PROGRAMMING FOR PROBLEMSOLVING (ANURAG GUPTA)

UNIT - 1

Generations of Computers:

It refers to the different advancements of new computer technology. With each new generations of computer, the circuitry becomes smaller and more advanced than that used in previous generation. The focus of every new generation has been on speed, power, reduce size and efficient computer memory.

Generations of computer classify into 5 parts:

1) First Generation (1940-1956): Vaccum tubes
2) Second Generation (1956-1963): Transistors
3) Third Generation (1964-1971): Integrated Circuits (IC)
4) Fourth Generation (1971-1989): Microprocessors
5) Fifth Generation (Present and Beyond): Artificial Intelligence (AI)

1) First Generation: • First generation computers used a very large number of vacuum tubes for circuitry and magnetic drums for memory.
- support machine language (0, 1).
- Needed a lot of electricity, and thus generated lot of heat.
- Very expensive and big in size.

Advantage: They were the fastest calculating devices of their time.

Disadvantage: • Generated a lot of heat.
- consumed a lot of electricity.
- very bulky in size.
- very expensive.

Example:
- UNIVAC in 1951 (Universal automatic computer)
- ENIAC (Electronic numerical integrator & calculator)

Second Generation: • second-generation computers were manufactured using transistors.

- Transistors were invented in 1947 but were used for manufacturing computers only in the late 1950s.
- Computers manufactured using transistors were smaller, faster, cheaper, more energy efficient and reliable.
- In this computers moved towards assembly languages.
- At this time, high-level languages such as COBOL, FORTR ALGOL were also being developed.

Advantages: • Consumed less electricity as compared to first-generation computers.

- They were faster, cheaper, smaller and more reliable than first-generation computers.
- They could be programmed using assembly language and high-level language.

Disadvantage: Computers were manufactured using transistors, which had to be assembled manually. This made commercial production of computers difficult and expensive.

Third Generation: • The development of Integrated Circuit (IC) was the hallmark of the third generation of computers. several electronic components such as transistors, registors, and capacitors were minimized and placed on silicon chips, calle I.C.

- Some more high-level languages such as PASCAL and BASIC were introduced at this time.

Advantages: • They were faster, very smaller, cheaper and more reliable than their predecessors.
• Widely used for scientific and business applications.

Disadvantage: • Difficult to maintain.
• Heated very quickly.

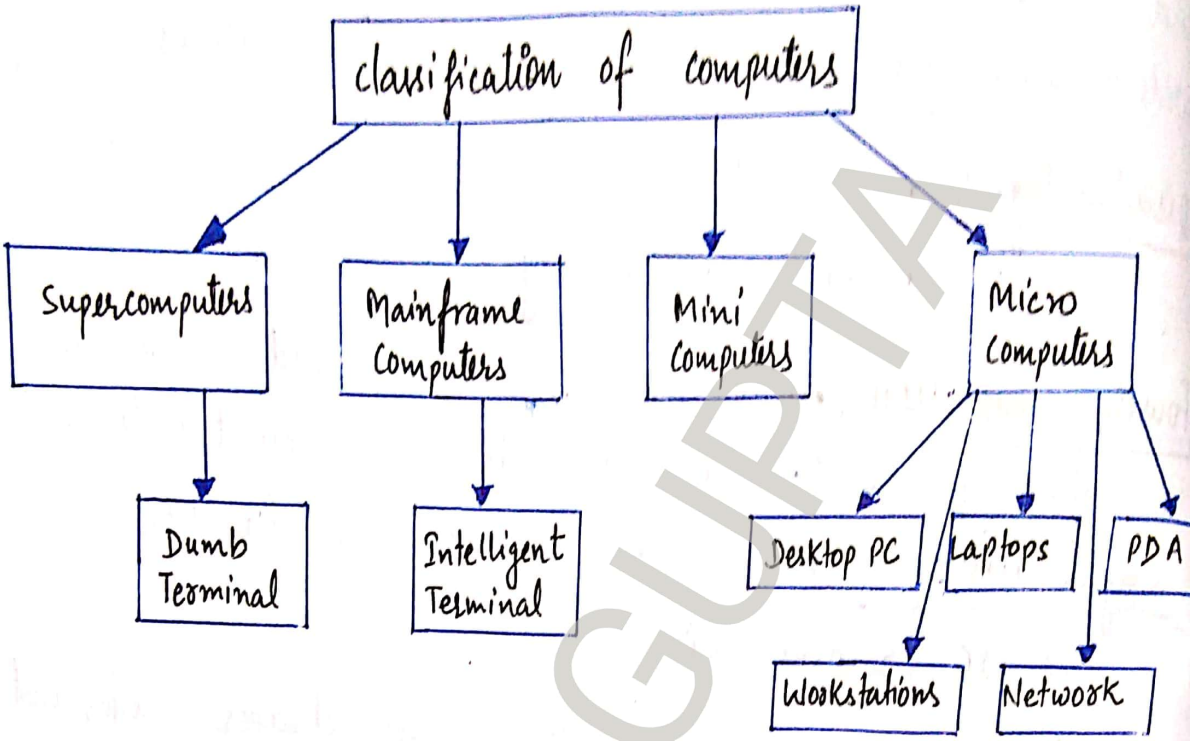
Fourth Generation: • The microprocessor launched the fourth generation of computers, with thousands of IC built onto a single silicon chip. For ex. Intel 4004 chip in 1971.
• In this IC becomes VLSI.

Advantages: • These computers were smaller, cheaper, faster and more reliable than their predecessors.
• These could be used as general-purpose computers.
• GUI (Graphical-user Interface) enabled computers.

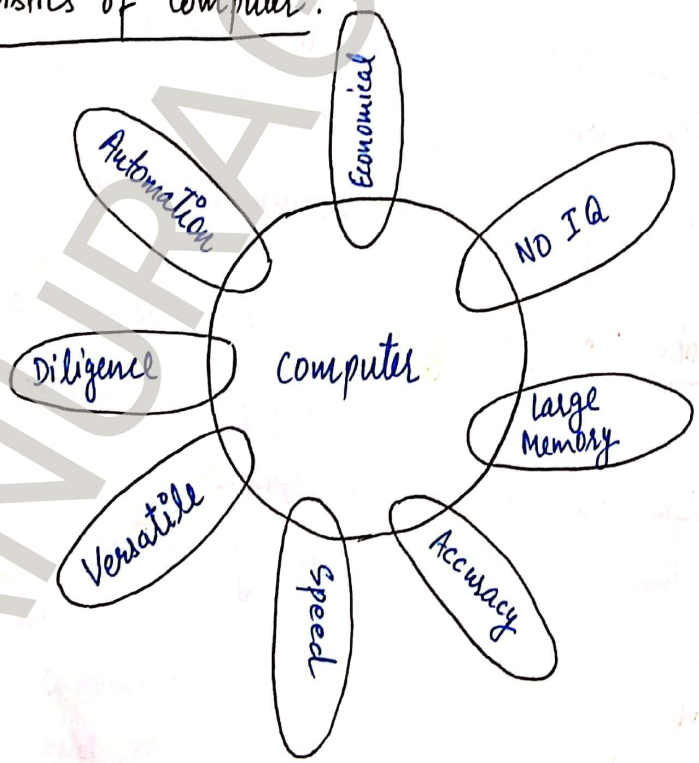
Disadvantage: They were not intelligent systems.

