that these files are needed to be processed before the compilation.
There are two types of header files i.e, User-defined header files and Pre-existing header files. For example, if our code needs to take input from the user and print desired output to the screen then 'stdio.h' header file must be included in the program as #include<stdio.h>. This header file contains functions like scanf() and printf() which are used to take input from the user and print the content.

12. What are reserved keywords?

Ans. Every keyword is meant to perform a specific task in a program. Their meaning is already defined and cannot be used for purposes other than what they are originally intended for. C Programming language supports 32 keywords. Some examples of reserved keywords are auto, else, if, long, int, switch, typedef, etc.

13. What is a structure?

Ans. The structure is a keyword that is used to create user-defined data types. The structure allows storing multiple types of data in a single unit. The structure members can only be accessed through the structure variable.

Example:

```
struct student
{
  char name[20];
  int roll_no;
 char address[20];
 char branch[20];
};
```

14. What is union?

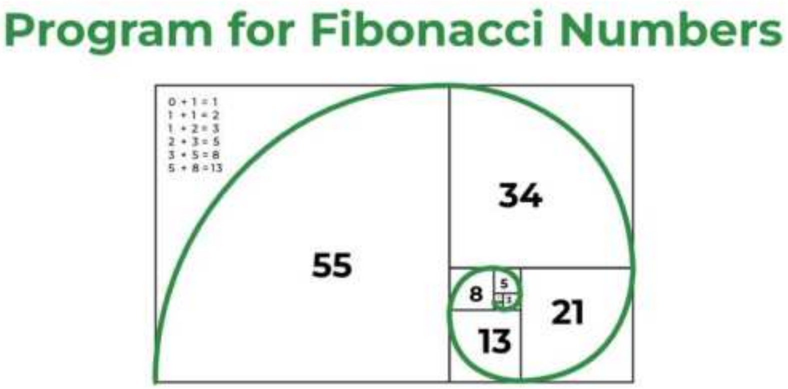
Ans. A union is a user-defined data type that allows users to store multiple types of data in a single unit. However, a union does not occupy the sum of the memory of all members. It holds the memory of the largest member only. Since the union allocates one common space for all the members we can access only a single variable at a time. The union can be useful in many situations where we want to use the same memory for two or more members.

15. What are enumerations?

Ans. In C, enumerations (or enums) are user-defined data types. Enumerations allow integral constants to be named, which makes a program easier to read and maintain. For example, the days of the week can be defined as an enumeration and can be used anywhere in the program.

Syntax: enum enumeration_name{constant1, constant2, ... };

16. Write a C program to print the Fibonacci series using recursion and without using recursion.



Note: Consider our program of Fibonacci.

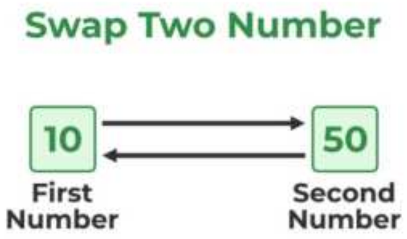
17. What is static memory allocation and dynamic memory allocation?

- **Static memory allocation:** Memory allocation which is done at compile time is known as static memory allocation. Static memory allocation saves running time. It is faster than dynamic memory allocation as memory allocation is done from the stack. This memory allocation method is less efficient as compared to dynamic memory allocation. It is mostly preferred in the array.
- **Dynamic memory allocation:** Memory allocation done at execution or run time is known as dynamic memory allocation. Dynamic memory allocation is slower than static memory allocation as memory allocation is done from the heap. This memory allocation method is more efficient as compared to static memory allocation. It is mostly preferred in the linked list.

18. What is an auto keyword?

Ans. Every local variable of a function is known as an automatic variable in the C language. Auto is the default storage class for all the variables which are declared inside a function or a block. Auto variables can only be accessed within the block/function they have been declared. We can use them outside their scope with the help of pointers. By default auto keywords consist of a garbage value.

19. Write a C program to swap two numbers without using a third variable.



Note: Consider our program of swapping.

20. Write a program to check whether a string is a palindrome or not.

Note: Consider our program of palindrome or not.

21. Write a program to print the factorial of a given number with the help of recursion.

Note: Consider our program of recursion.

