PROGRAMMING FOR PROBLEM SOLVING

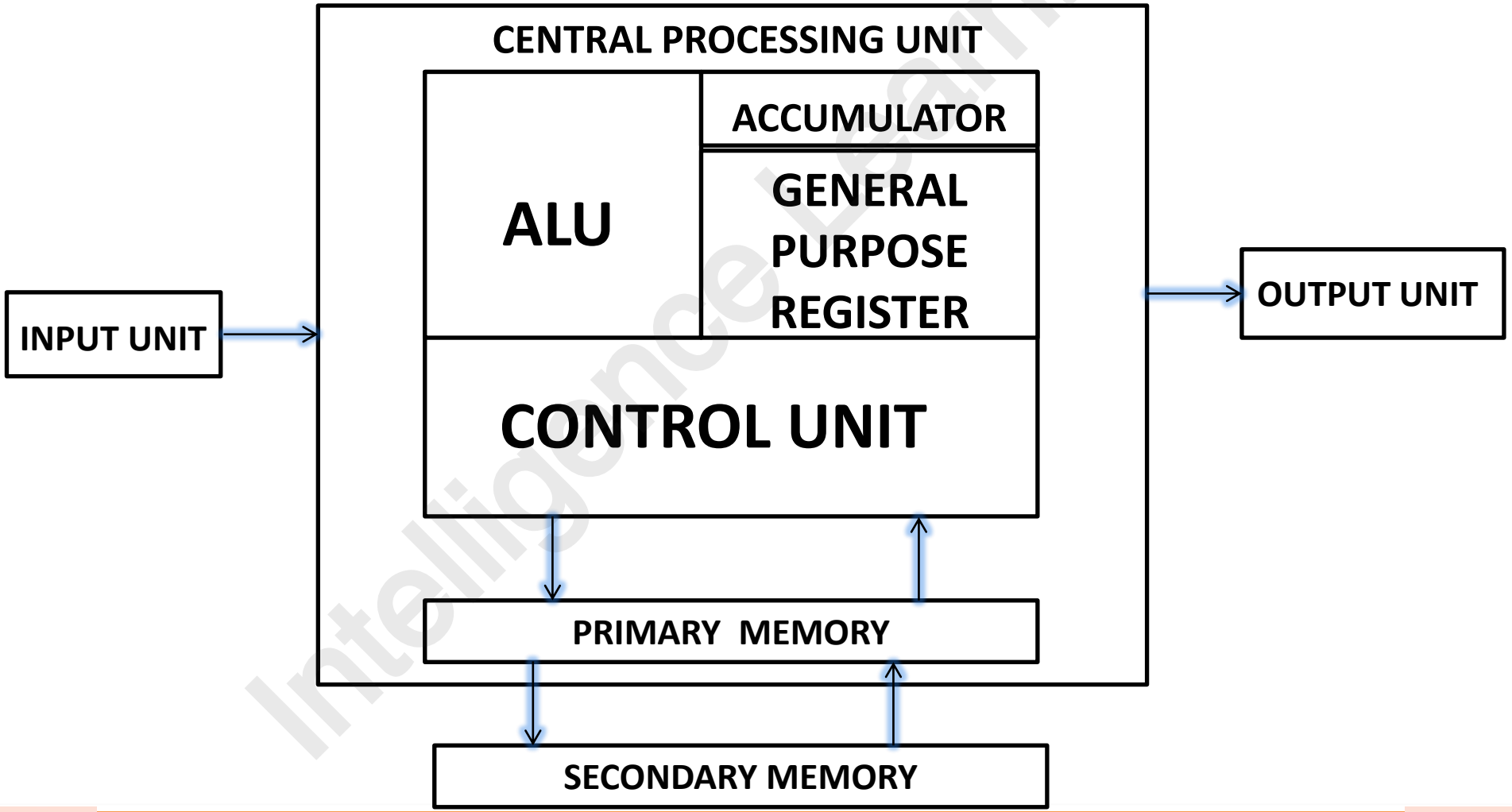
UNIT 1 ONE SHOT

WELCOME

INTELLIGENCE LEARNING

UNIT 1 BCS101 / BCS201: PROGRAMMING FOR PROBLEM SOLVING

BLOCK DIAGRAM OF DIGITAL COMPUTER



Introduction to Components of a Computer System

1. Input Unit:- It links the external environment with the computer system. All Input device must convert the input data into the Binary Codes. • Example:- Keyboard, Mouse etc.