Fifth Generation: • Fifth generation computers are completely based on the new concept of Artificial intelligence (AI).
• Such computers are still in development.
• AI touches the following areas, among others:
- Gaming
- Expert system
- Natural languages
- Neural Networks
- Robotics
- Machine learning

classification of computers:



characteristics of computer:

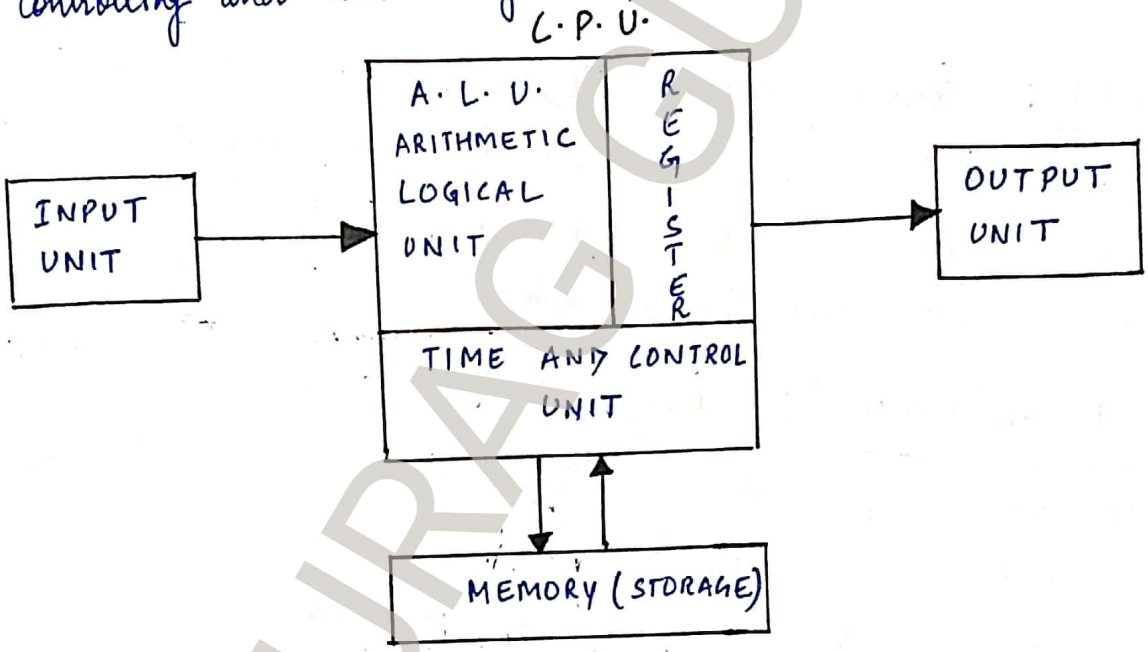


Introduction to components of Computer System:

DIGITAL COMPUTER: Digital Computer is an electronic device that is used to perform Arithmetic and logical operations.

Some major operations are:

- Accepting data or instructions (Input)
- Storing data
- Processing data
- Displaying results (output)
- Controlling and coordinating all operations inside a computer.



BLOCK DIAGRAM OF DIGITAL COMPUTER

Input: • This is the process of entering data and instructions (also known as program) in the computer system.

• The data and instructions can be entered by using different "input devices" such as:

Keyboard, mouse, Scanner, Pen, Joystick, touch screen

Output: • Output is the process of giving the result of data pro to the outside world (external to the computer system).

• The results are given through "output devices" such as: Monitor, printer, projector, speakers, Touch screen.

Processing: • The process of performing operations on the data as per the instructions specified by the user (program) is called processing.

• Processing, controlling and coordinating done by C.P.U.

C.P.U.: (Central processing Unit): The CPU is the brain of th computer. It performs all calculations and controls the devices connected to computer system. The faster the CPU, the quicker programs can process the instructions.

Storage (Memory): Storage is the process of saving data an instructions permanently in the computer so that they ca be used for processing.

A computer has two types of storage areas:

Primary storage (Main Memory): • It is the storage area that is directly accessible by the CPU at very high speeds.

• Primary storage space is very expensive and therefore limite in capacity.

• Primary storage is volatile (Temporary) in nature; that is, as soon as the computer is switched off, the information stored gets erased.

• Ex: RAM (Random Access Memory).

- Secondary Memory (Auxiliary Memory): This is just the opposite of primary memory.
- It is cheaper, Non-volatile (Permanent), and used to permanently store data and programs.
- It has more storage space, less expensive.

Ex: Harddisk.

(ANURAG GUPTA)

Comparsion of RAM and ROM:

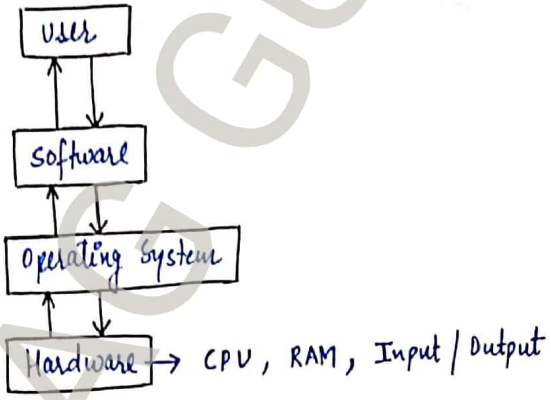
RAM (Random Access Memory)	ROM (Read-only Memory)
• It is a read-write memory. • It is a volatile memory. • RAM sizes is from 64MB to 16GB. • Data in RAM can be modified. • RAM is costly. • Types of RAM are: - Static RAM - Dynamic RAM	• It is read-only memory • It is a non-volatile memory. • ROM is comparatively smaller than RAM. • Data in ROM cannot be modified. • ROM is cheaper than RAM. • Types of ROM are: - PROM - EPROM - EEPROM

Software: Set of Programs is known as software and software are of two types:

1) System Software : Ex: Operating System, compiler, Interpreter, BIOS.
2) Application Software: Ex: MS-office, Games, etc.

Operating System: • An operating system is a collection of system programs that controls the operations of the computer system.

• Operating system is the Interaction between user and hardware.



Functions of Operating System:

1) Manages the computer hardware: The OS controls and efficiently utilizes hardware components such as CPU, memory, and I/O Devices.

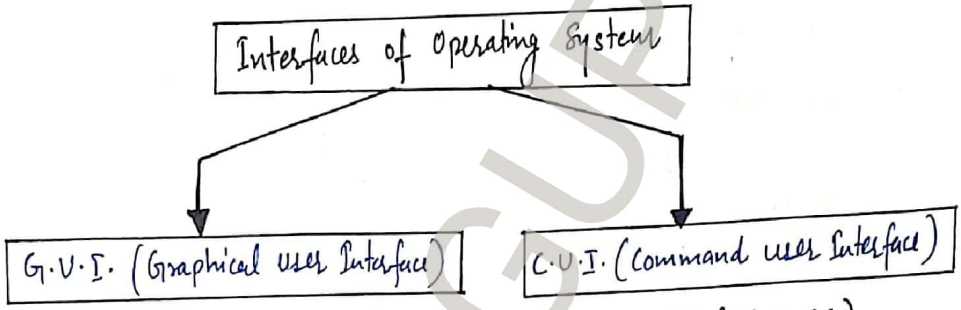
2) Provides a user Interface: The OS enables users to easily interact with the computer hardware.

3) Process Management: The OS enables a user to execute more than one job at the same time to enhance productivity.

4) Memory Management: Finding vacant spaces in the primary memory, loading the appropriate data and programs in the located space, execute them, & removing them from memory done by OS.

5) File Management: The OS allows users to create, copy, delete and rename files.

6) Security Management: The OS protects stored information from users. It ensures that the data and files stored cannot be accessed by unauthorized users.



ex: Windows
Android
LINUX

ex: MS-DOS (Disk OS)
UNIX

UNIX : • UNIX OS was first developed in 1960s, and since then it has been under constant development.

• UNIX is a stable, multi-user, multi-tasking operating system for servers, desktops, and laptop computers.

