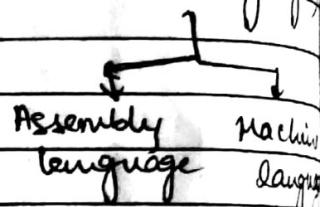
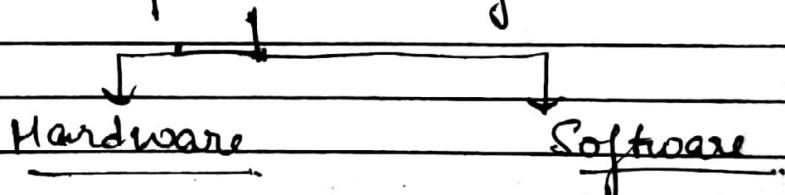


## UNIT - 2 Interaction with Computer

- Computer hardware, software, system software, application software, translator software  
(example, compiler, interpreter)
- Computer language (High level language, low level language)
- operating systems(OS), types of OS, functions of OS



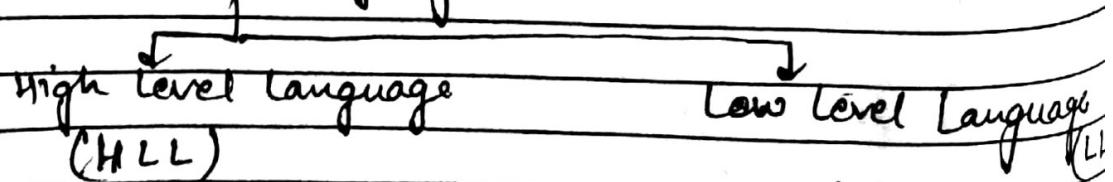
# Computer System consist of hardware and software



→ The ~~not~~ tangible parts of a computer that can be seen or touched.

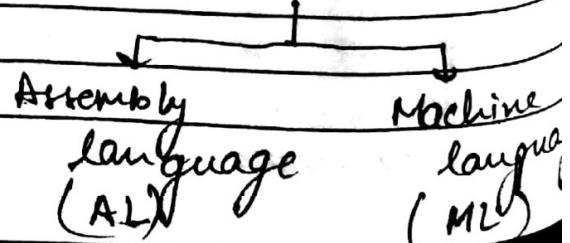
Set of programs that are required to run the hardware.

### Computer Language



Eg: C++, Java, etc.

GOOD WRITE



Program → set of instructions required to instruct the computer about a specific task.

HLL → It uses human readable language for writing programs.

ML → It uses symbols 0 and 1 for writing programs.

AL → It uses mnemonics for writing programs.

(Long Ques) Classification of Software

V.V.T. Software

System Software

- a) system operation s/w (OS)
- b) system utilization s/w
- c) system implementation s/w
  - ↳ Eg: Compilers, DBMS

→ It is a software written for a specific purpose to run different parts of a computer system.

Application Software

- a) Word Processor s/w
- b) Spreadsheet "
- c) Presentation "
- d) Database "
- e) Accounting "
- f) Image Processing "
- g) Geographical Information System (GIS) s/w
- h) Education s/w, etc.

→ It is a software written for a general purpose and can be used by everyone.

Word Processor s/w → An application you used to create and edit a document.

Eg: MS Word → one of the main components of MS office.

Spreadsheet s/w → used for calculation purpose as well as data analysis purpose.

Eg: MS Excel

(mostly in tabular form)

Presentation s/w → used to create presentations.

→ main feature of this is animations.