22. What is the use of an extern storage specifier?

Ans. The extern keyword is used to extend the visibility of the C variables and functions in the C language. Extern is the short name for external. It is used when a particular file needs to access a variable from any other file. Extern keyword increases the redundancy and variables with extern keyword are only declared not defined. By default functions are visible throughout the program, so there is no need to declare or define extern functions.

23. What is the use of printf() and scanf() functions in C Programming language? Also, explain format specifiers.

Ans. printf() function is used to print the value which is passed as the parameter to it on the console screen.

Syntax:

print(“%X”,variable_of_X_type);

scanf() method, reads the values from the console as per the data type specified.

Syntax:

scanf(“%X”,&variable_of_X_type);

In C format specifiers are used to tell the compiler what type of data will be present in the variable during input using scanf() or output using print().

- %c: Character format specifier used to display and scan character.
- %d, %i: Signed Integer format specifier used to print or scan an integer value.
- %f: Floating-point format specifiers are used for printing or scanning float values.
- %s: This format specifier is used for String printing.

24. What is the difference between getc(), fgetc(), getch() ?

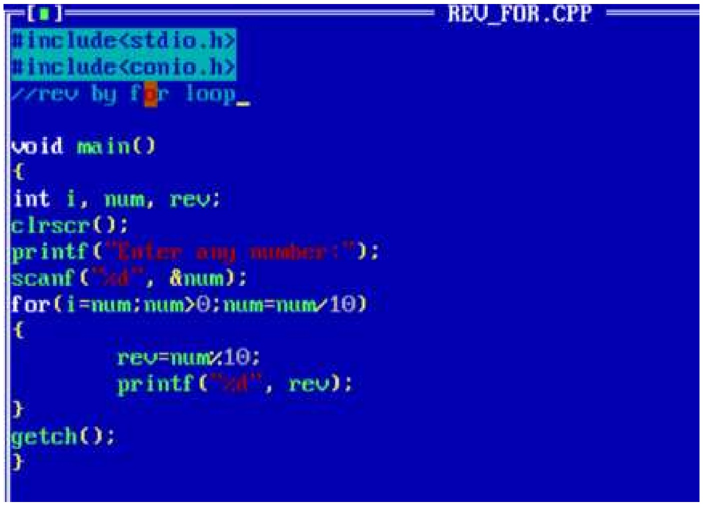
Ans. getc(): The function reads a single character from an input stream and returns an integer value (typically the ASCII value of the character) if it succeeds. On failure, it returns the EOF.

The fgetc() function reads the character at the current file position in the specified file, and increments the file position.

getch(): It is a nonstandard function and is present in ‘conio.h’ header file which is mostly used by MS-DOS compilers like Turbo C.

25. Write a program to reverse a given number.

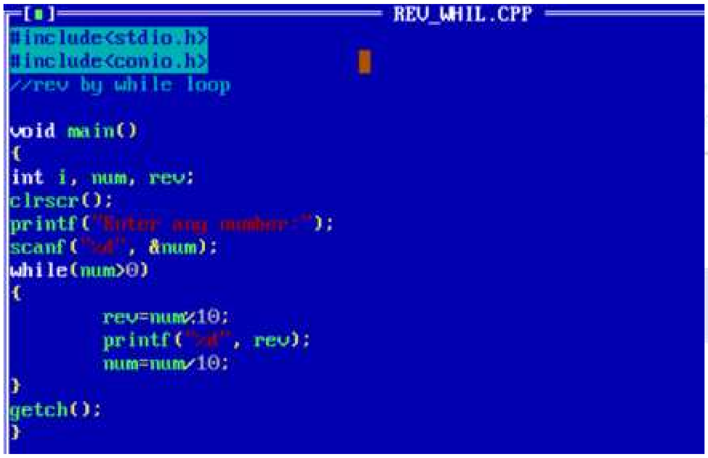
- **By For Loop**



```
#include<stdio.h>
#include<conio.h>
//rev by for loop

void main()
{
int i, num, rev;
clrscr();
printf("Enter any number:");
scanf("%d", &num);
for(i=num;num>0;num=num/10)
{
        rev=num%10;
        printf("%d", rev);
}
getch();
}
```

- **By While Loop**



```
#include<stdio.h>
#include<conio.h>
//rev by while loop

void main()
{
int i, num, rev;
clrscr();
printf("Enter any number:");
scanf("%d", &num);
while(num>0)
{
        rev=num%10;
        printf("%d", rev);
        num=num/10;
}
getch();
}
```