LINUX: LINUX is a very powerful, free, open-source operating system based on UNIX.

Components of LINUX System: LINUX has 3 components:

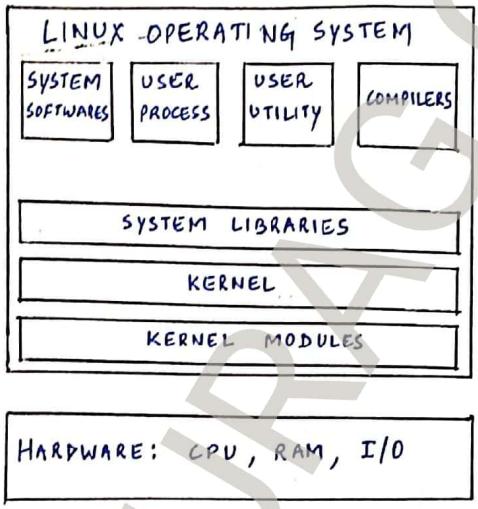
1) Kernel: Kernel is the core part of LINUX. It is responsible for all major activities of this OS. It is consists various modules and it interacts directly with the hardware.

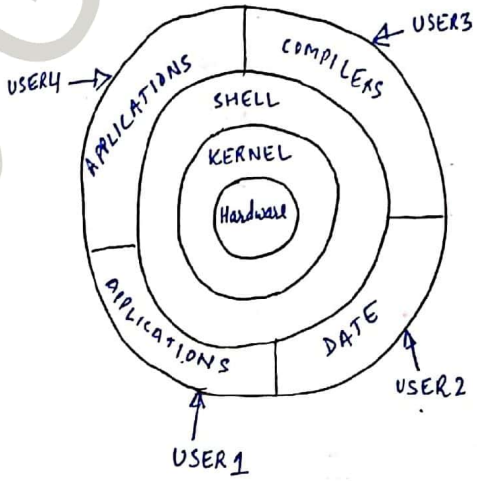
2) System Library: System Libraries are special function or program using which application program or system utilities accesses Kernel's features.

3) System Utility: They are responsible to do specialized, individual level tasks.

LINUX Architecture: LINUX system architecture is consists of following layers:

1) Kernel
2) Utilities
3) Hardware Layer: Hardware consists of all peripheral (I/O) devices (RAM / HDD / CPU etc).
4) Shell: An interface to kernel, hiding complexity of kernel's functions from users. Takes command from users and executes kernel's functions.





ANDROID: • Android is a mobile OS, currently developed by Google, based on the linux kernel and designed primarily for touch screen mobile devices such as smart phone and tablets.

• Android is open source which has encouraged a large community of developers to use the open source code for projects, which add new features for advanced users.

Computer Languages: computer understand its own language i.e. 0s and 1s (Binary Language).

It has two Basic types:

1) Low-level language (close to machine) → Machine Language
                                          → Assembly Language

2) High-level Language (close to humans) (Ex: C, C++, JAVA, etc)

Translators: As computer understands only 0's and 1's, so computer can not understand anything written other than 0's and 1's.
So Programs written in other languages must be translated into machine language using translator.
Translator are of 3 types:

1) Assembler
2) Compiler
3) Interpreter

1) **Assembler:** A program which translates an assembly language program into a machine language program is called an assembler.

Assembly Language —Assembler→ Machine Language

2) **Compiler:** It is a program which translates a high-level language program into a machine language program.

High level Language —Compiler→ Machine Language

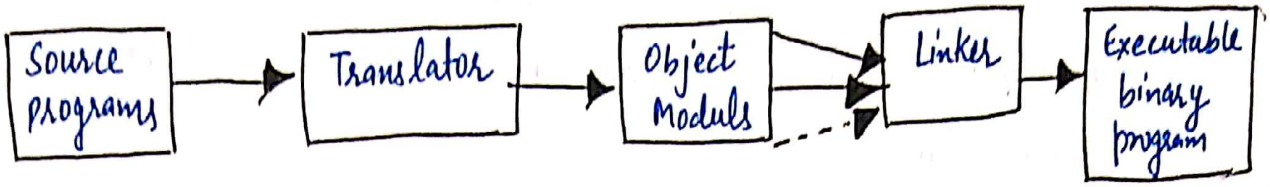
Interpreter: An interpreter is a program which translates statements of a high level language program into machine language.

High level Language (statements) → Interpreter → Machine Language

Difference between compiler and Interpreter:

Compiler	Interpreter
1. scan the entire program and translates it as a whole into machine code.	1. Translates programone statement at a time.
2. It takes less time to execute.	2. It takes more time to execute.
3. Processing is Fast.	3. Processing is slow.
4. C, C++ use compiler.	4. Python, Ruby use interpreter.

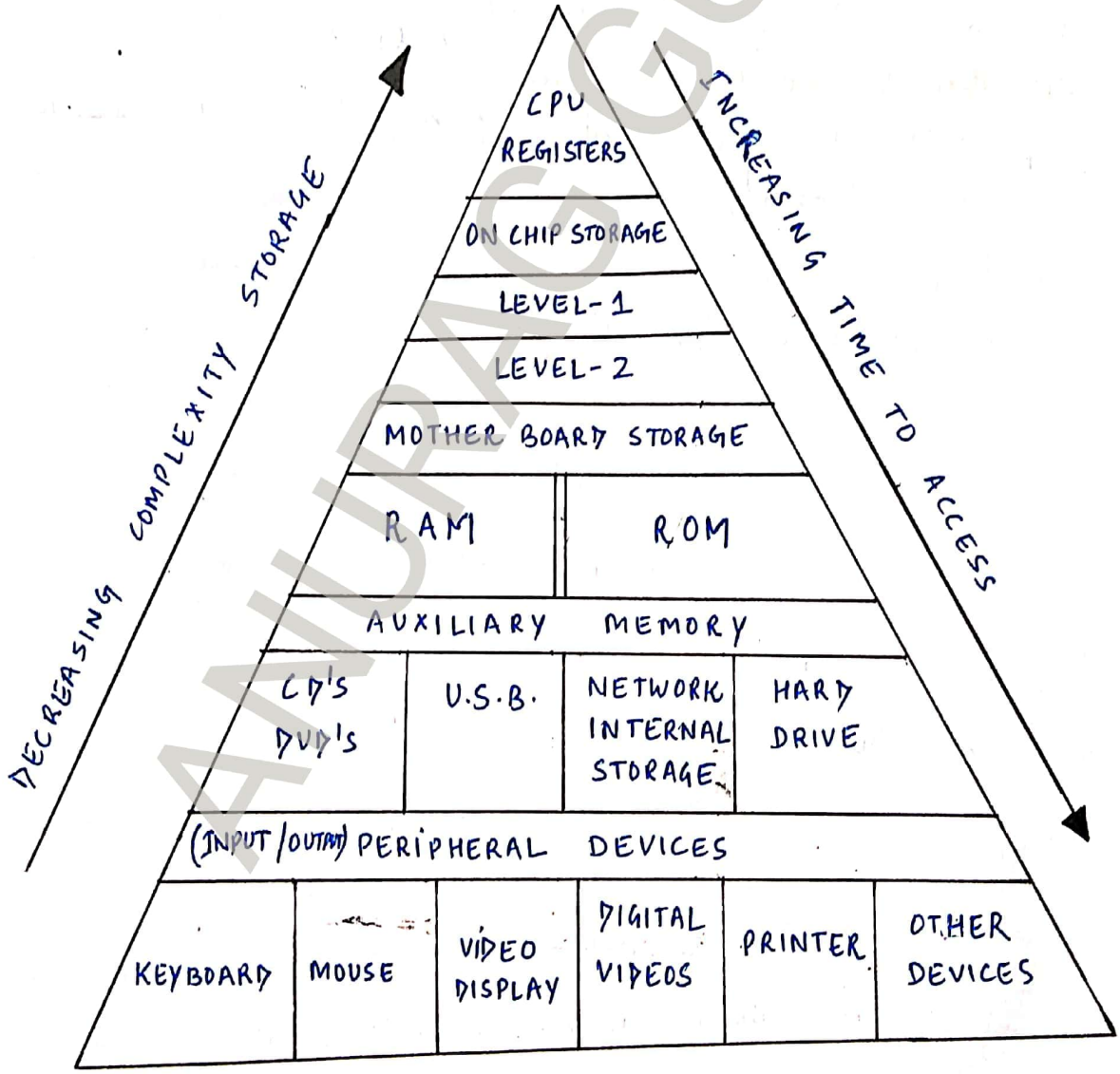
LINKER:
- In high level languages, some built in header files or libraries are stored. These libraries are pre-defined and contain basic functions which are essential for executing the program. These functions are linked to the libraries by a program called LINKER. (Also known as link editor or binder).
- If linker does not find a library of a function then it informs to compiler & then compiler generates an error.
- Compiler automatically invokes the linker as last step in compiling a program.



