Chapter – 1

Digital Computer:- "Digital Computer is an electronic device which accept input in binary formate, process it and produce output in binary formate."

"A Computer system is combination of hardware and software. Resources which integrate together and provide various functionalities to the user."

1. Hardware
2. Software.

1. **Hardware:-** Hardware represent the physical and tangible (effects) Componants of a Computer system i.e. the Componant that can be seen and touched.

Examples of Hardware are following.

① Moniter	④ Scanner	⑦ RAM/ROM	⑩ CPU
② Keyboard	⑤ UPS	⑧ Speaker	
③ Mouse	⑥ Harddisk	⑨ Printer	

2. **Software:-** Software is the set of the programs and instructions that are required by the hardware resources to function properly. Example of software are following

① operating system ② Complier ③ linker ④ loader etc.

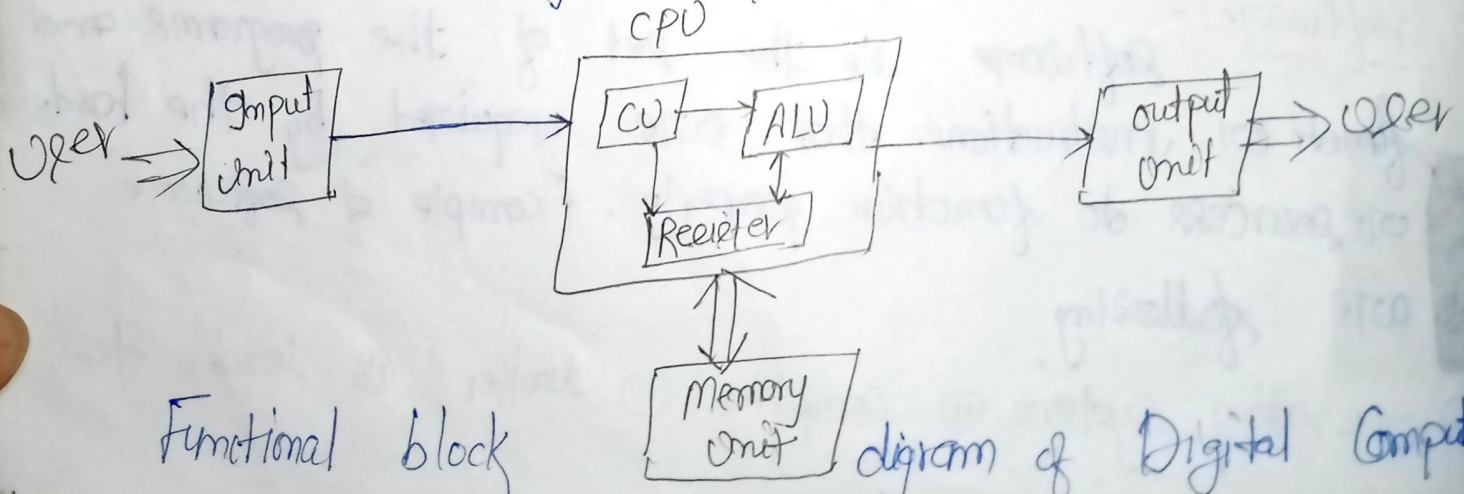
Functional Component of a Digital Computer:-

Working of digital Computer includes following three major task.

1. Data Input
2. Processing of data
3. Data output

The basic Components that help informing above mentioned tasks are called as the functional Components of the Computer. these functional Comparant as following.

1. **Input Unit:-** It takes Input via Input devicess.
2. **Central processing unit:-** It is responsible for processing of data.
3. **Output Unit:-** It produces the output and helps visualizing it.
4. **Memory Unit:-** It holds the data and Instructions during the processing.



Functional block digram of Digital Computer.

2 + 3 operands

Input Unit:- The Input consists of output devices that are attached to the computer. These devices take input and convert it into Binary format that computer understand. Some of the common input devices are Keyboard, Mouse, & Camera, light Pen etc.

CPU:- The CPU is called the brain of the computer because it is the control centre of the computer. CPU process the information provided by the input devices. CPU -

(a) First fetches the instructions from the memory And then interprets them so that it may be identify what is to be done.

(b) If required data is fetched from memory or input device.

(c) then after CPU performs the required computations

(d) After computations, stores the output on displaces it on some output device.

Components of CPU:- The CPU consists are three main Components that are as following-

① Arithematic and logic unit:- The ALU performs arithematic and logical operations. Arithematic Calculations includes addition, subtraction, Multiplication etc. logical Calculation involve relations & logical operations.

② Control Unit:- The CU is responsible for following Activities.

(a) It Coordinates and Controls the flow of data and Instructions among various unit of a Computer

(b) It Controls all the operations of ALU, memory, Registers and input/output unit.

(c) It takes care of executing all the Instructions stored in a program by fitching the Instructions decoding the Instruction and sending the appropriate Control signals to other functional unit. Until the required operation is Completed.

③ Register:- A Register can hold the data, instruction and address of memory location which is to be directly used by the processor. Registers can be a different sizes (16bit, 32bit, 64bit and soon).

