

## Components of Computer

(7)

### Software

- System software (OS or)
- Application software.

#### Stand alone

MS office

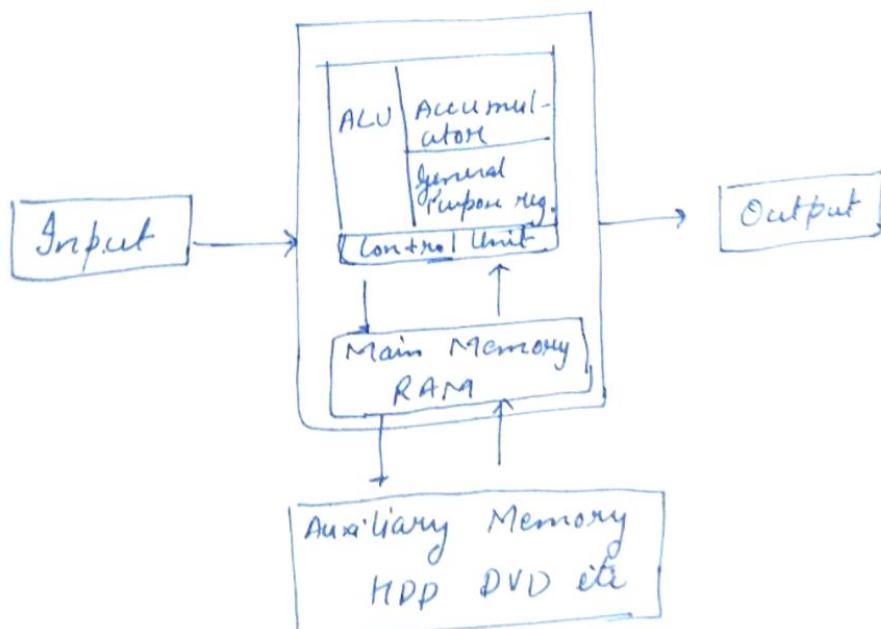
\* Can be used  
in one computer  
in which it's  
installed

#### Web application

Can be accessed in  
any PC.

### Hardware

## B Block Diagram:-



8

## Operating Systems :-

An operating system is a software that act as an interface b/w computer hardware and user. It helps us to communicate with the computer without knowing the computer language (Binary Lang).

### History of Operating Systems:

First operating system was developed in 1950 to access magnetic tape. In mid 1960 technology improved and OS was able to access magnetic disk.

In late 1960 unix was developed.

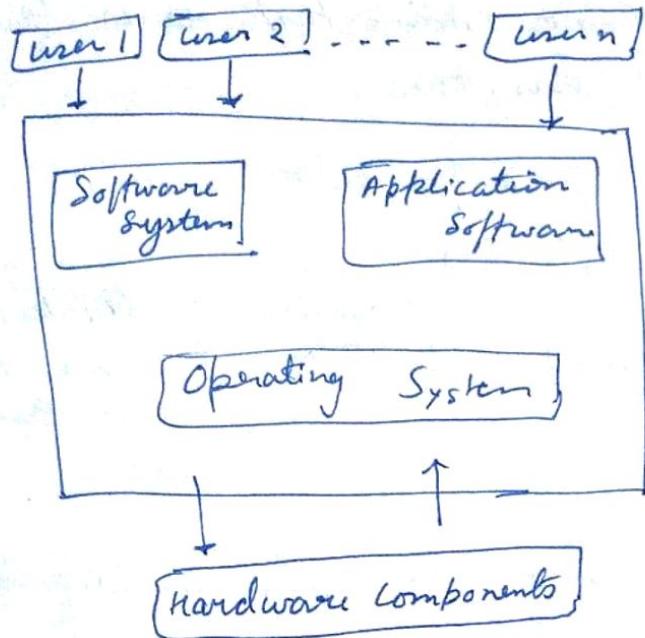
In 1981 microsoft developed DOS. OS-DOS was based on command line interface.

In 1985, ~~microsoft~~ developed GUI based O.S.

Then MS was added to DOS to be renamed as MS DOS. The first GUI.

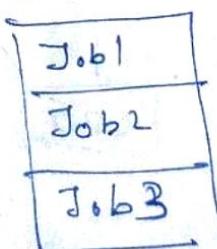
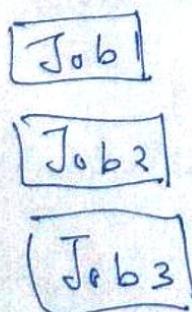
## OS Block Diagram:

⑨



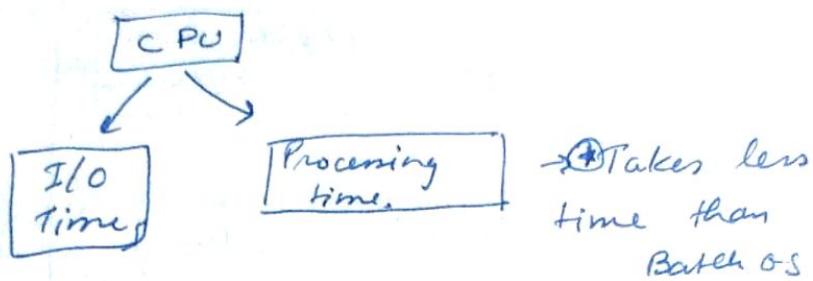
### Types of Operating System.

- ① Batch OS. - In Batch OS the sequence of a job is a program on a computer without manual intervention.  
To speed up the process the job with ~~the~~ similar type needs to batch together and run as a group.



## ② Multi programming OS.

It is an extended version of batch OS where main task is to busy C.P.U. CPU has two

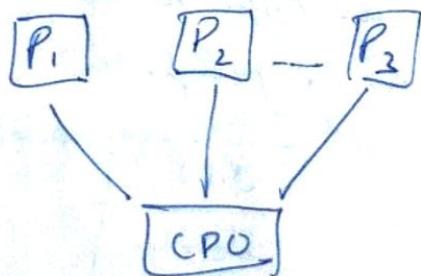


## ③ Multi processing OS. :- In multi processing O.S. parallel computing is achieved.

★ There are more than one processor present in a computer which can execute more than one process at the same time.

★ It will increase throughput.

↳ No. of tasks executing per unit time.



(11)

#### ④ Multi tasking OS.

Multi-tasking is logical extension of multi-programmings. ① It enables to execute multiple programs simultaneously.  
 ② It allows user to perform more than one task at same time.

#### ⑤ Time sharing OS.

① It enables people at different terminals to use single system at the same time.  
 ② It allows even many users to share computer resources.  
 ③ We can achieve maximum utilisation of comp. resources.

#### ⑥ Real time OS.

① In this OS each job carries a certain deadline within which the job is supposed to be completed.

Ex - (Army / space research).