LOADER :- A loader is a special type of program that copies programs from a storage device to the main memory, where they can be executed.

- Most loaders are transparent to the users.

MEMORY HIERARCHY:



MEMORY HIERARCHY STRUCTURE

NUMBER SYSTEM

"A system for representing (expressing or writing) numbers of a certain type."

Number system is represented by its base (also called radix).

We generally have four Number system

1. Decimal Number System :
   Base (Radix) 10, Digits used 0 to 9.
2. Binary Number System
   Base (Radix) 2, Digits used 0, 1
3. Octal Number System
   Base (Radix) 8, Digits used 0 to 7
4. Hexa Decimal Number System
   Base (Radix) 16, Digits used : 0 to 9, Letters used : A – F
   where A → 10, B → 11, C → 12, D → 13, E → 14, F → 15

* And we have some numericals on number system. (conversion of number system).

Problem Solving Approaches (Algorithm, flow chart, pseudo code):

"Problem is a kind of barrier to achieve something and problem solving is a process to get that barrier removed by performing some sequence of activities!!

Idea of Algorithm: " step by step solution of a problem is known as algorithm."

characteristics of algorithm: There are five characteristics of algorithm:

1) Input: Algorithm may have zero or more input.
2) Output: It should produce at least one output.
3) Finiteness: Total number of steps used in algorithm should be finite.
4) Definiteness: Each steps of algorithm must be clear and un ambiguous.
5) Effectiveness: Every step must be basic and essential.

Types of Algorithm: Algorithms are classified to the three types of control structures:

1. Sequence

step1: Input 1st Number A.
step2: Input 2nd Number B.
step3: set Sum = A+B.
step4: PRINT Sum
step5: End

2. Branching (Decision) (selection)

step1: Input 1st Number A.
step2: Input 2nd Number B.
step3: if A == B
then PRINT "equal"
else PRINT "Not equal"
step4: END

3. Loop (Repetition)

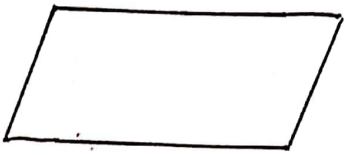
step1: [Initialization] set I=0, N=10
step2: Repeat step while I<=N
step3: PRINT I
step4: END

FLOWCHARTS : • A flowchart is a graphical or symbolic repres
of a process. It is basically used to design and documen
virtually complex processes to help the viewers to visualize
logic of the process.

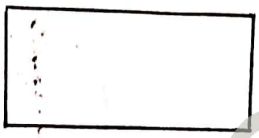
• Flowchart designing depicted by a different symbol and sy
of a flowcharts include:

start or end symbol

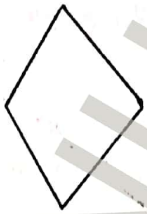
connector



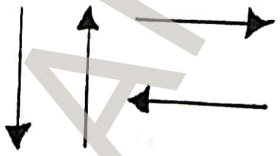
Input/output symbol



Processing step



Decision symbol

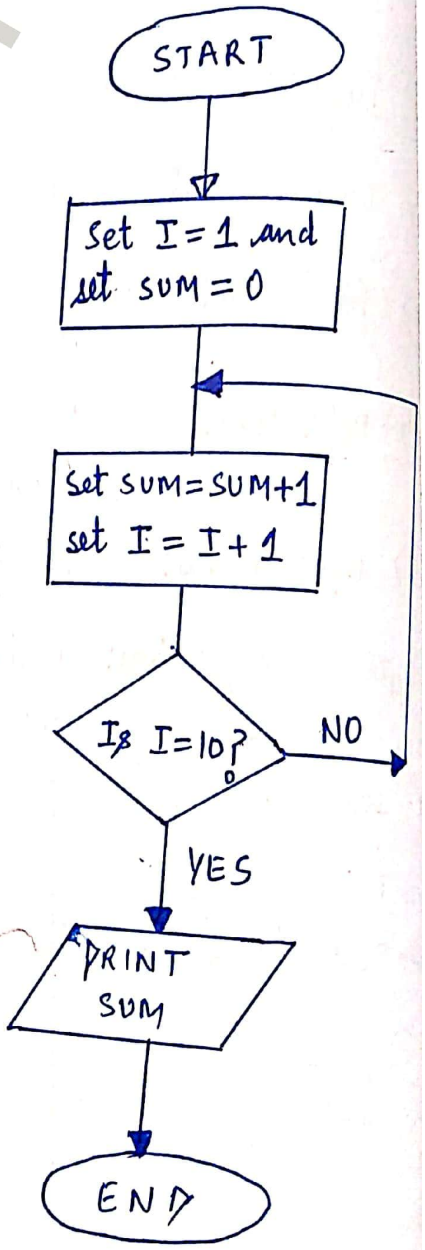


FLOW CONTROL (ARROWS)

Example:

Q. Draw a flowchart to calculate the sum of first 10 natural numbers.

sol?



PSEUDOCODE: Is a simple way of writing programming code in English. Pseudocode is not programming language. It uses short-phrases to write code for programs before we actual create it in a specific language.

Ex: Pseudocode for add two Numbers.

```
start program:
    Enter two Numbers A and B
    Add the number together
    PRINT Sum.
End program.
```