There are two major types of Registers. —

① General perpose ~~data~~ user Register :-

① User / General purpose Register :-
→ It is used to store operands & Intermidiate result etc. during assemb processing.

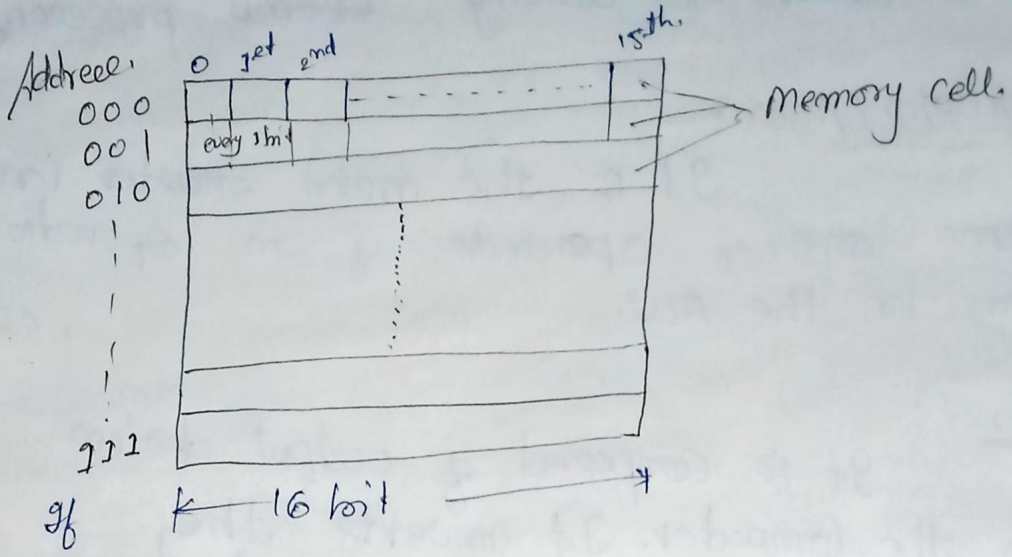
② Accumulator (ACC) :-
It is the main register in CPU and contains one of operands of an operation. to be perform in the ALU.

Binary operator
+
−
×
÷
=

③ Output Unit :-
It is composed of output devices attached to the Computer. It converts the binary output data coming from CPU to human understandable form. Some examples are Monitor, printer, speaker etc.

④ Main memory unit :-
This memory unit stores data and Instructions. It is also called as Primary Memory, Internal memory, and/or Random Access memory. Whenever a program is executed, its data is copied to this memory and is stored in the memory till the end of the execution. This memory is devided into many stores locations also called as Memory cells as shown blow in the figure. Each location is of equal size and can store data or instructions or address.

also each location has an address associated with it so that Computer can read/write any memory location easy



Storage devices!- storage devices is an hardware unit which is capable of holding informations. Either permanently or Temporarily.

There are two types of storage devices use with Computer system.

① Primary storage device/Primary memory/such as RAM and ROM.

② Secondary storage device/Secondary memory/such as Hard disk CD, DVD etc.

① Primary memory!- Primary memory is most essential element of a computer system because without it Computer can not perform even simpler task. Primary memory is the memory which is directaly Accessed By the CPU During program execution.

The programs and data that the CPU Required during execution of the program are stored in the primary memory. Primary memory is of two basic types given as Below:-

(i) Volatile Memory:- The data in Volatile memory vanished when ever the power supply to it goes off. RAM is an Example of Volatile memory.

(ii) Non-Volatile memory:- The data in non-volatile memory is not vanished even when power supply to it goes off. ROM is an example of non-Volatile memory.

* RAM:-
  (i) It is also called as Read/write memory
  (ii) The program and data that are recieved by the CPU during the execution of a program are stored in a RAM.
  (iii) It is a volatile memory so the data is lossed when the ~~power~~ power is turned off.

* ROM:-
  (i) It is also called as Read only memory.
  (ii) It's store crucial information essential to ~~boot~~ operate the Computer system like the program essential to boot the Computer.
  (iii) since it is non-volatile memory. It Retains the data even when the power is turned off

* static RAM that retain data bits in its memory as long as power is being supplied.

* NVRAM a type of memory that retain its contents when Power is turn off. one type of NVRAM is SRAM that is made non Volatile by Connection it to a Constant power source such as battery)

storage devices

Primary storage memory — Volatile (eg - RAM), Non-Volatile (eg-ROM)

Secondary storage memory

DRAM:- Dynamic Random access is typically used for data or program code that Computer processer needs to function.