#### ⑦ Distributed OS:-

① In this OS the resources are not installed at single systems. It is divided in parts and each part is installed at different machine.

(12)

\* As the processor located at different machines, it provides very fast computation.

### The functionality of OS:-

#1) Process management: It helps to create and delete the processes. It also performs synchronization and communication among the process.

#2) Memory management: It performs the task of allocation and deallocation of memory.

#3) File management: It manages all the file related activities such as organisation, storage, retrieval, running, sharing and protection of file.

#4) Device management:

\* It tracks all the devices and this tracking is done by I/O controller.

\* It also performs the task of allocation and deallocation of devices.

(5)

H5) Storage management:-  
\* different level of storage

management.

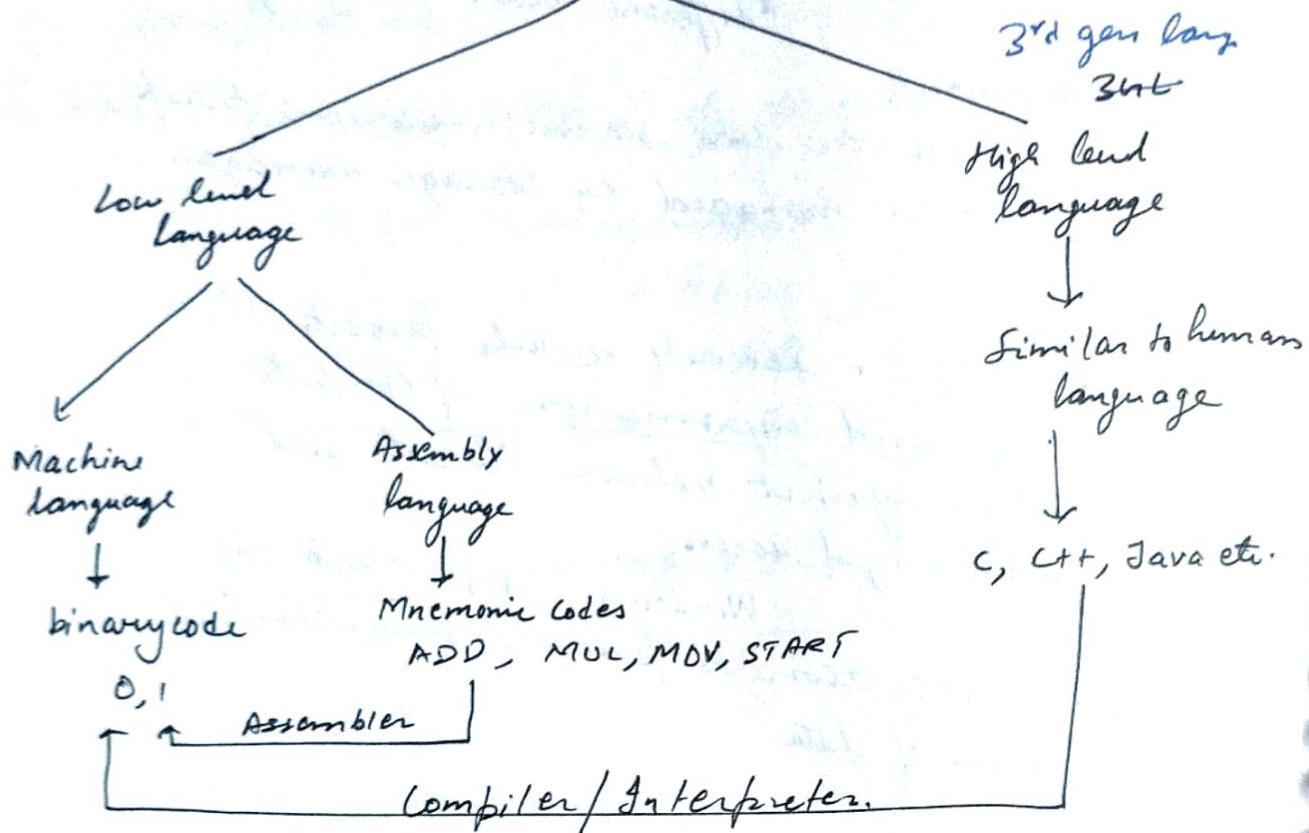
see the data stored in various track,  
that all is managed by storage manager.

H6) Security:- Security module protects the  
data and information of computer  
systems against malware threats and  
unauthorized access.

Windows uses Kerberos  
authentication to prevent unauthorized  
access of data

14

## Programming language



### ② Programming language

In order to communicate with the computer, user needs a language that should be understandable by the computer. For this purpose, different programming languages are developed to perform various tasks.

#### ⓐ Low level language.

It can only understand and execute the instructions given in the form of machine language . i.e binary code

## Machine language.

(15)

Machine language is basically the only language that computer understands. Since computer is capable of recognising electrical signals.  
0 → absence of electric pulse  
1 → presence " " " "

Therefore, it understands machine language.

\* The set of binary codes which can be recognised by a computer is known as machine code instruction set.

## Assembly language:-

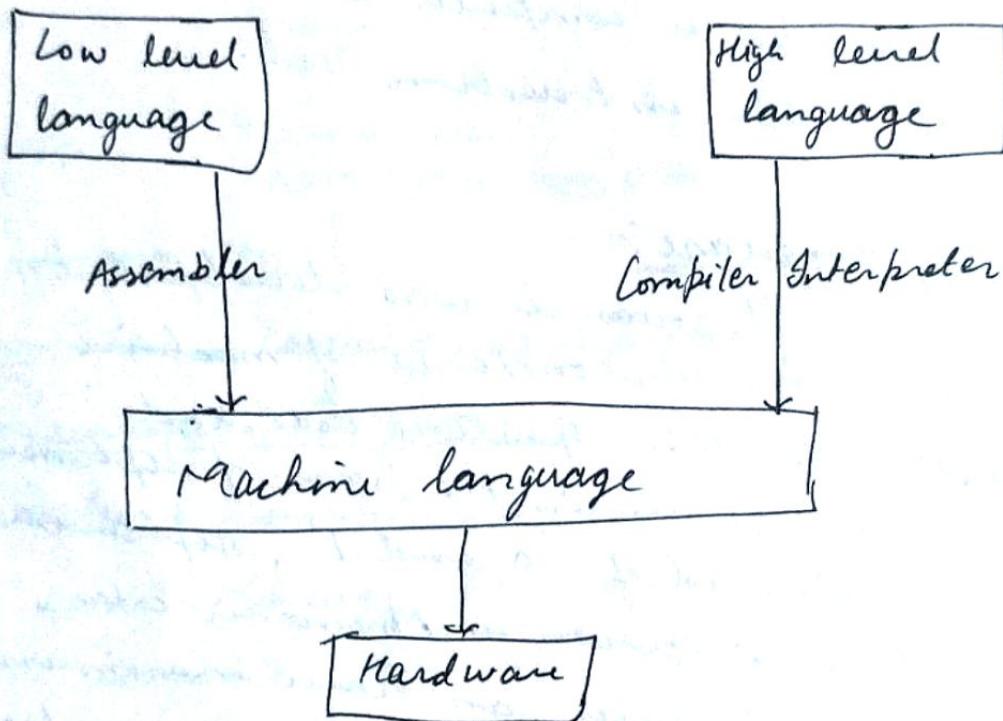
It was developed to overcome the inconvenience of machine language. In this operation code and operands are given in the form of alphanumerical symbols instead of '0' and '1'. Alphanumerical symbols are known as mnemonic codes. The mnemonic codes can have maximum 5 letters. Assembly language is also known as symbolic programming language.