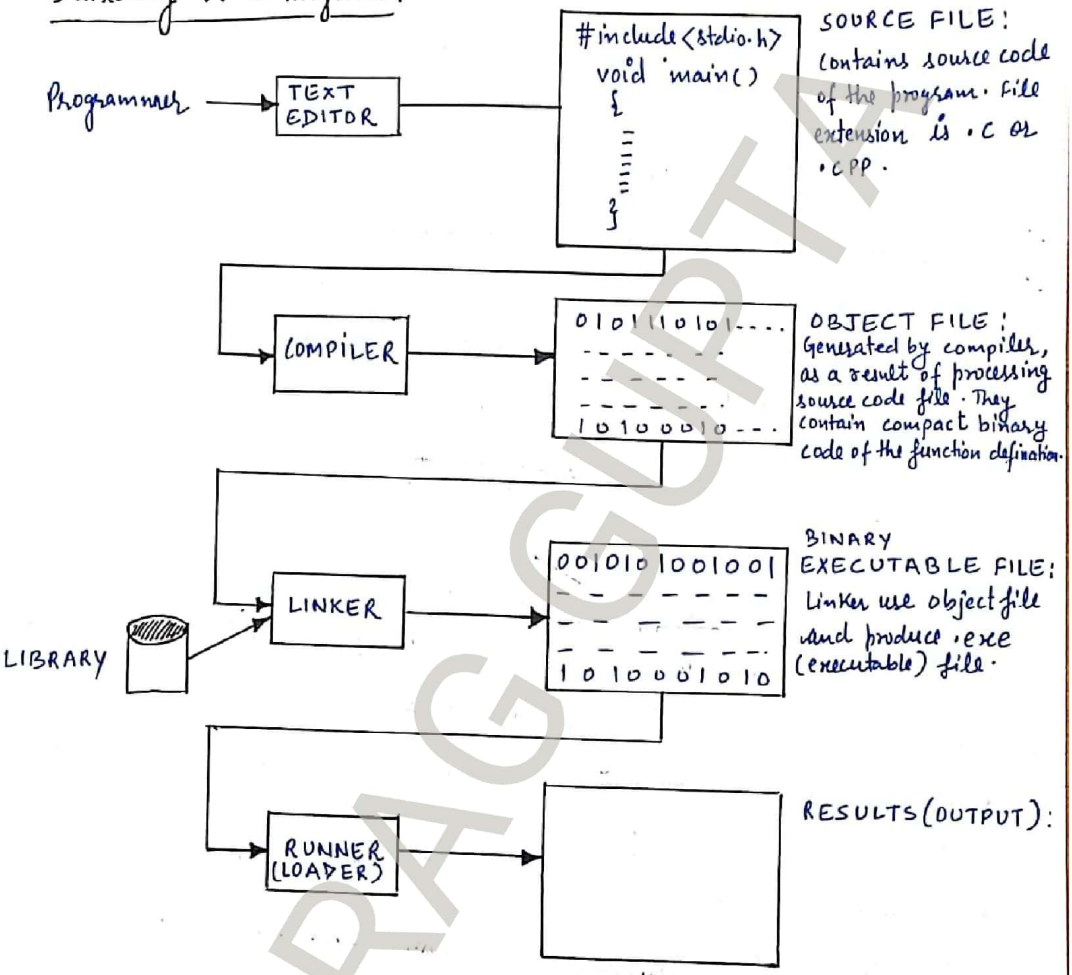
C Programming Language

- C Programming language was developed in 1972 by Dennis Ritchie at BELL Labs to develop the UNIX operating system.
- C Programming language took concepts from ALGOL, BCPL and B, and added some features of his own.
- C has no full form, this is the successor language of B.

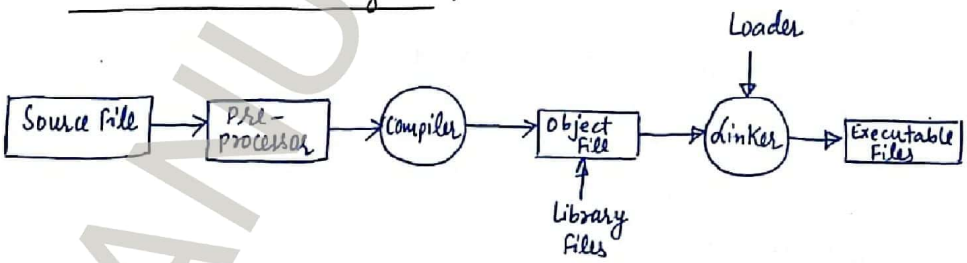
Characteristics of C Programming:

1) C is structured programming language.
2) C is a case-sensitive language.
3) C is Portable language.
4) C is simple, fast and efficient.
5) C has rich set of data type.
6) C has rich set of operators.

Building a C Program:



COMPILING C Program:



Syntax and logical errors in compilation :

Syntax error is also known as compilation error, these are mistakes such as misspelled keywords, missing punctuation character, missing bracket, etc.

Logical errors are those errors that prevent your program from doing what you expected it to do. logical error is also known as run time error.
In this your code may compile but the result is not the expected one.

C Program structure : Structure of C Program is defined by set of rules called protocol, to be followed by programmer while writing C program.

```
Preprocessor Directives
Global Declarations
main() function
{
  local variables
  statements
  ------
  ------
}
function 1()
{
  local variables
  statements
  ------
}
function N()
{
  local variables
  statements
  ------
}
```

Structure of C Program

Your First C Program :

```
#include <stdio.h>
#include <conio.h>
  void   main()
  {
   // my first program in C
    printf("Hello, world!");
     getch();
  }
```

Output : Hello, World!

Character set of C : Every programming language has its own set of characters which are used to write the statements of the program (source code).
In C language also, program is written by set of characters.
These character are:

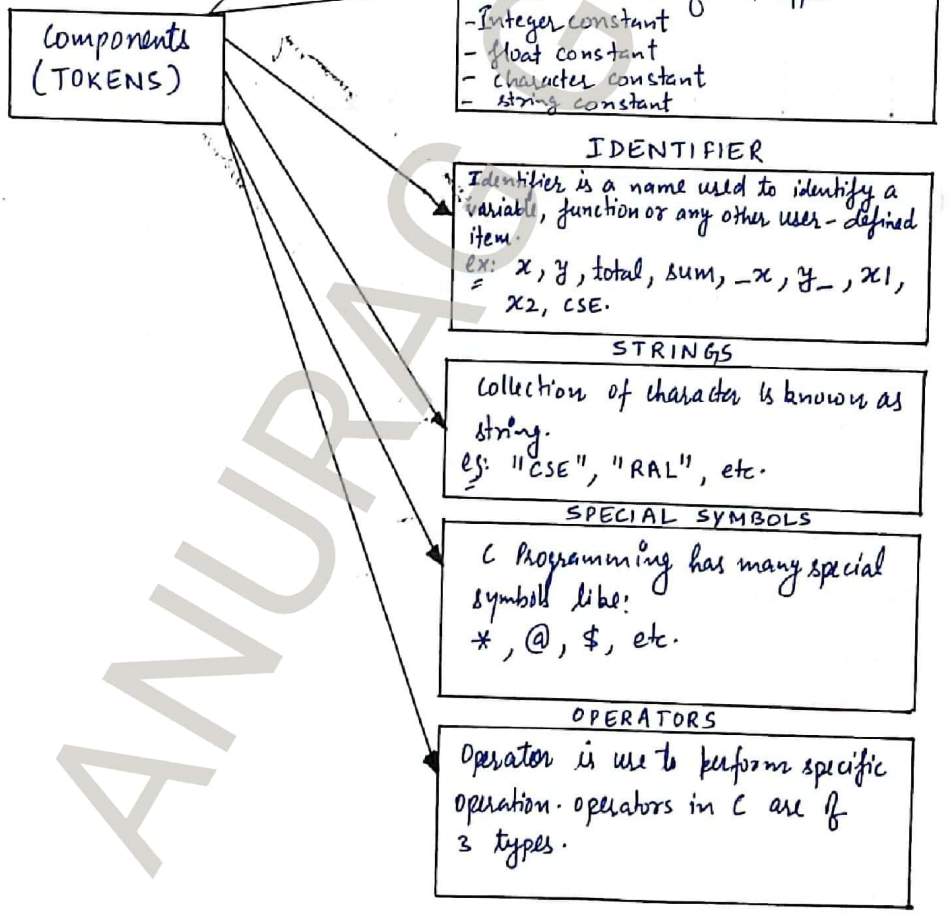
1. lowercase letters : a, b, c, ... z.
2. uppercase letters : A, B, C, ... Z.
3. Digits : 0, 1, 2, ... 9
4. Special characters : +, -, /, =, (,), [,], {, }, ", ', !, #, @, $
5. white spaces : blanks, newline, tab etc.

C- Header files :

	Name	Description
1.	stdio.h	Input / Output functions.
2.	conio.h	console Input / Output.
3.	ctype.h	character handling functions.
4.	math.h	Mathematics functions
5.	string.h	string functions.