RAM - DRAM, SRAM (static), NVRAM (Non Volatile)

1. Temporary storage
2. use for operation
3. Volatile
4. Read and write
5. RAM usally faster

ROM - PROM, EPROM, EEROM (electrically erasable)

1. Permanent
2. Basic startup process
3. Non-Volatile
4. only Read
5. Relatively slower.

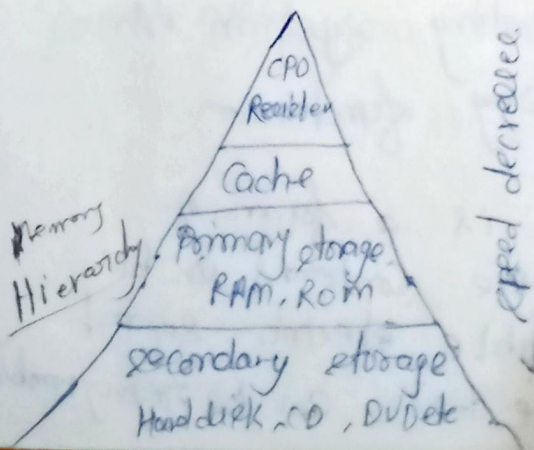
② secondary memory:- Secondary memory is a Computer memory that its Non-Volatile in nature and is not directaly Access by the processor (CPU). Data in secondary memory must be Copied into primary storage RAM before it's use.

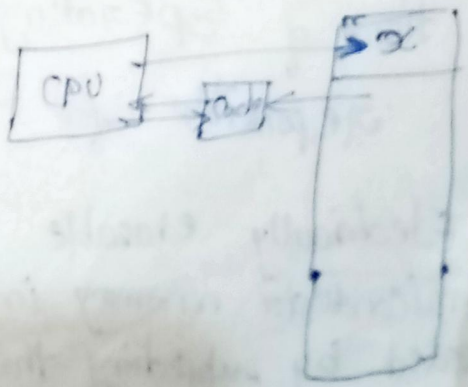
secondary memory are slower and cheaper form of Memory as Compaired to primary memory. Move over secondary memories are usually larger in Capacity as Compare to Primary Memories

some example of storage devices are Hard disk CD, DVD, Magnetic tape etc.

* **Memory Hierarchy!-** Memory Hierarchy is an arrangement of various types of memories is associated with the computer system Based on speed, capacity and cost in increasing / decreasing order.

* **Cache Memory!-** Cache memory lies b/w RAM and ROM. It used to store recently used data and instructions by CPU. whenever any data is required by the CPU, it is first looked into cache memory and if available it is used by the CPU. If data is not avialable in Cache it is looked into main memory and fetched from there since cache memory is faster than RAM so idea is that if frequently used data resides in cache it will, Improve the over all processing of the computer system.





Software

- System Software
  - (i) Operating system
  - (ii) Compiler
  - (iii) Interpreter
  - (iv) Linker
  - (v) Loader
- Application Software
  - (1) Office Package
  - (2) Browser
  - (3) Media player
  - (4) DBMS (Data Base Management system)

(1) Operating System:- (i) Operating system is a system software that acts as an interface b/w Hardware and user.

(ii) It manages system resources like memory, CPU, input, output etc.

(iii) It provides a platform on which other application softwares are install and executed properly.

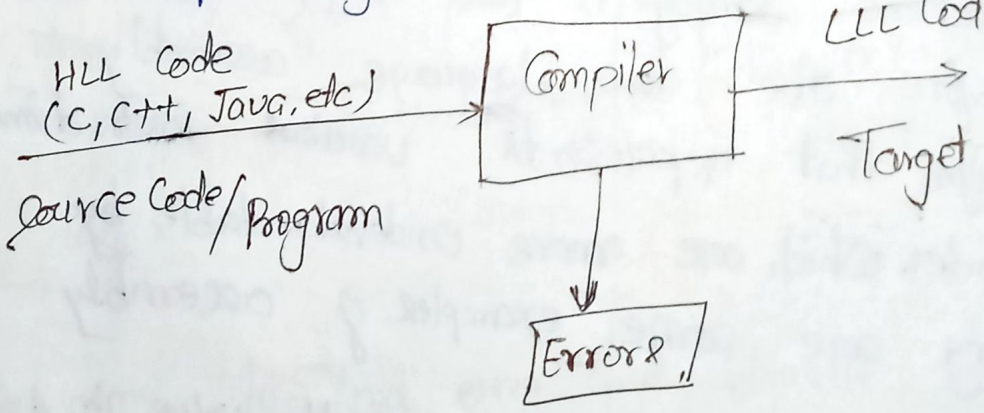
(iv) Some examples of operating system:- DOS, Windows, Linux, etc.