2. Storage Data:- The storage unit of a computer holds data and instructions that are entered through the input unit, before they procesed. It stores programs, data as well as intermediate results and results for output. Its Main function is to store information.

a) Primary Storage (Main Memory):- The memory is generally used to hold the program being currently executed in the computer, the data being received from input device, the intermediate and final result of a program. The primary memory is temporary in nature. The data is lost when the computer is switched off.

b) **Secondary Storage (Auxiliary Memory):-** Secondary storage, also known as auxiliary memory, refers to non-volatile storage devices that store data permanently (or semi permanently), even when the computer is powered off and it is used to store data and programs.

c) Cache Memory (High Speed):- Cache memory is a small, high-speed type of volatile memory located between the CPU and the main memory (RAM).

Introduction to Components of a Computer System

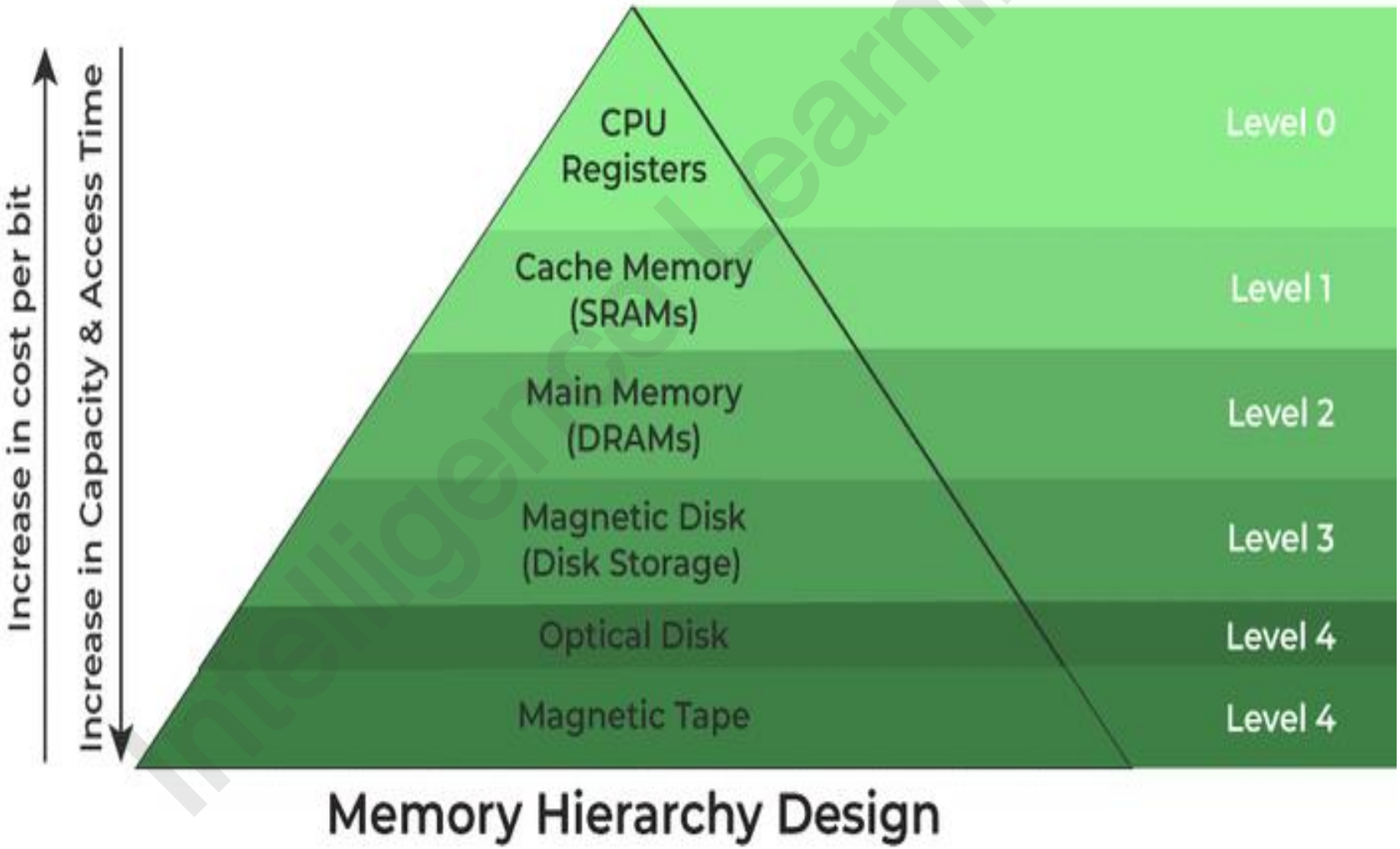
3. CPU (Central Processing Unit):- The CPU (Central Processing Unit) is often called the "brain" of the computer. It is the main component responsible for executing instructions, performing calculations, and controlling all operations of the computer system.