## High level language:-

They are basically symbolic language that uses English words or mathematical symbol instead of mnemonic codes. Each instruction in H.L.L is translated into many machine level instructions.

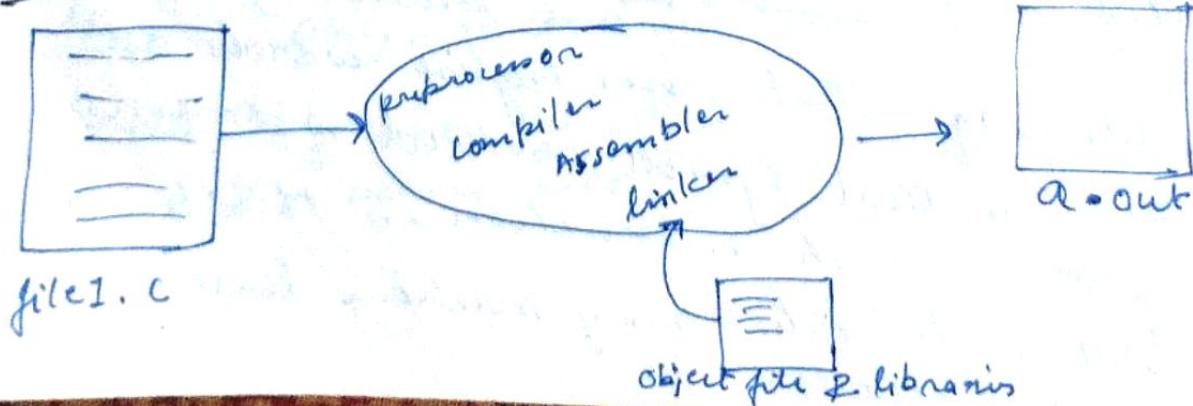
16 Thus, it shows one to many translations. The purpose of developing the high level language is to enable people to write program easily. Some H.L.L are

- \* Process oriented lang. / Problem.
- \* Procedure oriented lang.
- \* Non-procedure oriented lang.
- \* Mathematics oriented lang.



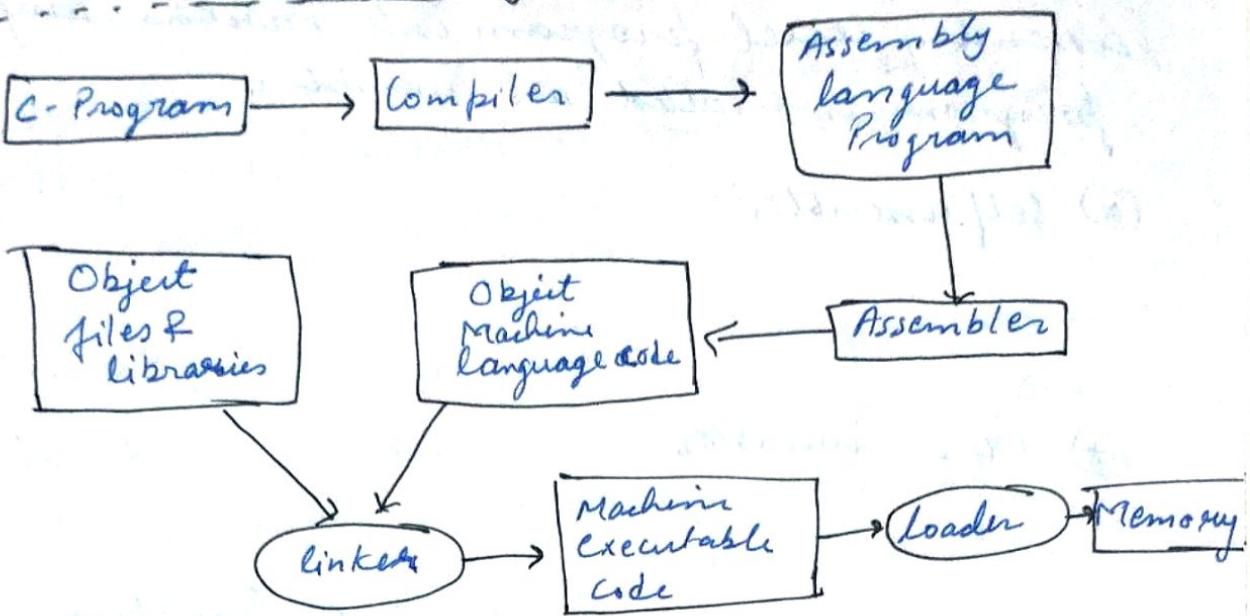
# Concept of compiler / interpreter, assembler,

linker, loader



## Translation Hierarchy

(7)



### Compiler

It is a program which translates high level language to low level lang.

Compiler is more intelligent than assembler because it checks errors, limits the range.

If a compl compiler run on a computer and produces machine code for same computer, then it is known as self compilation or resident compiler compilation.

If a compiler run on a computer and produces machine code for other computer then it is known as cross compilation.

④ Assembler : — A program which translates assembly level program into machine lang. programs is called an assembler.

⑤ Self assembler :-

⑥ Cross assembler

Assemblers are divided into two types

- 1) One pass assembler
  - 2) Two pass assembler.
- ⇒ One pass assembler :- assigns memory to the variable and translates the source code into machine code in first pass simultaneously.

⇒ Two pass assembler In this, the assembler reads the source code twice, in the first pass, it reads all the variables and assigns memory to them, in second pass it translates the source code to machine code.

(19)

# Interpreter: It translates source code into machine code. It reads one statement of program translates it and executes it immediately.

# Linker: In high level language some built in libraries and header files are stored. These predefined libraries contains basic functionality which are essential for the execution from of programs.

# Loader: It is a part of OS and is responsible for loading program in memory.

<u>Assignment</u>	<u>Compiler</u> - v/s <u>Interpreter</u>	<u>v/s - Assembler</u>
Translates HLL into machine code	Translates HLL into machine code	Translates assembly lang. into machine code.
Translates all code at one same time	Translates source one line at a time	uses processor's instruction set to convert
only needed once to create an executable file	Needed everytime you run the program	Runs quickly as conversion b/w two low level languages is just reliant on the processor's instruction set.
Will only inform the first errors it finds	Returns a list of errors found and on which line	
can compile run quickly but compiling can take a long time	Runs more slowly as it is being translated every time the code is run.	