Functions of operating system:- Operating system is responsible for providing following function -

* Electrically erasable read only memory is a form of semiconductor memory in which the entire content is can be erased by subjecting the device to suitable electric signal. After erasing, the device can be reprogramb

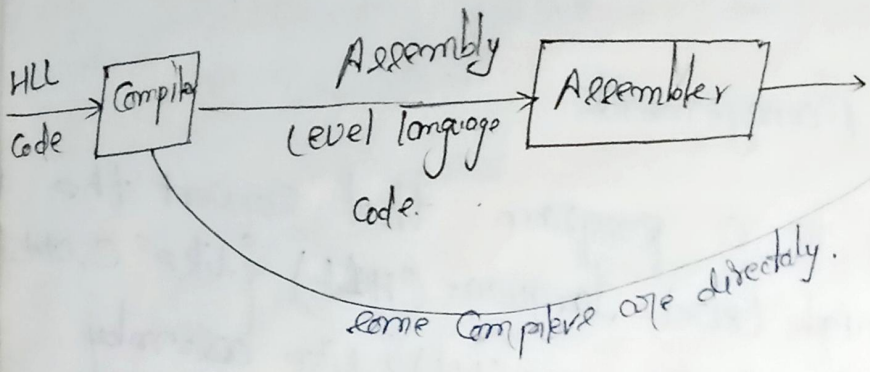
(i) Process Management
(ii) Memory Management
(iii) File Management
(iv) I/O " (Input/output)
(v) Network "
(vi) Security & Protection Management

(2) Compiler:- Compiler is a program that convert the codes written in high level language (HLL) {like C, C++, Java, etc} in to low level language (LLL) {like assembly language and machine language}. Machine language is the Binary code which is understandable by the Computer machine. Some Ex. of Compiler for C and C++ language are, portable C, Turbo C/C++, GCC etc. also, during this Conversion process compiler identify and report any error in the source program.



Assembly language - add, mul, ... (Code language for user)

Assembler :- Assembler is a program that convert the assembly language code into machine language code. Ex. Turbo Assem., GNU Asm., Microsoft Asm. etc.



Assembly level language :- Assembly language is an extremely basic form of programming. It is a low language level programming language. Each personal computer has a microprocessor (CPU) that manages the computer's Arithmetic, logical and control activities. A processor understand only machine language instructions which are strings of 1s and 0s. However, machine language is too complex for use by humans. Therefore the low language assembly language is design that represents various instructions in symbolic codes which are more understandable by humans. following are some examples of assembly level language.

INC A - Increment the value in A
DEC B - Decrement " " " B
MOV C, 5 - Store the value 5 into register C.

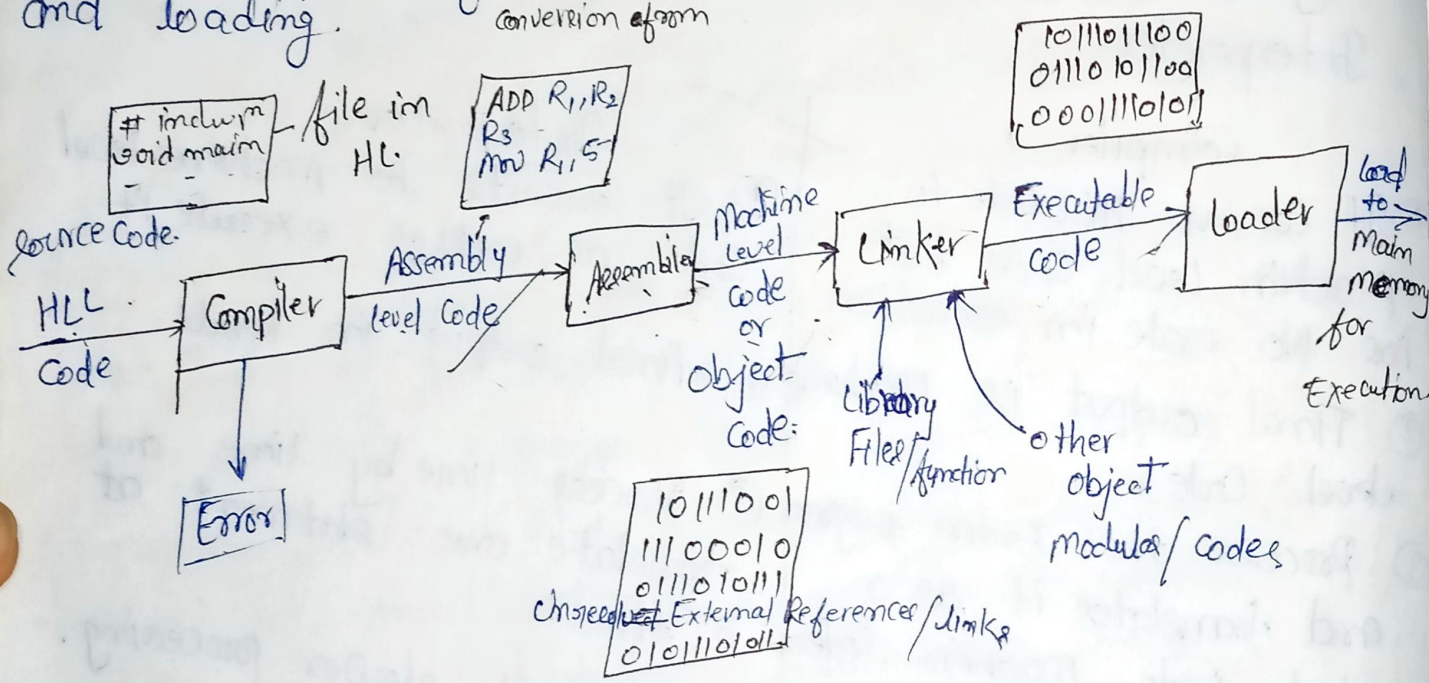
⑧ Interpreter :- An Interpreter is a Computer program which translate and execute the statement (written in high level language) line by line. It continuous translating the program until the first error is made. As it gets first error it stops. For ex, languages such as Python, Lisp, Ruby, etc. use Interpreter.