a) **Control Unit (CU):-** The Control Unit is a part of the CPU (the brain of the computer). It works like a manager or a traffic police that guides and controls everything happening inside the computer.

b) **ALU (Arithmetic Logic Unit):-** The Arithmetic Logic Unit (ALU) is a core component of the CPU responsible for performing all arithmetic and logical operations. It is the "calculator" or "decision-maker" inside the CPU.

4. Output Unit:- The Output Unit is the part of a computer that shows the final result to the user after the CPU has finished processing the data.

Draw The Memory Hierarchical Structure of a Computer System.



Differentiate Between RAM And ROM

Feature	RAM (Random Access Memory)	ROM (Read-Only Memory)
Full Form	Random Access Memory	Read-Only Memory
Volatility	Volatile – data is lost when power is off	Non-volatile – data stays even when power is off
Function	Temporarily stores data and programs during use	Permanently stores startup instructions (firmware)
Data Modification	Can read and write easily	Read-only or hard to modify
Used For	Running applications and the OS	Storing BIOS or boot-up instructions
Speed	Faster	Slower than RAM
Size	Usually large (in GBs)	Smaller (in MBs or KBs)
Location	Installed on RAM slots on the motherboard	Embedded in motherboard or chip
Types	DRAM, SRAM	PROM, EPROM, EEPROM
Example	DDR4 RAM in laptops and desktops	BIOS chip, firmware in embedded systems

Differentiate Between Low Level Language And High Level Language. (2017-18, 2020-21

Aspect	Low-Level Language	High-Level Language
Definition	Close to hardware, hard to read	Closer to human language, easy to read
Examples	Machine language, Assembly language	C, Java, Python, C++
Ease of Use	Difficult and time-consuming	Easier to learn and use
Speed of Execution	Very fast, directly interacts with hardware	Slightly slower due to translation (compiler/interpreter)
Portability	Not portable, hardware dependent	Portable across systems
Control Over Hardware	Offers full control	Limited hardware control
Translation Required	Assembler (for Assembly)	Compiler or Interpreter

Brief the generations of the programming language with example.

Generations of Programming Languages (with Examples)

Generation	Name	Description	Example
1st Gen	Machine Language	Binary code (0s and 1s), directly understood by the computer	10110000 01100001
2nd Gen	Assembly Language	Uses mnemonics, easier than binary but still low-level	MOV A, B
3rd-Gen	High-Level Language	More user-friendly, uses English-like syntax	C, Java, Python
4th Gen	Very High-Level Langu-	More abstract; designed for specific purposes like database queries	SQL, MATLAB
5th Gen	Natural Language/ AI-based	Used in AI development, relies on solving problems using constraints	Prolog, Mercury

What is Operating System? What are the various component of Operating System. Describe the functionalities of Operating System. (AKTU 2015-16, 2018-19)

An Operating System (OS) is a special software that helps you use the computer easily. It works like a manager between you and the computer hardware.