(20)

## Introduction to Problem Solving.

### # Procedure:-

A computer cannot solve a problem on its own. One has to provide step by step by step solution of a problem to the computer.

The steps are:-

- 1) Understanding the problem
- 2) Analysing the problem
- 3) Developing the solution
- 4) Coding and implementation.

### # Algorithms:-

An algorithm can be defined as a complete, unambiguous, Finite number of logical steps for solving ~~step~~ a specific problem.

steps involved in algorithm dev

- 1) Identification of input
- 2) Identification of output.
- 3) Identification of processing operation.
- 4) Processing definiteness. (fixed valuation)
- 5) Processing finiteness. (termination condition)
- 6) Processing effectiveness. (efficiency) (space / time)

2)

Q Write an algorithm to find average of three numbers:

Step 1) read input a, b and c

Step 2) find sum of a, b & c.

Step 3) divide sum by 3.

Step 4) store the result in d.

Step 5) print the value of d.

Step 6) end

Q Write an algorithm to calculate simple interest.

Step 1) read input P, r, t and SI.

Step 2) find product of P, r, t and t

Step 3) divide product by 100

Step 4) store the result in SI

Step 5) print the value of SI

Step 6) end.

(22)

Qn<sup>m</sup> Write an algorithm to find area of A.

Step 1) read input h(height) & b(base) & A(area)

Step 2) find half product of  $0.5 \times h$  and  $\times b$ .

Step 3) divide the output by  $\times 2$ .

Step 4) store the result in A.

Step 5) print the value of A

Step 6) end.

Assignment

Qn<sup>m</sup> Write an program algorithm to find largest among all three number.

Step 1) read input a, b & c & D, D & G.

Step 2) compare the <sup>1<sup>st</sup></sup> two number

Step 3) Now ~~store~~ store the greater of first two in D

Step 4) Now compare ~~the~~ D and C take

BB

Algo (By hand). (.

(23)

Step 1) read input a, b and c

Step 2) if ( $a > b$ )

    Big = a

else Big = b

Step 3 | if ( $c > \text{Big}$ )  
    Big = c

Step 4) Print Big

Step 5) end.

On <sup>Assign.</sup> Write ~~an~~ an algorithm to find the largest from given set of numbers.

- 1) read the given set of ~~two~~ numbers as input
- 2) for each number, see if it is bigger than any earlier number.
- 3) if it is, remember that number
- 4) keep going until you have looked at <sup>all of</sup> them
- 5) the last numbered you remembered is the largest
- 6) end.

24

Assignment

On Write an algorithm that will test whether the given number is prime or

Assignment

On Write an algorithm to find factorial of given number.

After the student will follow the following steps:

Step 1: Ask the user to enter the number.

Step 2: Initialize the variable sum = 1.

Step 3: Initialize the variable i = 1.

Step 4: If i <= n then go to step 5.

Step 5: sum = sum \* i.

Q Assignment  
Write an algorithm to calculate per perimeter (25)  
and area of triangle, rectangle.

Step 1: read given numbers as  $l, b$  for rectangle

Step 2: find double of sum of  $l$  and  $b$  and  
store in  $d$

Step 3) store value of  $(2)$  in  $d$

Step 4) find product of length and breadth

Step 5) store product in  $e$ .

Step 6) Print  $d$  and  $e$ .

Step 7) End program.

(26)

## Flowchart:-

\* A flowchart is a step by step diagrammatic representation of logic paths to solve any problem. It helps to analyse the problem and plan its solution in systematic manner.

\* Flow chart when translated into a programming language, it results to a complete program.

Advantage of flowchart:-

- \* It displays logic of the problem in pictorial fashion which facilitates error checking of the algorithm.
- \* It is good means of communication to other user.
- \* It allows the problem solver to break the problem in parts
- \* ~~It is~~ It is a permanent record of a solution which can be consulted later.

Assignment

Dif. b/w algo & flowchart.

- |  |  |
|--|--|
| 1) Algorithm is step by step procedure to solve a problem  | 1) It is a pictorial representation of a process |
| 2) In algorithms plain text are used                       | 2) In flowchart symbols / shapes are used        |
| 3) Solution is shown in non-computer language like English | 3) Solution is shown in graphical format         |
| 4) Algorithm does not follow any rules                     | 4) Flowchart follows rules to be constructed     |

- |  |  |
|--|--|
| 5) Algorithm is easy to debug              | 5) Flowchart is hard to be debug       |
| 6) Difficult to show branching and looping | 6) Easy to show branching and looping. |

(27)

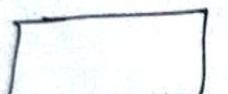
### ② Symbols used in flow chart:



oval used to indicate either start or stop of the program.



parallelogram used to indicate and used to for input and output of a problem.

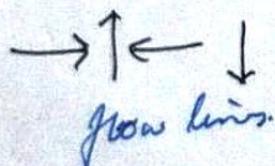


rectangle it is used to indicate any set of processing instruction and processing operation.



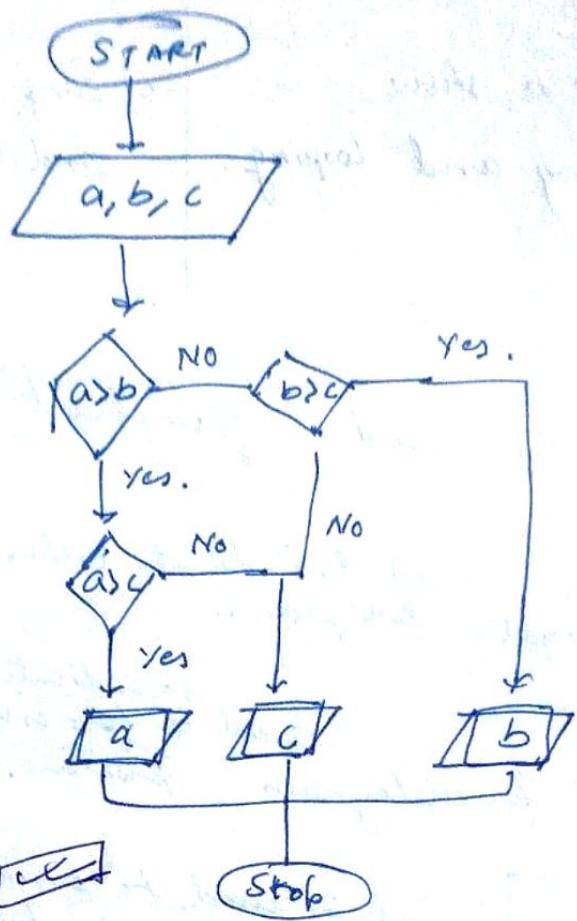
diamond box

it used to indicate step of decision making.