Compiler	Interpreter
① It convert HLL Code to Machin Level Code but has No role in execution	① It Converts to machine level Code as well as execute it.
② Final output is machine level Code.	② Final output in Result.
③ Process the Entire program and translates it as a whole into machine Code.	③ Process line by line and translate one statment at a time.
④ Relatively faster processing	④ Relatively slower processing.
⑤ Debuging is Comparatively difficult	⑤ Debugging Comparatively easy.

Debuging - To find error and slove it.

Linker and Loader:-

Linker and loader are the system utility programs that plays a major role in the execution of a program. The source code of a program passes through Compiler, Assembler, Linker, loader in the respective order, before the execution. Following is the Execution sequence of the source code into Executable code and loading.



Linker:-

the assembler/or Compiler generates the object code of a source program and hands it over to the linker. The linker takes this object code and generates the executable code for the program and hands over to the loader. Linking is perform as the last step after Compiling of the program.

Linker is responsible for linking of library functions and object module.

Loader:-

As the program that has to be executed must reside in the main memory of the Computer, It is the responsibility of the loader to load the executable file into the main memory for execution. It allocates the memory space to the executable code in the main memory.

Source Code:-

The source code consists of the Programing statement that are written by a programmer and save in a file. For example- A Programmer writting the a program in the C-language, types desired sequence of C-language instructions using a text editor and then saves it as a file name (myfile.c) this file is said to Contain the source code. It is now ready to be Compile by some Compiler.

Object Code:- When the source code file is compile, it is translated into binary machine language code. file which is called as object code file/object file/ object module. the object code file Contains.

a sequence of instructions that processor (CPU) can understand. these object file contain unresolved external references. that are required to be link against other object file or libraries.

Executable Code:- The object modules/files are not yet ready for execution. the object file is link with other object files and/or libraries. this linking is done by the linker. after linking, the linker produces an executable code which is ready for execution by the processor. Now, this executable file may be loaded into main memory by the loader.

Algorithm

An algorithm is sequence of finite and well defined instructions for completing a task or solving a problem. An algorithm is given an initial state, Proceed through a well defined series of successive states, Eventually terminating in an end state.

Properties of Algorithm :- An algorithm must have following properties.

1. Finiteness
2. Definiteness
3. Input
4. Output
5. Effectiveness.

1. **Finiteness :-** An algorithm must always terminate after a finite no of steps.
2. **Definiteness :-** Each step of an algorithm must be precisely define and must be unambiguous.
3. **Input :-** An Algorithm has 0 or more input
4. **output :-** An Algorithm has 1 or more output

⑤ Effectiveness:-

It means that all of the operation of an algorithm must be so basic that it can be executed in a finite length of time. In an effective manner.

Representation of Algorithm:- An algorithm can be represented for written in any of the following form.

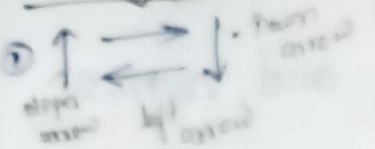
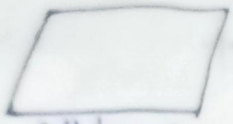
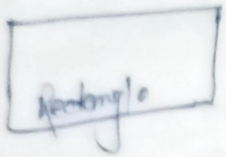
① Natural language:- It is the informal representation of an algorithm where steps of Algorithm are written in any natural language. such as- Hindi, english etc.

Pseudo Code:- This representation is a formal representation of algorithm in which the step of Algorithm follows some formal rules and notations to be written. This representation some what similar to programming language.

Flow chart:-

Flow chart is a diamagnetic representation of an algorithm which is used to depict sequence finite step for Completing a task or solving a problem.

A flow chart uses different symbols to represent various flow and steps of algorithm for eg. example following is the list of such symbol.

symbol	Name	Description or use
① 	Flow line	To indicate flow of the logic by connecting other symbols.
② 	Terminal (start/stop)	Use to represent start and end of the flow chart
③ 	Input/output	It is used to represent Input and output operation.
④ 	Processing	used to denote mathematical operation and data manipulation
⑤ 	Decision Box	Used to represent the conditional operation in which a decision has to be taken in based on to alternates: True or false.

Algorithm (1) :- write an algorithm to Calculate the Additional of two Input numbers.

Natural Language :-

Step-1 :- start

Step-2 :- Read two value in 'num1 and num2

Step-3 :- Take a variable 'sum'.