Eg: Power Point

Database s/w → used to store the database  
(collection of records)

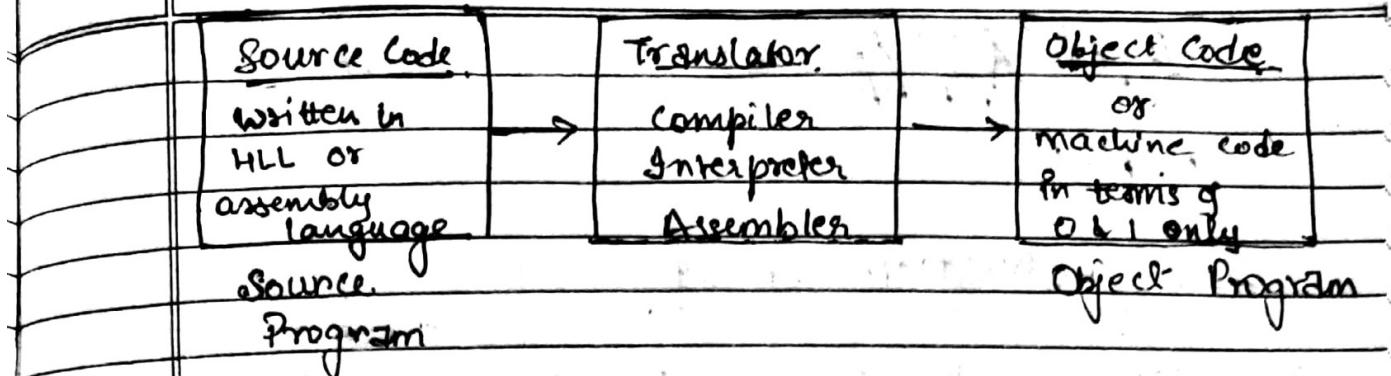
Eg: MS Access

Accounting s/w → used for ~~all~~ ~~editing~~ ~~image~~

Eg: Tally

Used by specific people for business purpose.

Translator Software, it is used to translate the high level or low level language into machine code / object code.



Q. What is the difference b/w compiler and interpreter

### Compiler

- 1) It is a translator software which translates the HLL into machine code in one go.

a) Eg: C++, C

### Interpreter

- 1) It is a translator software which translates the HLL into machine code line by line.

b) Eg: Basic, Java

c. Program → compiler → Object Program

Neto.: c → Compilation → Neto.: obj

Linker ] → system spec  
Loader ]

V.V.I. operating System (OS)

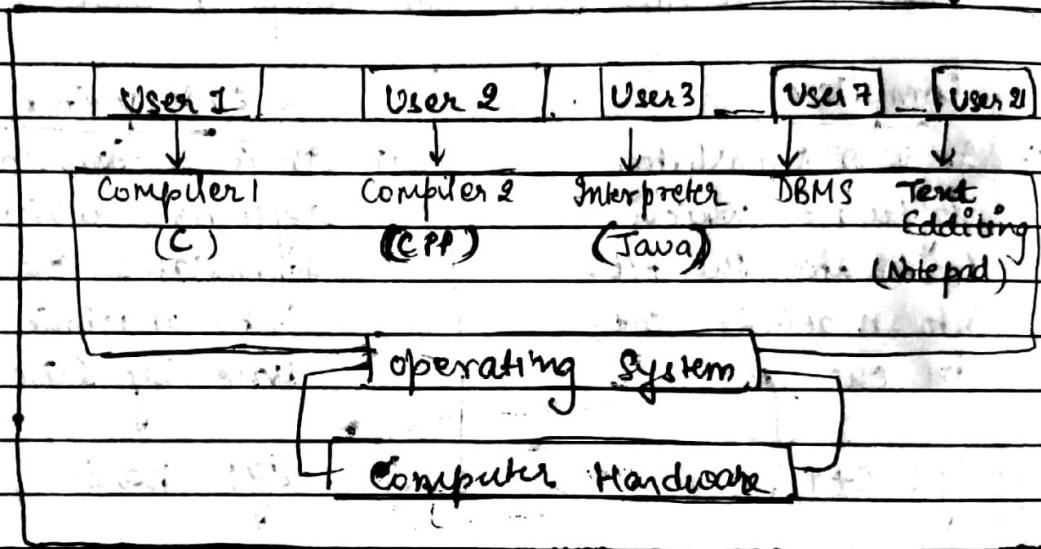
Q. what do you mean by operating system?  
Explain the types of OS and different functions of OS.