Component	Description
1. Kernel	Core part of OS, manages CPU, memory, and devices.
2. Process Management	Handles creation, scheduling, and termination of processes.
3. Memory Management	Manages RAM usage, allocates and deallocates memory.
4. File System	Organizes data in files and directories.
5. Device Management	Controls input/output devices (keyboard, printer, etc.).
6. Security & Protection	Protects system resources and data from unauthorized access.
7. User Interface	Allows users to interact with the system (GUI or CLI).

Functionality	Description
Process Management	Creates, schedules, and terminates processes. Handles multitasking and synchronization.
Memory Management	Allocates memory to programs and ensures efficient usage.
File Management	Creates, stores, retrieves, and deletes files with proper organization.
Device Management	Controls and communicates with I/O devices using drivers.
Security Management	Protects system with passwords and access control.
User Interface	Provides GUI (like Windows) or CLI (like Linux Terminal).
Job Scheduling	Manages execution order of programs based on priority.
Error Detection & Handling	Monitors the system and handles errors properly.

Why Operating System is Required? What are the types / classification of Operating System?

An Operating System (OS) is needed because:

- It starts the computer and helps it work.
- It manages all parts like the keyboard, screen, and printer.
- It helps you run apps like games, browsers, or music players.
- It stores and organizes files (photos, videos, documents).
- It lets you do many things at the same time (like listening to music and writing).
- It keeps your computer safe and helps it work properly.
- Without an OS, the computer is just a box. It won't know how to work.

Type	What it means (Easy Language)
1. Batch OS	In this system, many tasks (jobs) are collected together in a "batch" and run one by one.
2. Time-Sharing OS	Many users can use the computer at the same time. The OS gives each user a small time slot.
3. Distributed OS	Many computers work together and look like one big system.
4. Network OS	It runs on computers that are connected in a network. It allows sharing of files, printers, and internet across computers.
5. Real-Time OS	It works very fast and gives a response in real-time. Used in systems where timing is very important

What is Algorithm ? Write Characteristics of an algorithm ?

An algorithm is a step-by-step set of instructions used to solve a problem or complete a task. It tells the computer what to do and how to do it, in a proper order.

Characteristics of a Good Algorithm

Characteristic	Meaning (Easy Language)
1. Clear and Simple	Every step should be easy to understand.
2. Finiteness	It must end after a limited number of steps.
3. Input	It should take **zero or more inputs**.
4. Output	It must give **at least one output/result**.
5. Effectiveness	Every step must be **simple and possible** to do.
6. Specific	The steps must be **specific**, not confusing.
7. Well-Ordered	The steps should be in **proper order**.