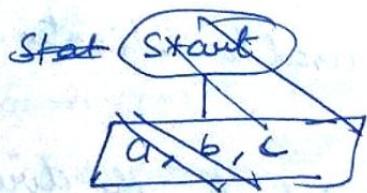


it indicates the direction of path being followed in flowchart.

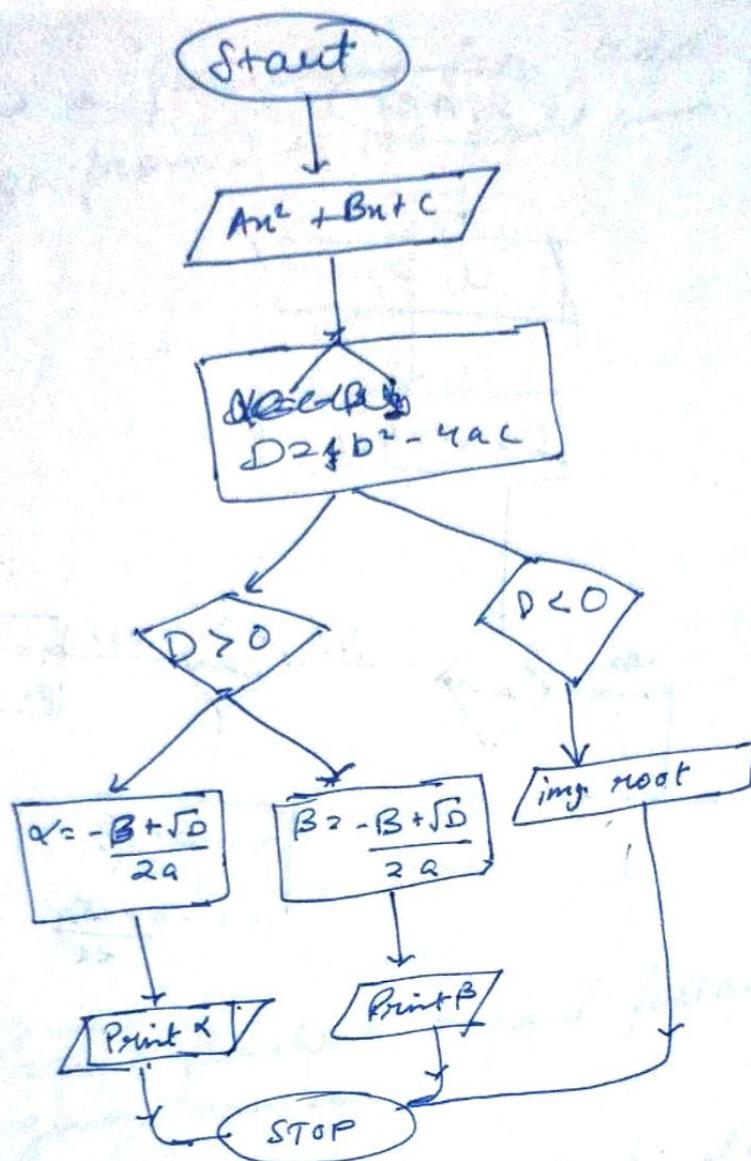
② Draw a flowchart to find largest among 3 numbers.



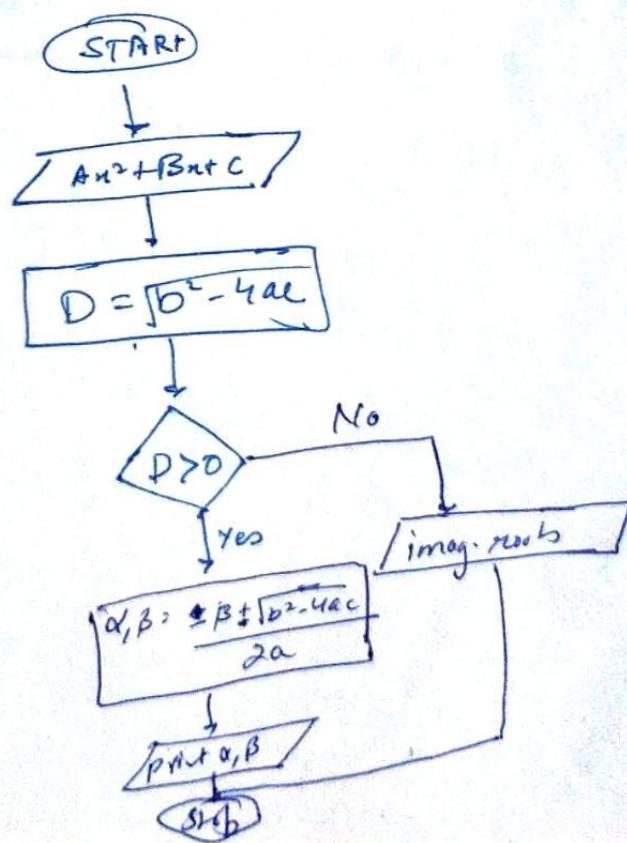
Draw flowchart to find roots of quadratic equations:-