*Affrep / d / 821*

110

## INTRODUCTION to OG

Fig View of the components of a computer system



- An OS is a system software or a program that acts as an interface b/w the user of a computer system and computer hardware.
  - It controls and coordinates the use of the hardware among the various application softwares / application programs for the various users.
  - The purpose of OS is to provide an environment in which a user can execute programs.
  - The primary goal of OS is to make the computer

## 4 components

### 1 Computer

system convenient and easy to use for the users.

- The secondary goal is to use the computer hardware in an efficient manner.

Q why OS acts as resource manager?

## Classification / types of OS

OS are classified into different types depending upon their capabilities of processing.

They are :-

1. Single User OS / Single Program OS Eg: MS DOS
2. Multi User / Time sharing OS Eg: MS Windows
3. Multi-tasking / multi-programming OS
4. Multi processing OS
5. Real Time OS
6. Embedded OS

## 4 components of computer system

1. Computer Hardware (Keyboard, Mouse)
2. System Software (OS, compiler)
3. Application software
4. Users

### 3. Multi processing OS.

- if we have more than 1 CPU in computer system, then the OS is called multi-processing.
- Concept of parallel processing
- main objective is to improve CPU utilization

OR

Throughput

Q what is the difference b/w Multiprogramming and multiprocesssing OS.

- Multiprogramming is based on single user
- Multiprocessing / Multitasking OS is based on multiple user

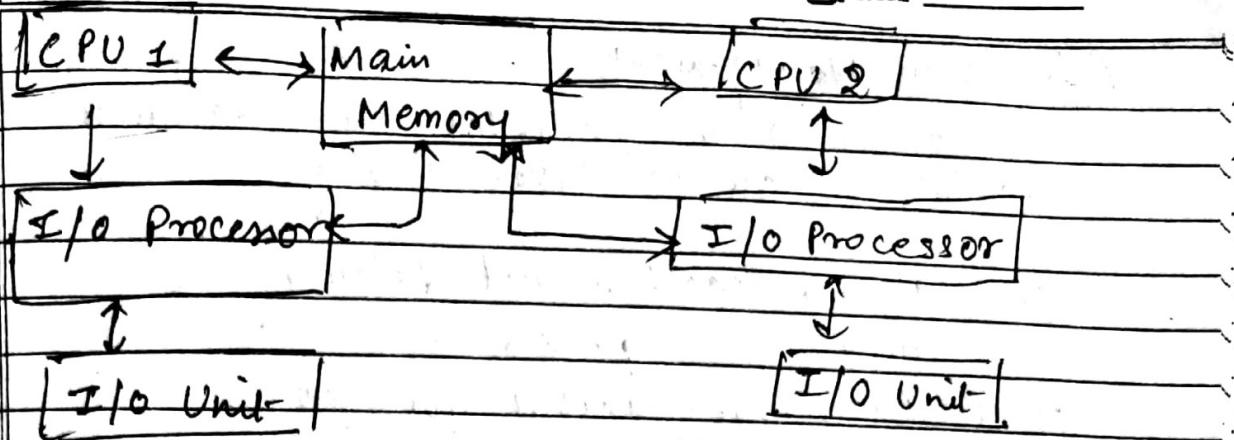
UNIX → multiprogramming (that is single user based)

↓  
Advanced version of UNIX is LINUX

#### Advantage of Multiprocessing OS.

- 1) Increasing throughput
- 2) Economy of scale
- 3) Increase reliability

#### Organisation of Multiprocessing OS.



(2)

Ex:

```

int a, b, sum;           → CPU 1
printf("Enter 2 nos : "); → CPU 2
scanf("%d %d", &a, &b); → CPU 1
sum = a + b;
printf("Sum = %d", sum);
  
```

4. Real Time OS

→ On the Internet doing transactions.