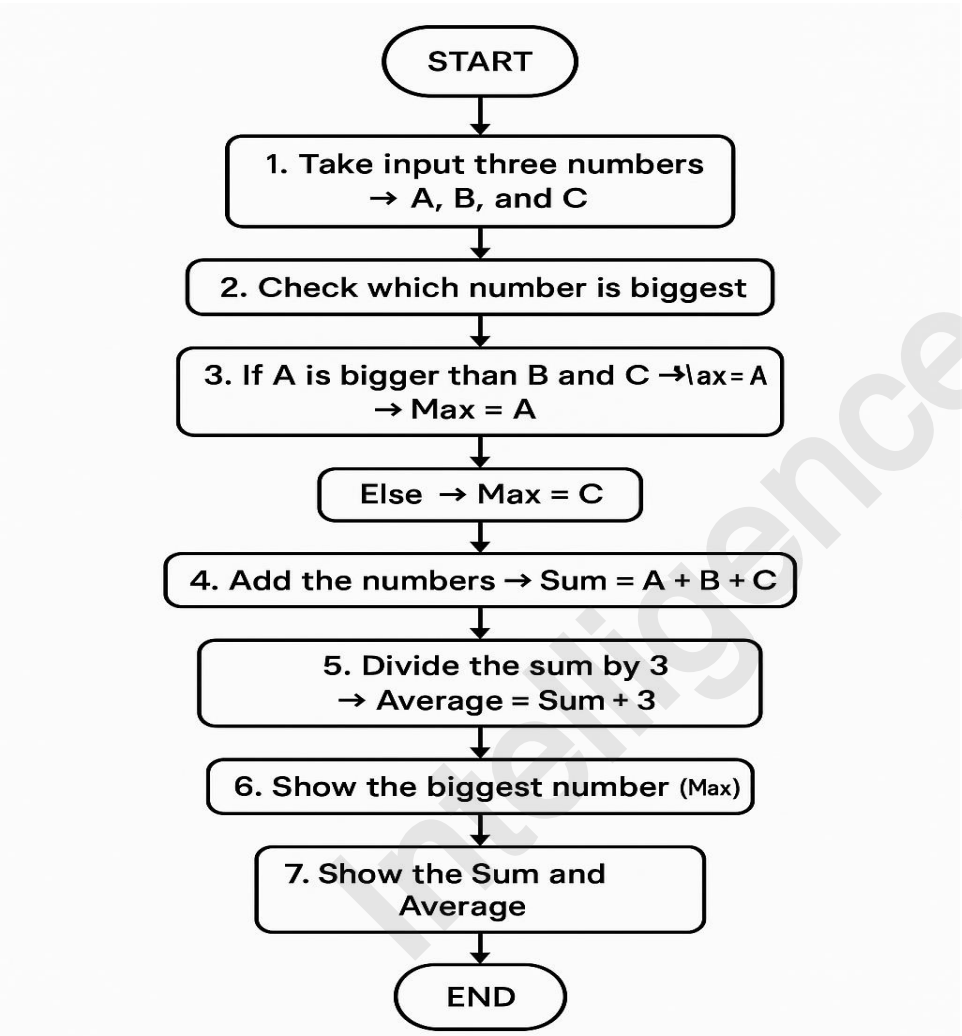
What is flowchart ?

A flowchart is a diagram that shows the steps of a process or algorithm using symbols like arrows, rectangles, and diamonds. It helps to understand and plan a program visually.