Step4 :- sum = num1 + num2 } - Processing

Step-5 :- Display the value of 'sum' } output

Step-6 :- End

Pseudo Code :-

Input

Add (num1, num2)

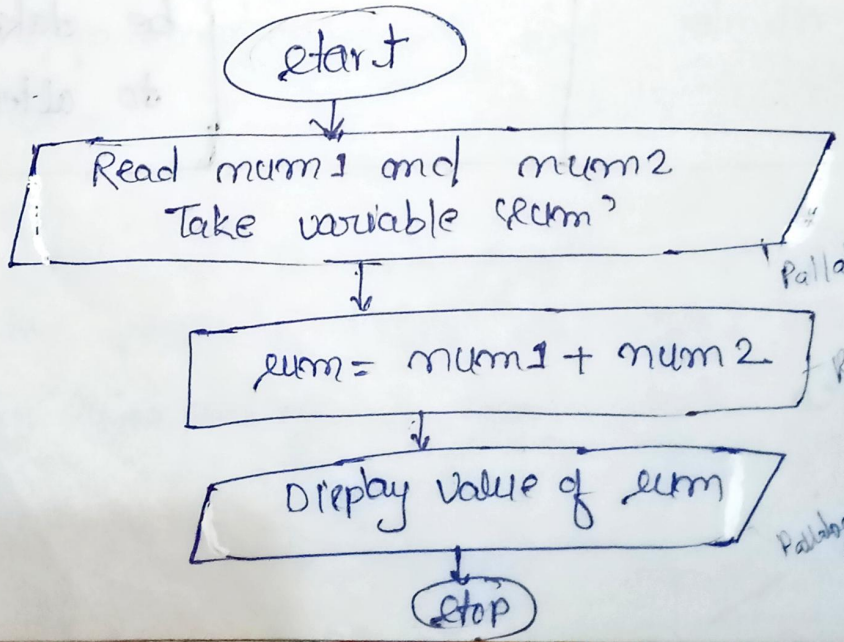
(1) take variable 'sum'

(2) sum ← num1 + num2

(3) Print (Sum)

(4) End

Flow chart :-



Algorithm②:- Write an algorithm to check the whether an Input no 'n' is even or odd.

Step 1:- Start

Step 2:- Read a numbe 'N'

Step 3:- Calculate N/2 and store remainder in variable 'r'

Step 4:- check the value of 'r' If it is zero then display N is even number otherwise odd no.

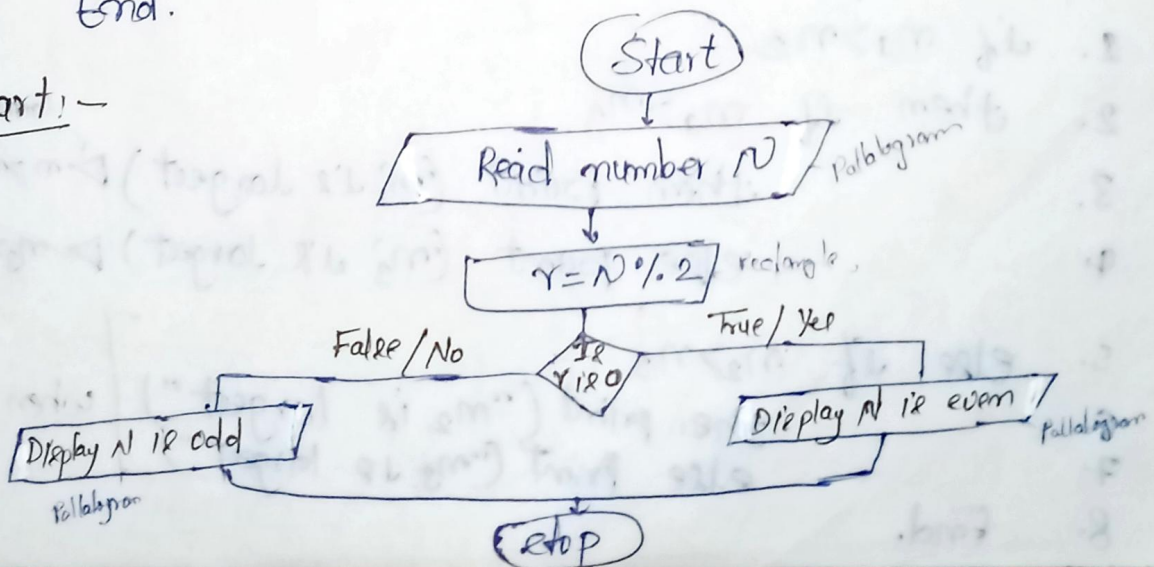
Step 5:- End

Pseudo code:-

Check_Even_odd (N)