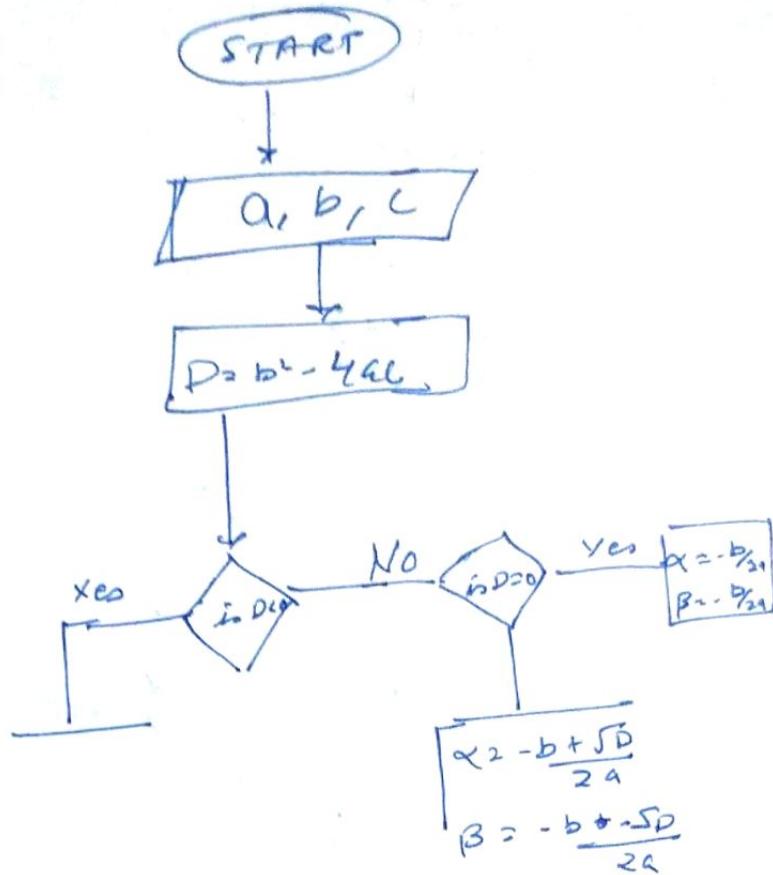
29



50

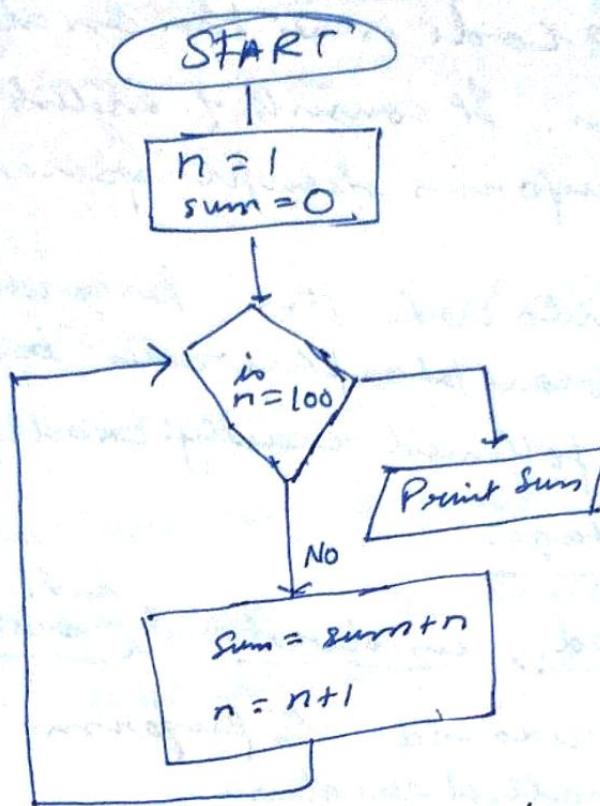


30



Qn Draw a flowchart for adding integer from 1 to 100 print sum.

(31)



Qn <sup>Assignment</sup> Draw a flowchart to find factorial of given number.

Qn <sup>Assignment</sup> Draw a flowchart to check if a given no. is prime.

(32)

## Pseudo Code

Pseudo code is neither an algorithm or program. It consists of English like statement which performs specific operation.

In ~~pseudo~~ pseudo code the program is written in words or short phrases but the syntax is not followed exactly exactly.

### Advantages

Easy to read, understand and modify.

- \* Write a pseudo code to perform basic mathematical operations.

Read  $n_1, n_2$

sum =  $n_1 + n_2$

diff =  $n_1 - n_2$

prod =  $n_1 * n_2$

quo =  $n_1 / n_2$

print sum, diff, prod, quo.  
end.

## UNTF II - C

(33)

→ C is a high level language designed in Bell laboratory in year 1970 by Dennis Ritchie Dennis Ritchie.

Some features:-

- ① C is a combination of low level and high level.
- ② C supports 44 operators ~~32~~ keywords and 14 separators
- ③ C is a case sensitive language.
- ④ Every C statement ends with semicolon ( ; )
- ⑤ C language supports large set of library functions.
- ⑥ C is function oriented and procedure oriented language

### Structure of C Program

#### ① Comments

Ex → // program to add two numbers.

#### ② Link

Ex → #include <stdio.h>.

#### ③ define

Ex → #define Max 10,000

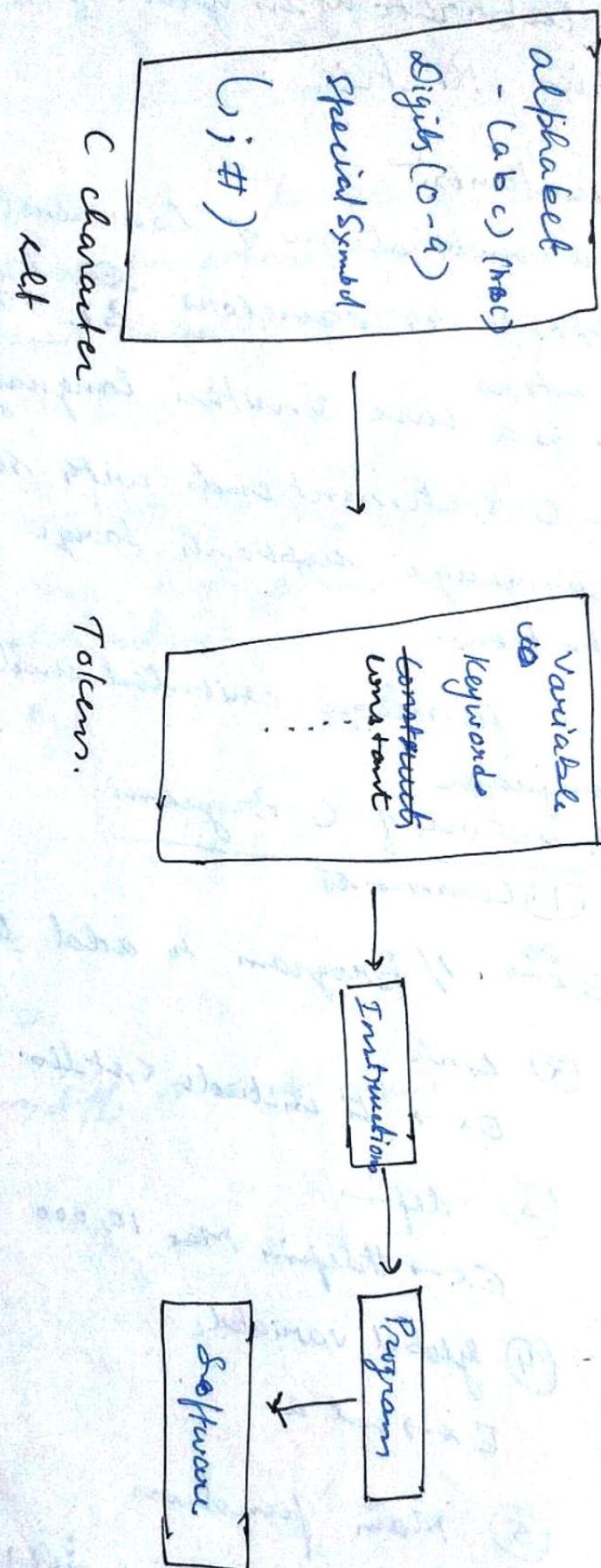
#### ④ global variable

Ex → int a;

#### ⑤ Main function

Ex → int main()  
{}  
}

## How to write C Programming



Character set:-

35

It is a collection of alphabets,  
digits, special symbol

Eg (abc) (ABC)

digit - (0-9)

special symbol (, ; #)

Tokens.

variable → A token whose value is  
not fixed throughout the program  
int a.