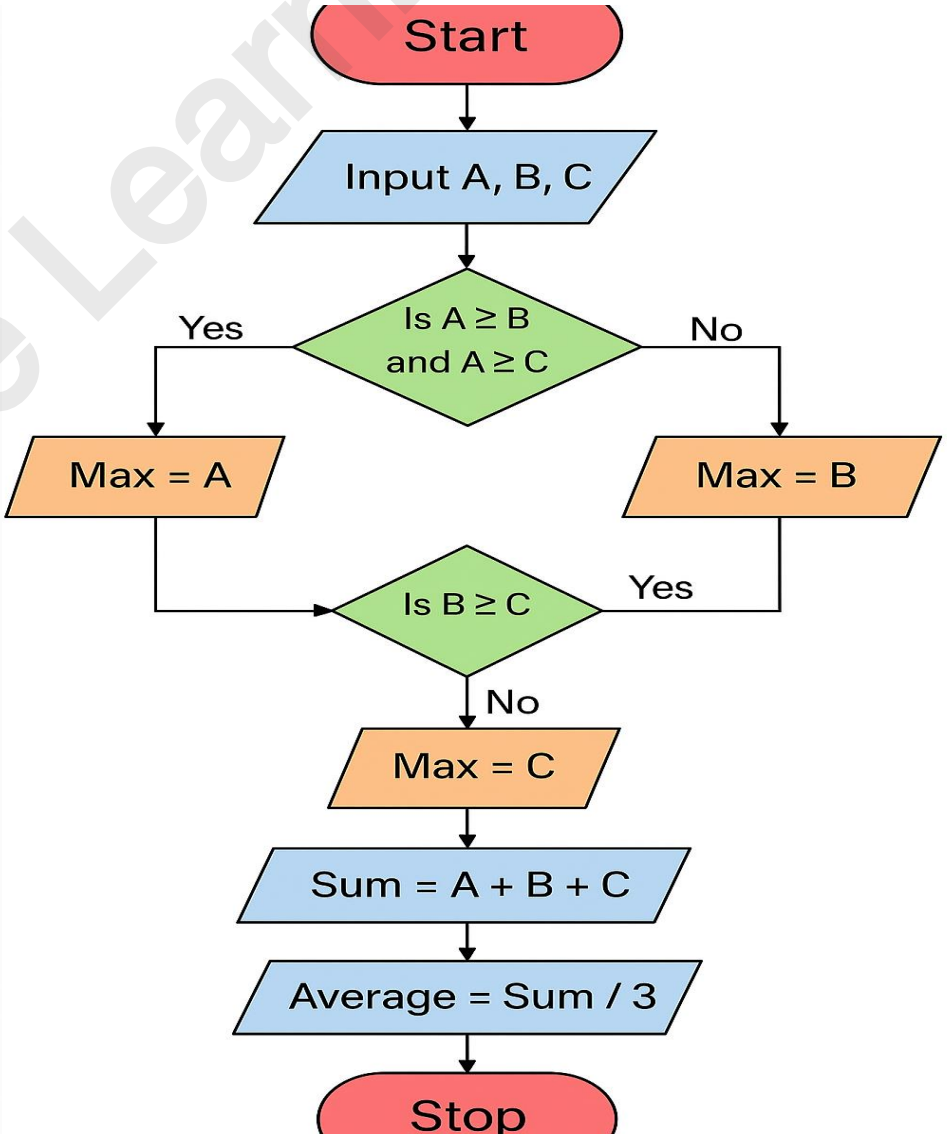
Common Flowchart Symbols:

Symbol	Meaning
◇ Diamond	Decision (Yes/No)
▭ Rectangle	Process/Step
▱ Parallelogram	Input/Output
⬭ Oval	Start/End

Draw a flowchart and write also algorithm to find maximum of three numbers and sum average of three number also.



Algorithm to Find Maximum, Sum and Average of Three Numbers



Draw Flowchart And Write Algorithm to reverse an integer number by the user.

Algorithm to Reverse a Number

1. **Start**

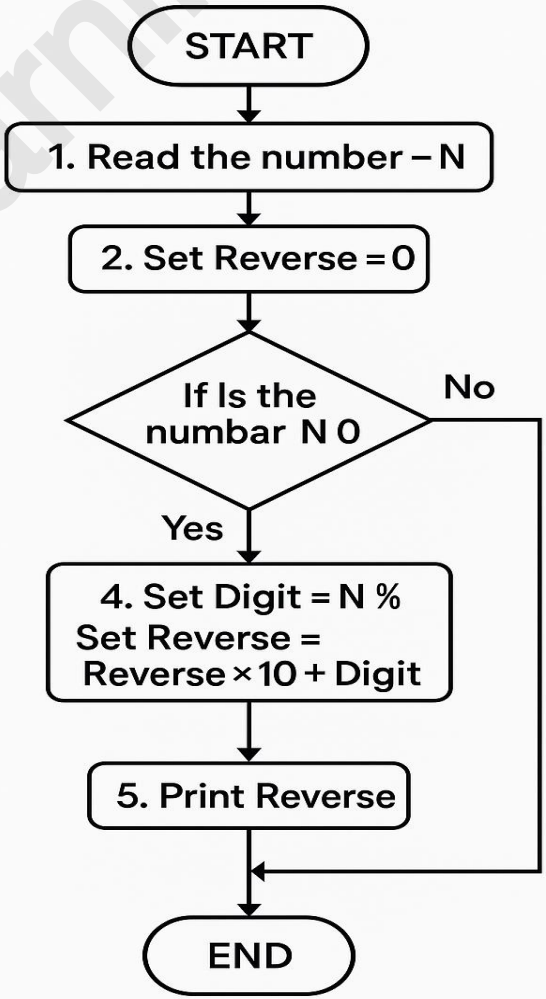
2. **Input** the number (N)