Component of C Program (C-Tokens): This is the smallest unit of C Program which are constructed together to write a C Program.

C Tokens are of 6 types:



Data Types : • Data type define the type of data a variable can hold.

• Data types refer to an extensive system used for declaring variables or functions of different types.

There are Five data types in C Language. They are,

S.No.	Types	Data Types
1.	Basic Data types	int, float, char, double
2.	Enumeration data type	enum
3.	Derived data type	pointer, array, structure, union
4.	Void data type	void
5.	User-defined data type	typedef

Qualifiers (Modifiers) in C: Modifiers are prefixed with basic data types to modify (either increase or decrease) the amount of storage space allocated to a variable.

There are 5 qualifiers available in C Language. They are,

1. short
2. long
3. signed
4. unsigned
5. long long

Data type Table :

S.No.	C Data types	Storage Size (In Bytes)	Format Specifier	Range
1.	int	2	%d	-32768 to +32767
2.	long int	4	%ld	-2147483648 to +2147483647
3.	short int	2	%d	-32768 to +32767
4.	signed int	2	%d	-32768 to +32767
5.	unsigned int	2	%d	0 to 65535
6.	unsigned long long int	8	%lld	$2^{64}-1$
7.	float	4	%f	$3.4E^{-38}$ to $3.4E^{38}$
8.	double	8	%lf	$1.7E^{-308}$ to $1.7E^{308}$
9.	long double	10	%Lf	$3.4E^{-4932}$ to $3.4E^{4932}$
10.	char	1	%c	-128 to +127
11.	signed char	1	%c	-128 to +127
12.	unsigned char	1	%c	0 to 255
13.	void	—	—	—
14.	long long int	8	%lld	$-(2^{63}-1)$ to $(2^{63}-1)$
15.	signed long int	4	%ld	-2147483647 to 2147483647

Escape sequence : C supports the combination of backslash (\) and some characters from the C character set to print these characters.

\a	Bell (speaker beeps)	\\	Backslash
\f	clear screen	\?	question mark
\n	New line	\'	single quote
\t	Tab	\"	Double quote
\r	carriage Return	\b	back space

ASCII (American standard code for Information Interchange):

	ASCII value
a	97
z	122
A	65
Z	90
0	48
9	57

Standard I/O (Input/Output) in C: Input - Output functions can read and write values of all data types.

These are of two types:

1. Formatted I/O function: In this we have to specify the type of data we are going to use. (we have to specify format specifier).

for ex: printf() → Output function
scanf() → Input function

Syntax: printf("user defined message");
printf("Format Specifiers", value1, value2, ...);
scanf("Format Specifiers", &value1, &value2, ...);

2. Unformatted I/O function: In this we do not have to specify the type of data we are going to use. (No need to specify format specifier).

for ex: puts(), putchar() → Output function
gets(), getchar() → Input function

Variable: A variable is an identifier which holds data or another one variable.

Syntax: variable name = value;
↓
can be user-defined

Variable declaration:

Datatype variable name;

ex: int a;
= float b, c, a;

Variable Initialization: Give value to variable.

datatype variable name = value;

ex: int a = 10;
= int a = x;

Scope of variable: In C, variable can be either local or global.

→ Global variables defined outside of all the functions, generally on top of the program.

→ local variable defined within the body of a function or a block.

eg:

```
#include <stdio.h>
#include <conio.h>
  int a;                    // Global Variable (outside function)
  void main()
  {  int b;                 // local Variable (within function)
     a = 10;
     b = 20;
  printf ("%d", a);
  printf ("%d", b);
     getch();
  }
```

Storage class: Storage class tell to the compiler about the details of a variable like storage area of variable, how to store variable, scope of variable, default value of value (if it is not initialized it) and what is the lifetime of the variable.

Types of storage class:

Storage classes mainly divided into four types:

		Keyword
1.	Automatic storage class	auto
2.	static storage class	static
3.	external storage class	extern
4.	register storage class	register

Type	STORAGE AREA	SCOPE	LIFE	DEFAULT VALUE
auto	CPU MEMORY	BODY	within the function	Garbage value
static	CPU MEMORY	FUNCTION	program	ZERO (0)
extern	CPU MEMORY	PROGRAM	Till the end of the main program.	ZERO (0)
register	REGISTER MEMORY	BODY	within the Function	Garbage Value

(ANURAG GUPTA)

ALGORITHM AND FLOWCHART

① Write an algorithm and flowchart to add two numbers entered by user.

Step 1 : Start

Step 2 : Initialize varriables n_1, n_2 and sum

Step 3 : Read values n_1 and n_2

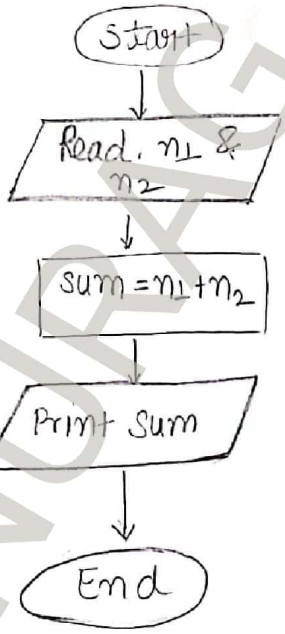
Step 4 : Add n_1 and n_2

$sum = n_1 + n_2$

Step 5 : Print sum

Step 6 : Stop

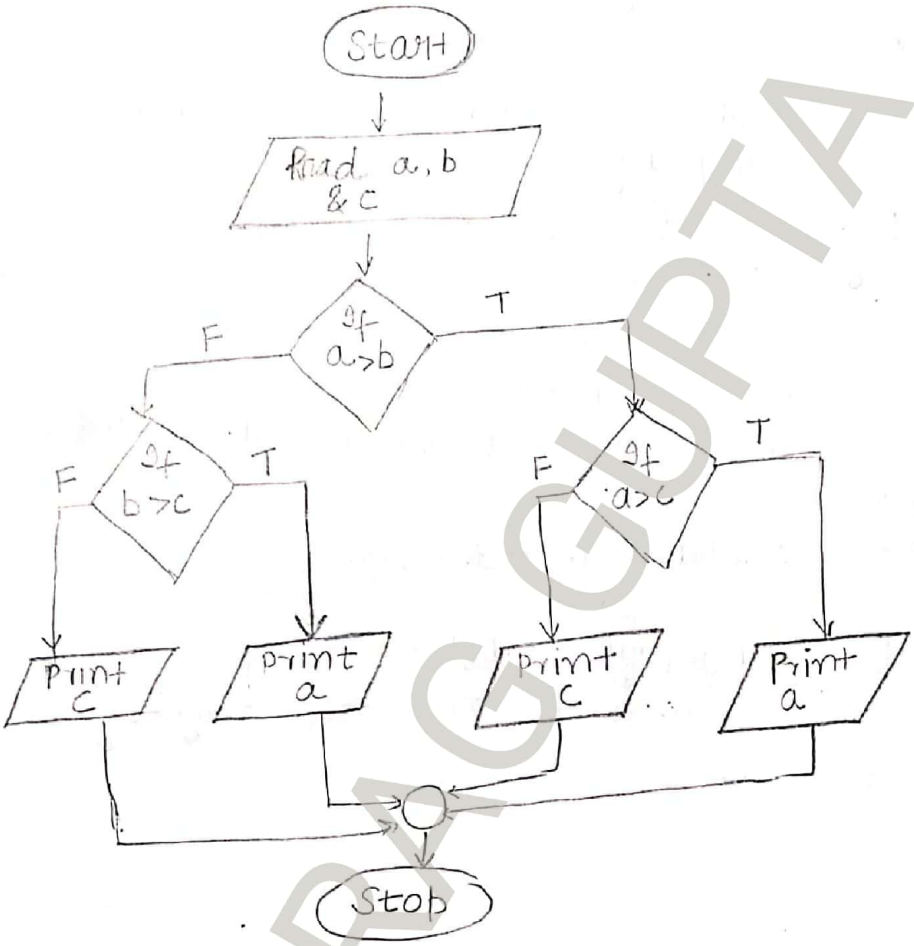
flowchart :



(2) Write an algorithm and flowchart to find the largest among three different number.