Keyword fixed and predefined words for a  
particular task.

Constant.  
 $a = 4$

Def: Token is the smallest unit of the  
program, it consists of variable,  
constants, keywords, special symbols,  
data types, integer.

Find the no. of tokens?

int main()

    {  
        int a;  
        printf ("Enter any number");

        scanf ("%d", &a);

        printf ("%d", a);

    }  
    return 0;

5

3

32 tokens  
in total

36

int

main( )

int( )  
int

print( ) ("y.d") (+ + @ + + + )

return( )

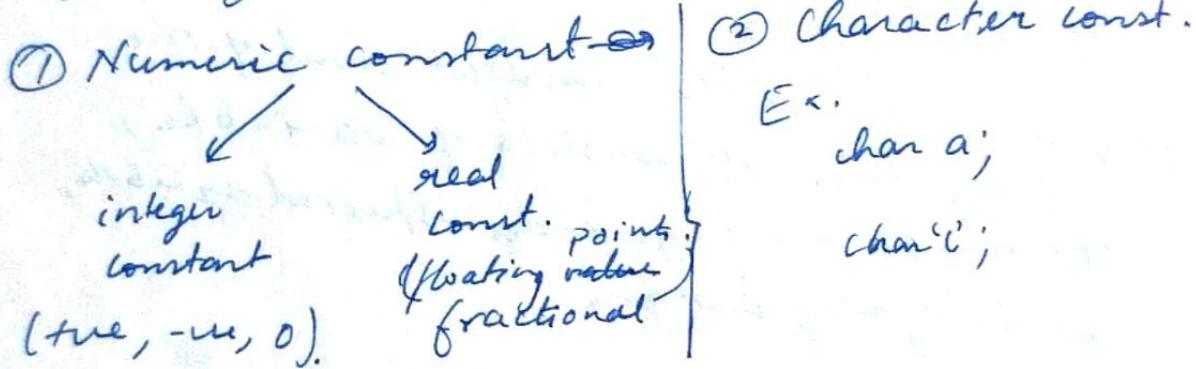
②

25

26  
27  
28

There are 25 tokens in this program.

Constants:- Constant is the value that does not change while execution of the program. It is of two type



Variable. Variable is a ~~res~~ name given for any computer memory location. The purpose of declaring variable is to store data or value. user will access the variable by its name, where compiler will access by its address.

## Rules for declaring variable.

37

- ① Variable can be in lower case, upper case and Mixed ~~lower~~ case and numbers.

Ex → `int a; X`  
`int a1; ✓`

- ② Variable name can't start with numbers

- ③ Variable name cannot be keyword.

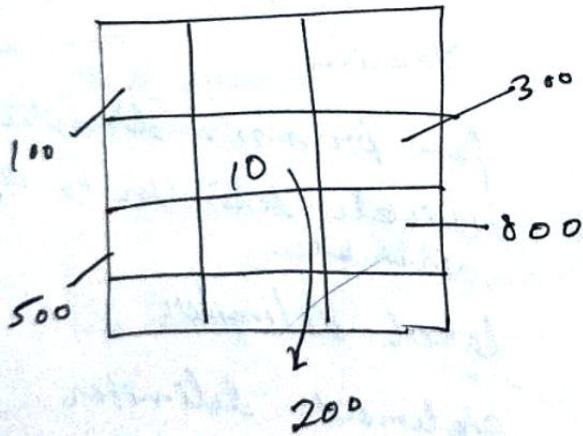
- ④ Variable name can't contain be special symbols except '-' underscore

- ⑤ Variable name can start with character, underscore but not digit.

`int a1;`  
`int a2_;`  
`int _a1;`

NOTE

\* max length of any variable can be 255 characters  
But computer only reads first 32 characters



Syntax:-  
`data type identifier;`  
`int a;`

`a = 10;`

`point ("%.d", a);`

`float per; k: 30`

`printf`  
`printf ("%d", per);`

38 Keywords: In C language some words are reserved to perform specific tasks. They are known as keywords.

[<sup>char</sup> int, float, ~~etc~~, double, unsigned, short,]  
long, const, void,  
data types

### Statement

if, else, break, continue, goto, for,  
while, do, return, size of, switch, case, default,  
struct, union, volatile, extern, ~~using~~,  
enum, register static, ~~typedef~~, ~~type def~~, entry,  
global,

Delimiters: Delimiters are the symbols that has syntactic meaning and has got significance, these will not specify ~~any~~ any operation, to result in a value

They are: Delimiter

#### Meaning

# pre processor directives.  
,

variable delimiter to separate variables

:

label delimiter

;

statement delimiter

()

used for fn call & expression

{}

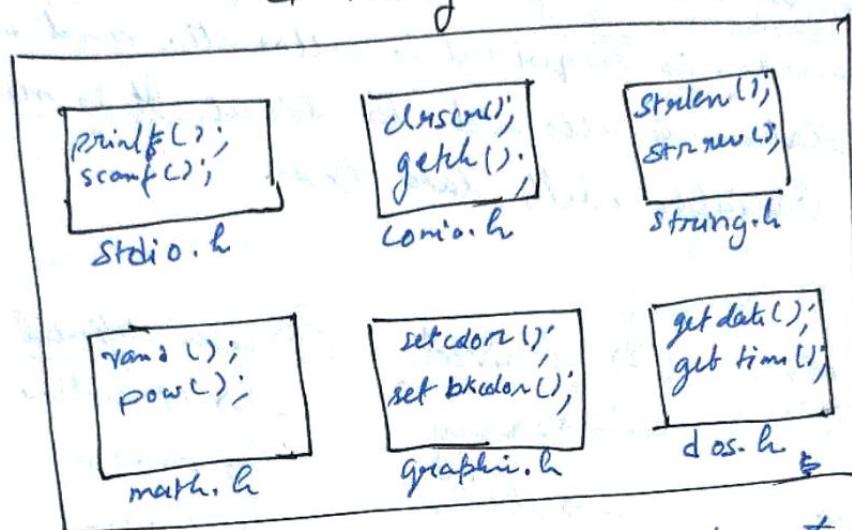
used for blocking making  
blocks of statements

[ ]

used with arrays.

39

Libraries It is the collection all the header files present in C language.

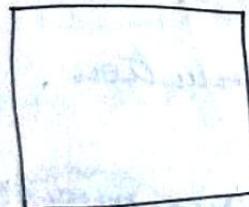


Header file: Collection of similar functions.