3. Set **Reverse = 0**

4. Repeat while N is not 0:
   a. Get the last digit → **Digit = N % 10**
   b. Add it to Reverse → **Reverse = Reverse × 10 + Digit**
   c. Remove last digit from N → **N = N // 10**

5. Show the Reverse number

6. **End**



Flowchart to Reverse a Number

Difference Between Linker and Loader

Linker (जोड़ने वाला)

1. Linker अलग-अलग code files को जोड़ता है।
2. 2. यह सभी files को मिलाकर एक पूरा प्रोग्राम बनाता है (जैसे .exe)।

3. यह प्रोग्राम चलाने से पहले काम करता है।

1. The linker joins different parts of a program.

2. It puts together code files and libraries.

3. It makes one complete program (like a .exe file).

4. It works before you run the program.

Loader (लोड करने वाला)

1. Loader उस ready program को computer की memory में लोड करता है।

2. ताकि प्रोग्राम चल सके/run हो सके।

3. यह तब काम करता है जब तुम प्रोग्राम को open/run करते हो।

1. The loader loads the program into memory (RAM).

2. It helps the program start running.

3. It works when you open or run the program.

Difference Between Algorithm, Flowchart, and Pseudocode

Algorithm

1. An algorithm is a step-by-step written method to solve a problem.
2. It uses simple language, like instructions.
3. It does not follow any coding or drawing rules.

Pseudocode

1. Pseudocode means "false code" – it looks like code, but is written in plain English.
2. It's used to plan a program before writing real code.
3. It has no strict syntax, but looks a little like programming.

Flowchart

1. A flowchart is a diagram showing steps using shapes and arrows.
2. It is visual, not written like code.
3. Helps people understand the process quickly.

Why C Programming is Structured Programming

C is called a structured programming language because it organizes code into functions and blocks, making it easy to read, manage, and reuse.

Write a note on Top – Down Program Development Approach ?

The **Top-Down approach** is a method of program design where:

1. You **start with the main idea or big problem**,
2. Then **break it into smaller parts or steps**,
3. And finally **write code for each small part** one by one.

How Binary File Differ From Txt File ?

A text file stores data in human-readable characters (like letters and numbers), while a binary file stores data in 0s and 1s, which computers can read but humans cannot. Text files are easy to read and edit, whereas binary files are faster for computers to process and often smaller in size.

Explain the basic Structure of C ?

```
#include <stdio.h>      // 1. Preprocessor Directive

int main() {            // 2. Main Function
    int a, b, sum;      // 3. Variable Declaration

    a = 5;              // 4. Logic / Processing
    b = 10;
    sum = a + b;

    printf("Sum = %d", sum);      // 5. Output Statement

    return 0;           // 6. Return Statement
}
```

Why do use curly bracket in C

We use curly brackets { } in C to group multiple statements together as a block, especially in functions, loops, and conditionals.

Difference Between Syntax Error, Run-Time Error, Logical Error, and Input Data Error

Type of Error	Meaning	When It Happens	Example
Syntax Error	Mistake in writing code rules (grammar error)	At compile time	Missing ; or misspelled keyword
Run-Time Error	Error that occurs while the program is running	During program execution	Dividing by zero
Logical Error	Code runs but gives **wrong result** due to wrong logic	After program runs	Using + instead of *
Input Data Error	Error caused by **wrong user input** or data format	During or before execution	Entering a letter when number is needed

Tokens in C (or Components of C)

Tokens are the smallest building blocks of a C program.