Step 1 : Start
Step 2 : Enter three number a, b & c
Step 3 : Read variables a, b & c
Step 4 : If a>b then goto step (5), otherwise goto step (8)
Step 5 : If a>c then goto step (6), otherwise goto step (7)
Step 6 : Print a is the largest, goto step (11)
step 7: print c is the largest, goto step (11)
Step 8 : if b>c, goto step (9), otherwise goto step (10)
Step 9 : Print b is the largest, goto step (11)
Step 10 : print c is the largest
Step 11 : Stop

Flowchart :-



(3) Write an algo and flowchart to find all roots of a quadratic equation $ax^2+bx+c=0$

Step 1: Start

Step 2: Read variables a, b, c.

Step 3: calculate $D = b^2-4ac$

Step 4: If $D>=0$ then goto step (5), otherwise goto step (6)

Step 5: $r_1 = (-b+\sqrt{D})/2a$

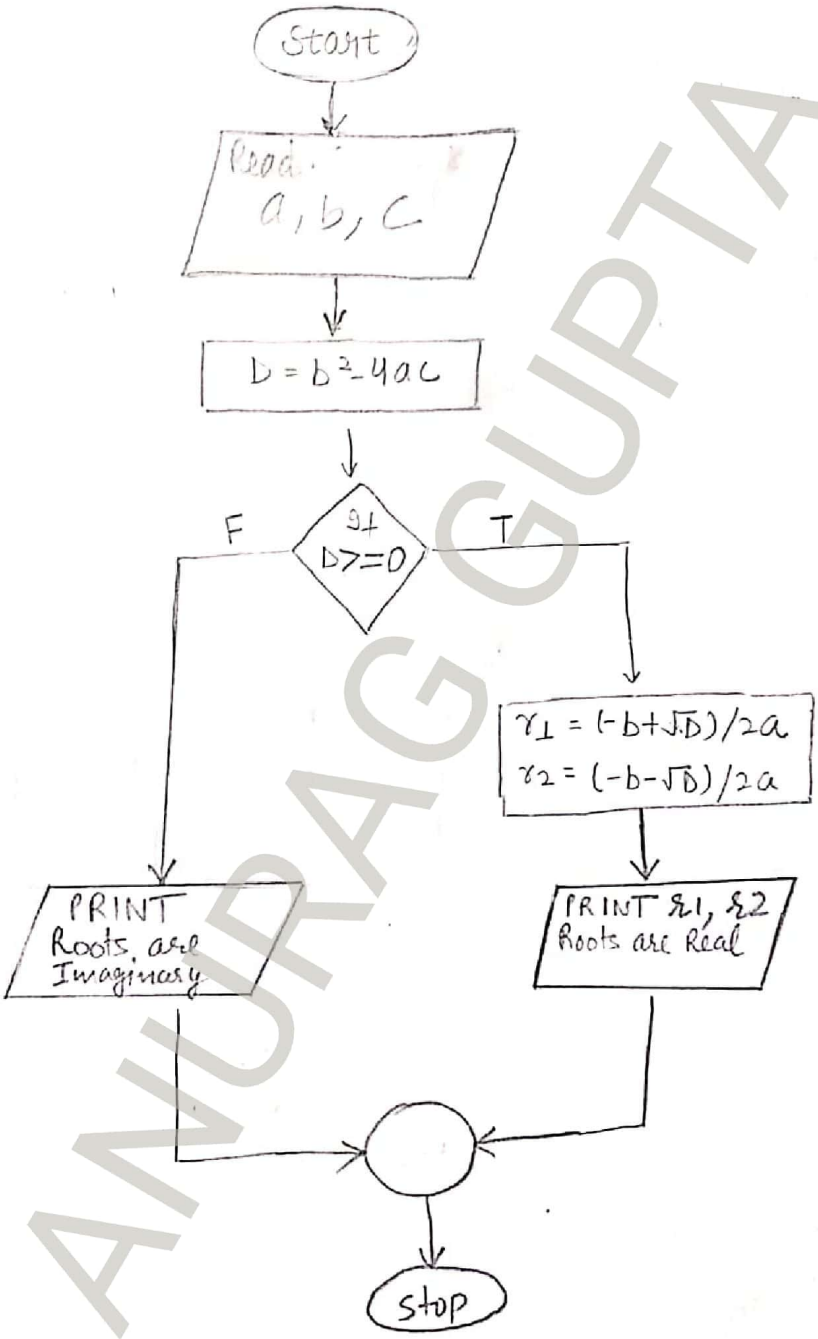
$r_2 = (-b-\sqrt{D})/2a$

print r_1 and r_2 and roots are real, goto step (7)

step 6: print roots are imaginary.

step 7: Stop.

flowchart



(4) Write an algorithm and flowchart to swap two numbers.

Step 1 : Start

Step 2 : Read a, b

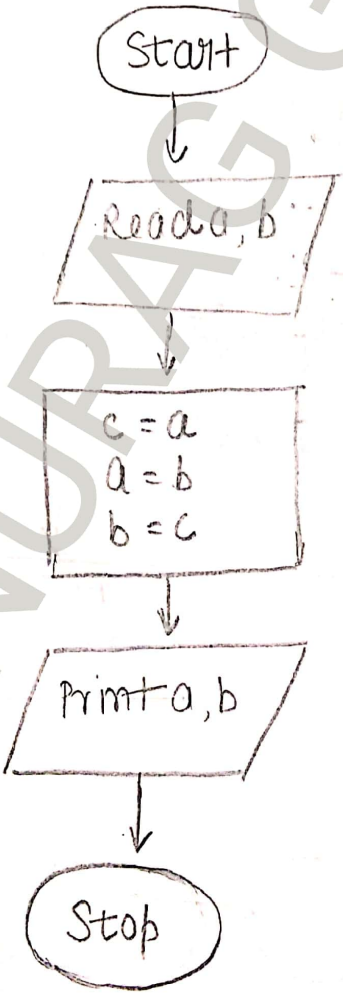
Step 3 : c = a

Step 4 : a = b

Step 5 : b = c

Step 6 : Print a, b

Step 7 : Stop



(5) Write an algorithm and flowchart to convert temperature.