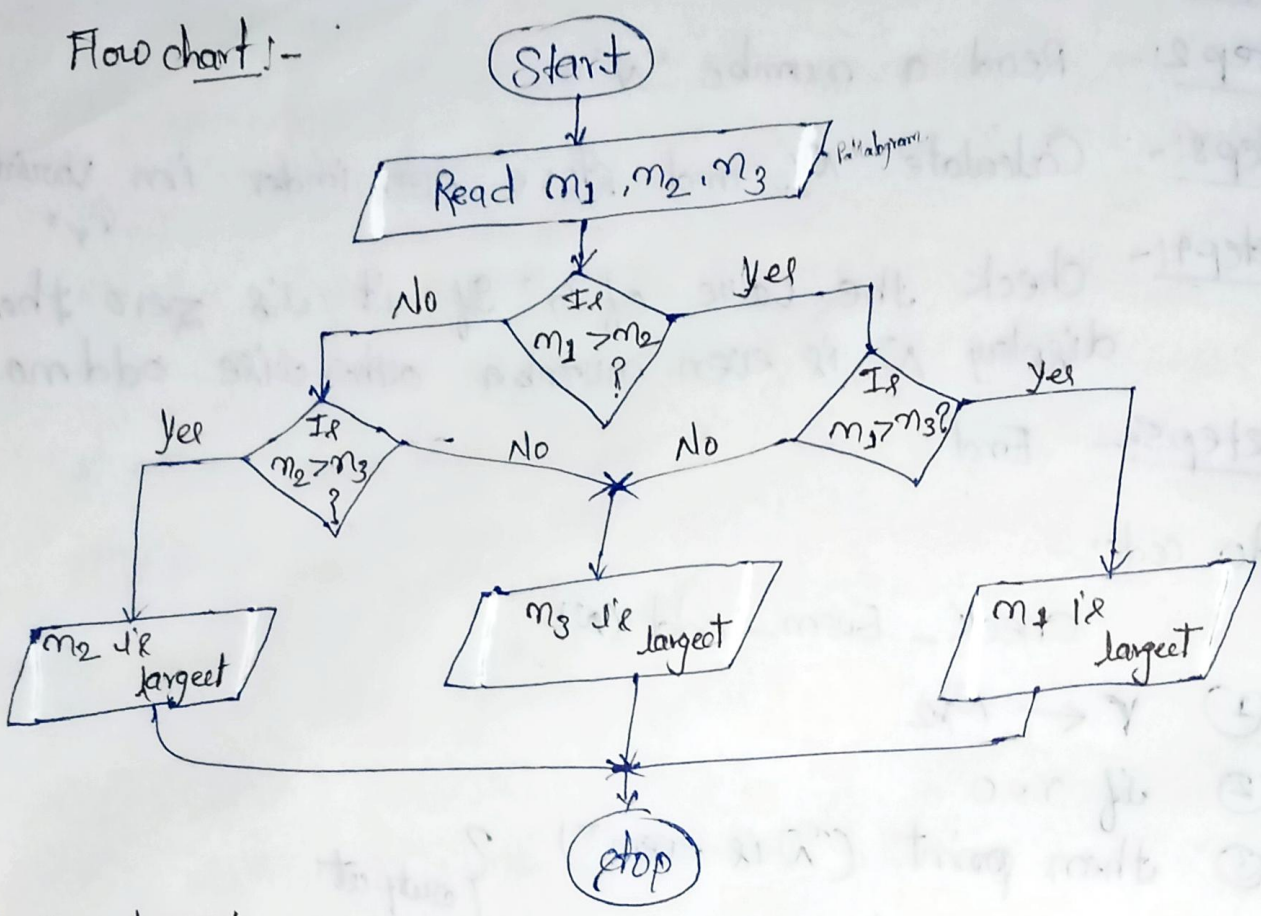
① r ← N%2
② if r=0
③ then print ("N is even")
④ else print ("N is odd") } output
⑤ End.

Flowchart:-



Algorithm 3:- Write an algorithm that finds the largest number from among three distinct input numbers.

Flow chart :-



Pseudo code:-

largest_three (n_1, n_2, n_3)

1. if $n_1 > n_2$.
2. then if. $n_1 > n_3$
3. then print (n_1 is largest) ▷ $n_1 > n_3$
4. else print (n_3 is largest) ▷ $n_3 > n_1$

Comment: $n_1 > n_2$

5. else if $n_2 > n_3$
6. then print ("n_2 is largest")
7. else Print ("n_3 is largest")