There are **6 main types of tokens** in C

Keywords – Reserved words with special meaning. (int, if, return)

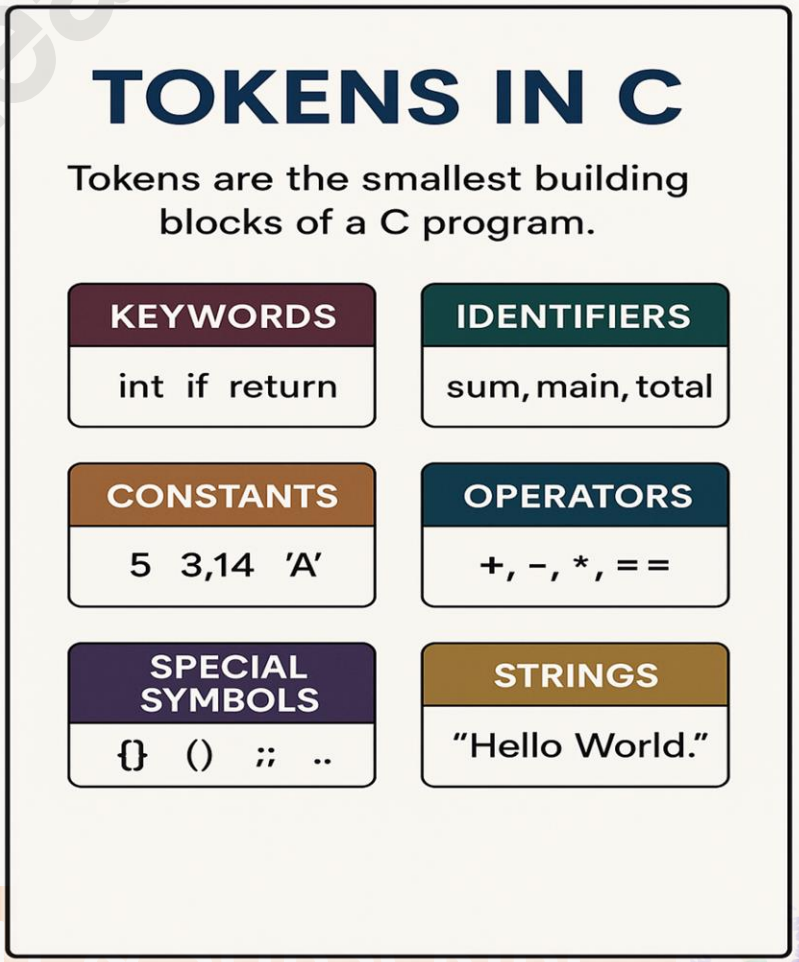
Identifiers – Names used for variables, functions, etc. (sum, main, total)

Constants – Fixed values that do not change. (5, 3.14, 'A')

Operators – Symbols used for operations. (+, -, *, ==)

Special Symbols – Characters with special meaning. ({}, (), ;, ,)

Strings – A group of characters inside double quotes. ("Hello World")



Define Data Type in C. Discuss in Terms of Memory Size, Format Specific and Range.

A data type in C defines the type of data a variable can store, such as integer, float, character, etc. It also tells the amount of memory required, the value range, and the format specifier used in functions.

Data Type	Format Specifier	Size	Range
char	%c	1 byte	–128 to +127
int	%d	2 bytes	–32,768 to 32,767
short int	%d	2 bytes	–32,768 to 32,767
long int	%ld	4 bytes	–2,147,483,648 to 2,147,483,647
unsigned int	%u	2 bytes	0 to 65,535
unsigned short int	%u	2 bytes	0 to 65,535
unsigned long int	%lu	4 bytes	0 to 4,294,967,295
float	%f	4 bytes	3.4E–38 to 3.4E+38
double	%lf	8 bytes	1.7E–308 to 1.7E+308
long double	%Lf	10 bytes	3.4E–4932 to 1.1E+4932

THANK YOU FOR WATCHING

DO LIKE SHARE COMMENT AND SUBSCRIBE