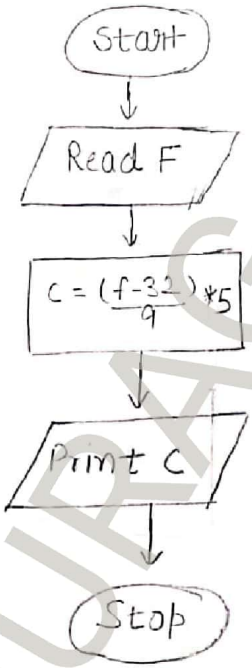
Step 1 : Start

Step 2 : Read F

Step 3 : C = ((F-32)/9 * 5)

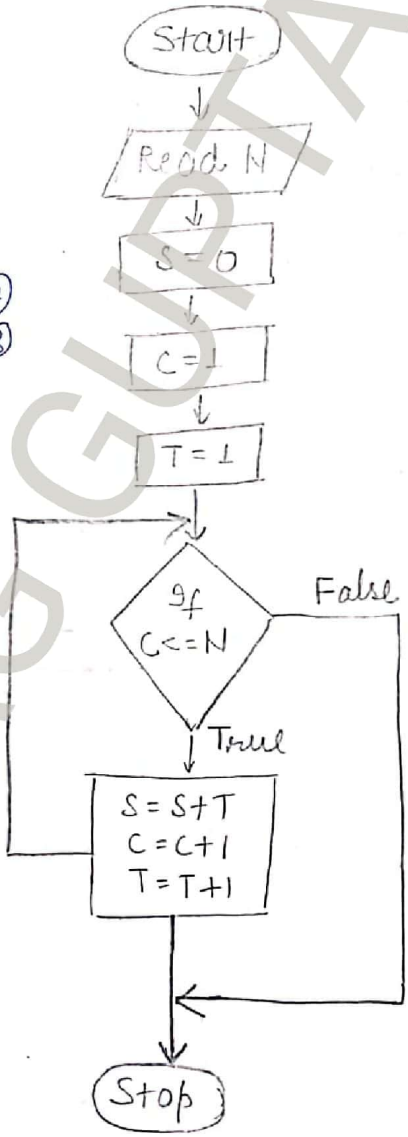
Step 4 : Print C

Step 5 : Stop



⑥ Write an algorithm and flowchart for sum of series $1+2+3+\ldots\ldots n$th.

Step 1: Start
Step 2: Read n
Step 3: S = 0
Step 4: C = 1
Step 5: T = 1
Step 6: If (C <= n) then goto step ⑦ otherwise goto step ⑧
Step 7: S = S + T
C = C + 1
T = T + 1
goto step 6
Step 8: Print S
Step 9: Stop



(7.) Write an algorithm and flowchart to find the sum of digit of given number.

Step 1 : Start

Step 2 : Read n.

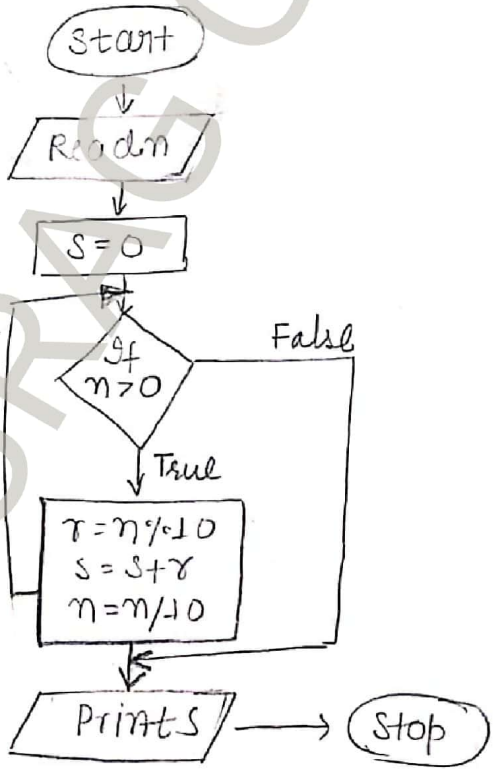
Step 3 : S = 0

Step 4 : If (n > 0) then go to step (5), otherwise goto step (6)

Step 5 : r = n % 10
S = S + r
n = n / 10
go to step 4

Step 6 : Print S

Step 7 : Stop



⑧ Write an algorithm and flowchart to find the area of triangle.

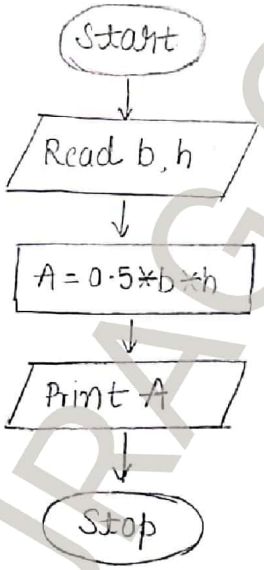
Step 1: Start

Step 2: Read b, h

Step 3: A = 0.5*b*h

Step 4: Print A

Step 5: Stop



(9) Write an algorithm and flowchart to find Armstrong number.

Step1: Start

Step2: Read num.

Step3: Assign n = num, sum = 0.

Step4: If n>=1 then goto step(5), otherwise goto step(6)

Step5: r = n%10

Sum = sum + r*r*r

n = n/10

goto step 4

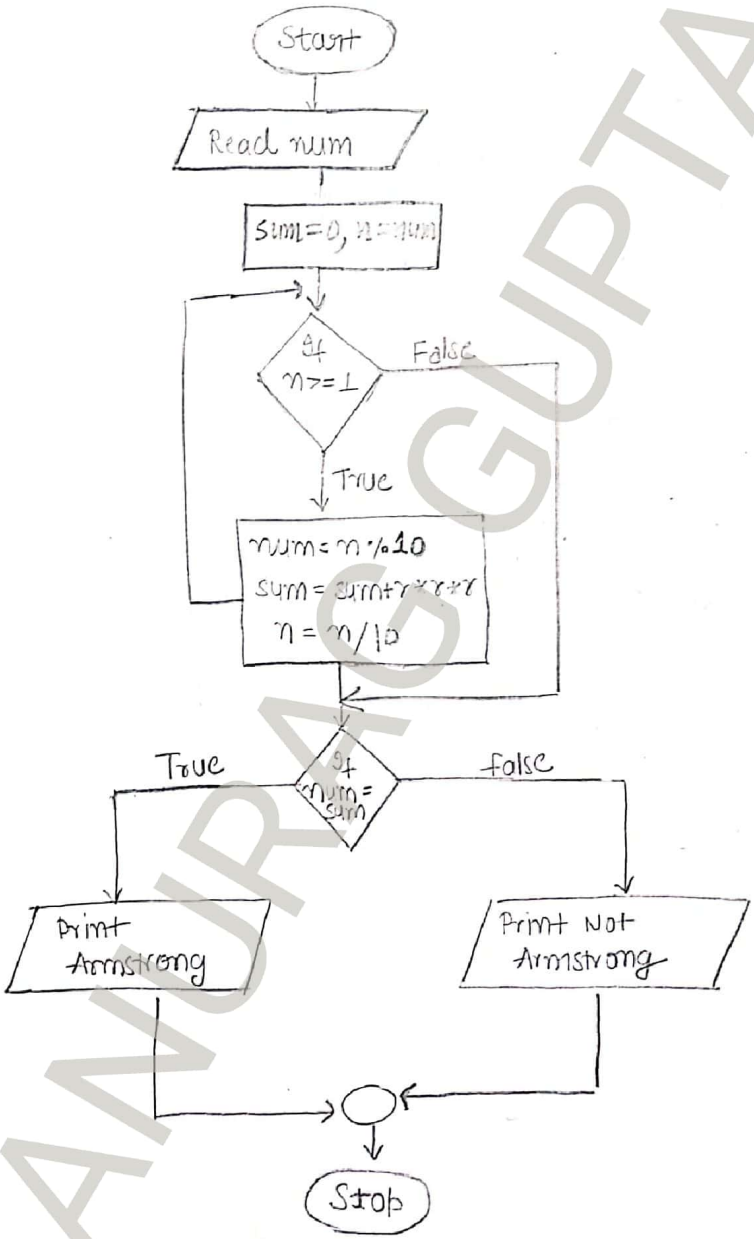
Step6: If num = sum then goto step(7), otherwise goto step(8)

Step7: Print number is armstrong, goto step(9)

Step8: print number is not armstrong

Step9: stop

Flowchart :-



(10) Write an algo and flowchart to check number is even or odd.

Step 1 : Read n

Step 2 : rem = n % 2

Step 3 if rem == 0 then goto step (4), otherwise goto step (5)

step 4: print n is even, goto step (6)

step 5 : print n is odd

Step 6 : Stop