CSB Lab-1

Q1) Write features of C.

C is a procedural programming language, and it is considered both high-level and low-level (middle-level). Some of the key features of C include:

- Simple: C provides a structured approach, easy syntax, and rich libraries.
- Efficient: The programs written in C are highly efficient.
- Portability: C programs can run on different machines with little to no modification.
- Rich Library Support: C offers a wide variety of in-built functions.
- Memory Management: C provides dynamic memory allocation features.
- Modularity: Code can be divided into functions.
- Speed: C is faster because it has very low-level programming features.
- Pointer Support: C allows manipulation of hardware, which is useful in low-level programming.
- Structured Language: It follows a structured approach for writing programs.

Q2) Different type of data types.

In C, data types specify the type of data that a variable can store. There are various types of data types in C:

Primary Data Types:

- int: Integer type to store whole numbers.
- char: Character type to store single characters.
- float: Single precision floating-point.
- double: Double precision floating-point.
- void: Represents the absence of any type.

Derived Data Types:

- Array: A collection of data items of the same type.
- Pointer: Stores memory addresses of another variable.
- Function: A block of code that performs a task.

User-Defined Data Types:

- Structure: A collection of different types of data.
- Union: Similar to structure, but only one member can contain a value at a time.

- Enum: Represents a set of named integer constants.

Qualifiers:

- signed/unsigned: For specifying whether a variable can hold both positive and negative values.
- long/short: Modifies the size of an integer.

Q3) Familiarization of Linux environment – How to do Programming in C.

To begin programming in C on a Linux system, first install a Linux distribution (like Ubuntu or Fedora) and access the terminal. Update your package manager and install the GNU Compiler Collection (GCC) using commands like sudo apt install build-essential for Debian-based systems.

Create a new directory for your projects, then use a text editor (like nano or vim) to write a simple C program. For example, create a file named hello.c and include the basic structure to print "Hello, World!" Compile the program using gcc -o hello hello.c, and run it with ./hello.

For debugging, compile with debug information using gcc -g and utilize GDB by running gdb ./hello. Familiarize yourself with basic Linux commands such as ls, cd, cp, and rm for file operations, and consider using Makefiles for larger projects to simplify the build process.

Explore online tutorials and books like "The C Programming Language" by Kernighan and Ritchie for further learning. With these steps, you'll be well-equipped to develop and debug C programs in a Linux environment.