when $n_2 > n_1$

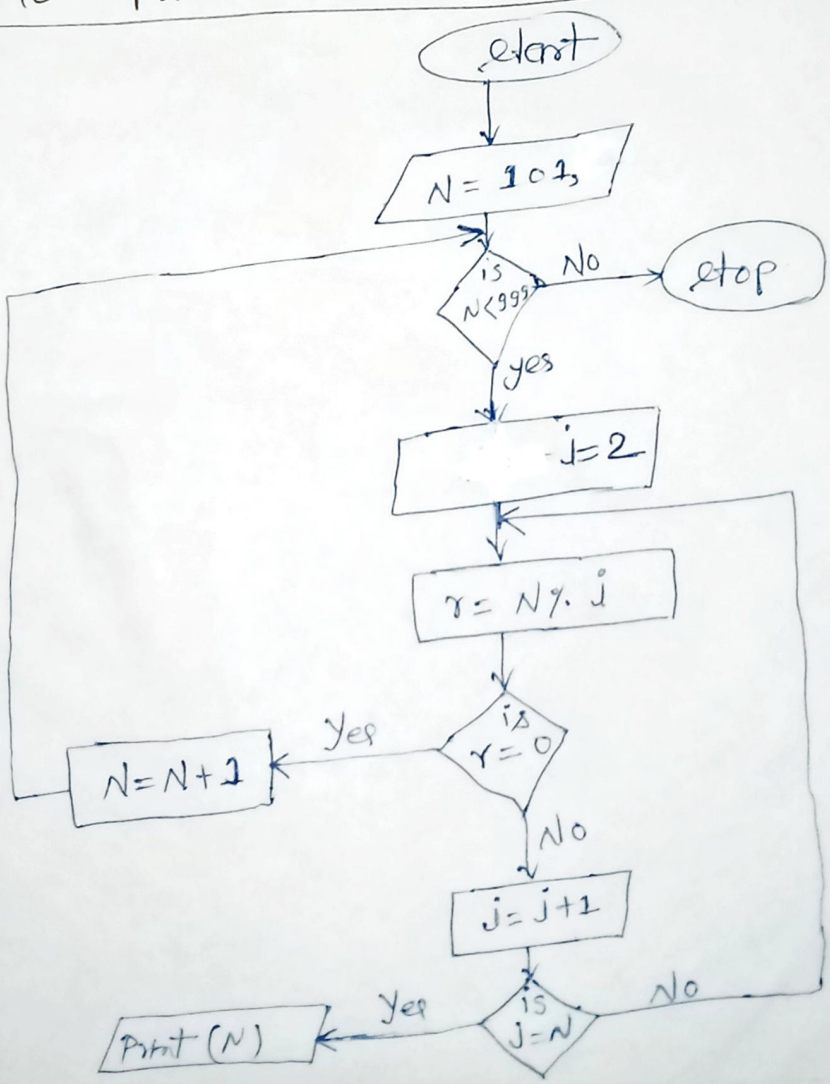
8. End.

Pseudo Code for Prime No:-

check Prime (n)

1. j ← 2
2. for j ← 2 to n-1
3. if (n % j == 0)
4. then break.
5. if (n == j)
6. then Print (n is prime)
7. else Print (non prime)
8. end

Flow chart to Print Prime No. b/w 101 to 999.



Pseudo code for factorial :-

facto (n)

1. $i \leftarrow 1$, $f \leftarrow 1$
2. for $j \leftarrow 1$ to n
3. do $f = f * j$
4. Print (f)
5. End

-: C-Program :-

Introduction

C-Tokens:-

'C-tokens' are the basic building block of the c-programming language. All the constructs of a c-program are formed using this c-tokens. Thus we can say that the Entire c-program is made up of c-tokens. There are following 6 c-tokens in any c-program.

1. Identifiers
2. Keywords
3. Constant
4. strings
5. operators
6. special symbols.

1. Ide.
2. Keywords:- Keywords are the words in c-programming language that have fixed meaning and use. It can not be used for defining any indentifier and the meaning and use can not be changed. For ex. for, if, else, break, while, goto, continue, do-while, etc.

int, float, void

① Identifiers:- Identifiers are user defined names given to some variable, array, functions etc. Identifiers follow following properties—

(1) Made up of lower case Alphabates {a...z}, upper case Alphabates {A, B...Z}, digits {0, 1...9}, underscore {_}.

(2) Must start with an alphabate or underscore

Ex. abbas ✓
abbas_1 ✓
1 abbas (X)
_abbas ✓
Abbas ✓

(3) Identifiers are case sensitive.

{ abb, Abb, aBb, abB } All are different identifiers.

② Constant:- Constants represent fixed value that may be assigned to some variable. Value of a Constant can not be changed during the programming. Constant are also called literals. Constants can be of different data types. and based on there types. Constants may be of following types or category.

1. Numeric
2. character Constant

Constant or literals
- Numeric
  - Integer :- 1, 12, 105, -12, -13
  - Real :- 1.2, 0.5, 12.63
- character Constant
  - e.g. – 'c' 'C', 'a', 'A'
  - '1', '2', '$', 'J'

Any single charcter of a c - program written b/w single codes is called a character Constant

ASCII Value :- Value of characters

④ strings – strings are sequence of charater's written in b/w pair of double code (" ")

ex. "A", "Aa", "12", "A12b", "Ab$C1", "12.45" ← character

⑤ operators :- +, −, *, %, [illegible], ||

&& — impartant

⑥ special symbols :-

$, <, >, ( ), [ ], etc.

& - address of operator

(;) - semicolumn.

C - Program Structure :-

File Inclusion → It includes other library functions into your program.

Global Declaration → It declares variables or functions etc. globally.

turbo C otherwise start from main

```
void main( )
{
    Local Declarations
    Input processing
    Rest of Body
    output
}
```

opening Braces

closing Braces

→ It contains all the processing statements inside it

→ The execution of the program starts from the main function.

User defined function

→ It has two sub-sections
- → local declarations, Declaring local variables, functions.
- → Rest of processing statement

→ If we use functional programming then we defined other functions.

Input and output

for C

scanf

Printf

```
# Include <stdio.h>
# include<conio.h>
```