I D E

[Integrated Development Environment]

Black screen

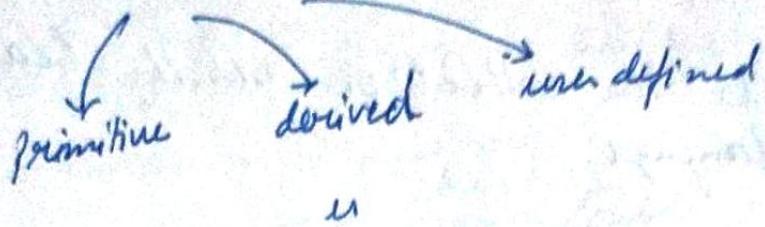


Blue Screen

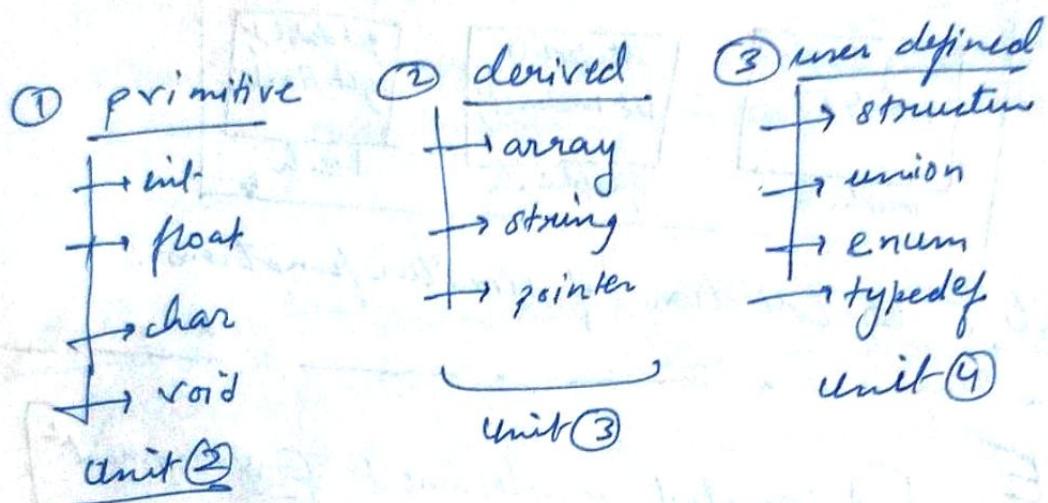
Save - F2
Compiler - Alt+F9
Run - Ctrl+F9

→ [Can't see both  
at same time.] ←

## 40 Data Type :-



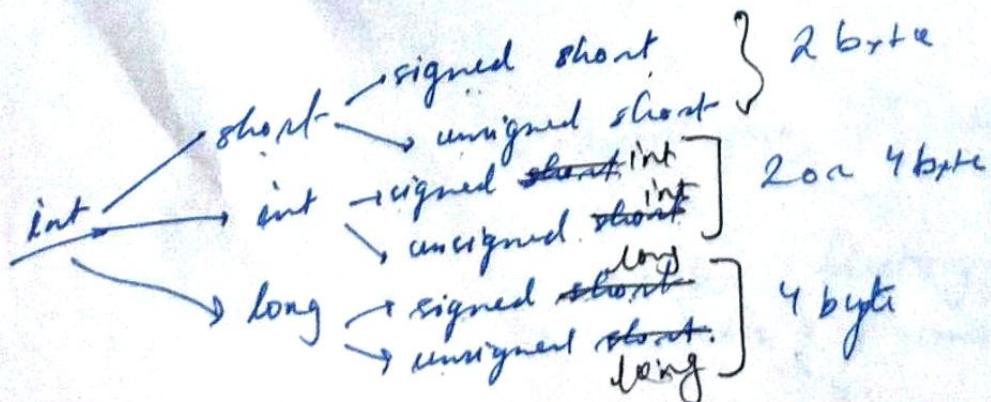
Data type is the representation of data how much memory is required to allocate and what type of data is allowed to store. It is mandatory to define variable with data type.



The primitive data type.

- ④ The primitive → data type directly interact with machine instruction.

- ④ Every data type supports type modifier.



(41)

```

    float   ] 4 byte
    double  ] 8 byte
    long double ] 16 byte.
  
```

\* declarations of signed in 4 ways

short a;

short int a;

signed short a;

signed short int a;

] -ve and +ve both  
values are stored  
format specified (%d)

\* unsigned short a;

unsigned short int a;

] only +ve  
values.  
format specifier (%u)

\* By default the type of variable is signed.

int / short

2 bytes = 16 bit

$$2^{16} = 65536$$

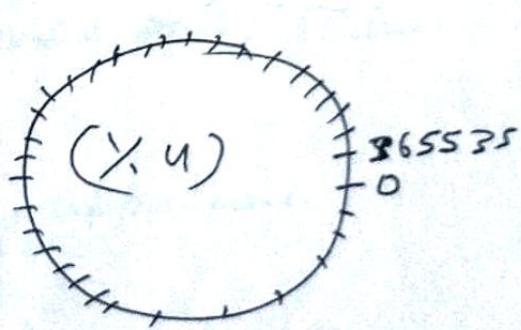
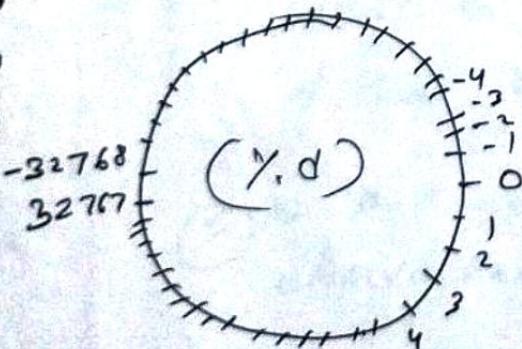


-32768 to 32767

unsigned



0 → 65535



✓ 42

int main()

{

short a = 32767;

printf ("%d", a);

}

a  
-32767

2064.

int main()

{

unsigned short a = -4;

printf ("%u", a);

~~printf~~

3

a  
65532

int main()

{

signed short a = 65538;

printf ("%u", a);

printf ("%d", a);

}

→ a

→ a <

~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~

char → signed → (-128 to 127) 43

1 byte → unsigned (0 - 255)

8 bit

$$2^8 = 256$$

char ch = 'A'

### ASCII

American Standard Code for Information

Interchange.

A - 65  
B - 66

a - 97  
b - 98

0 - 48

# - 35

1 - 49

space - 32

,

enter - 13

:

esc - 9

:

150

:

9 - 56

10

212 total

Assignment  
① Write a program to print ASCII Value of that character.

② W.A.P to convert any upper case character to lower case.

③ W.A.P to convert lower case character to upper case

④ W.A.P to convert digit character to integer.