5. ~~Explain~~ Embedded OS

→ In washing machine, refrigerator

N.V.I

Explain various functions of OS

- ① Process Management / Job Management
- ② Memory "
- ③ File "
- ④ I/O "
- ⑤ Protection & sharing "
- ⑥ Device "

GOOD WRITE

## ① Process Management

A process is a program in execution. This module of an OS takes care of:-

- ① creation, Execution and deletion of a system and User process
- ② Providing mechanism for process synchronization
- ③ Process scheduling
- ④ Cancel or returning of process.

## ② Memory Management

- ① The memory management module of an OS manages the main memory of a computer system. Its job is to keep track of free and occupied memory.
- ② It also allocates memory to processes when they need it and de-allocates when they don't need it.

## ③ File Management

- ① It is used for storing files on various storage devices and to transfer files from one storage device to another devices.

## ④ I/O Management

The role of the I/O management is to manage all

GOOD WRITE

control input / output operations and input /

output devices. This results in the flow of data among computers, terminals

and other devices such as ~~per~~ printers, etc.

### (5) Protection & Security Module

This module of an OS provides the protection against destruction and unauthorized access.

The basic operations provided by OS are :-

1. Memory protection
2. Hardware protection
3. I/O protection
4. Encryption
5. Backup of data, etc.

### (6) Device Management

The device management task handled by an OS are :-

1. Open, Close and write device drivers
2. Communicate, control and monitor device drivers.

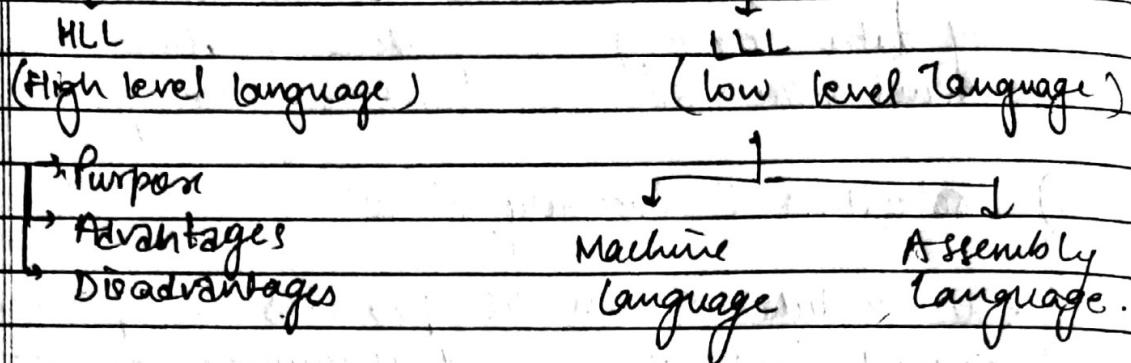
Device Drivers → Software or programs for a device to run.

### (7) User Interface / Command Interface

→ Desktop is the main user interface for windows.

→ OS provides an interface b/w the computer user and the hardware so that they can interact with **GOOD WRITE** applications.

# Introduction to Algorithm & Programming Languages



Algorithm (logic of a problem)

- Step by step representation of instructions to perform a specific task.

Representation of Algorithm

- ① Program
- ② Flow chart
- ③ Pseudo code

I using Program

- Algo is represented by writing ~~any~~ <sup>any</sup> programming language
- Normally, we can use 3 types of control statements.
  - 1) Sequence
  - 2) Decision
  - 3) Repetition

## Characteristics of An Algorithm

- always based on some input.
  - Processing at definite time
  - Instructions should be unambiguous (single meaning)
  - Followed by O/P

Q Write an algorithm for sum of 2 no.s using

Step 1 : Reading 2 no.s say A & B       $\downarrow$  Sequence.

Step 2: : SBT sum = A + B

### Step 3 : Print sum

#### Step 4 : End

## ① SEQUENCE

Every step in our algorithm is in some prescribed order.

## ② DECISION

FORMAT ① if condition  
then process

③ if condition  
then process 1  
else  
process 2

⇒ Switch Case → when we have to check multiple

→ Dangling else  
GOOD WRITE conditions.

**GOOD WRITE**

Q Write an algorithm to check whether 2 nos are equal or not.

Ans

Step 1 : Read two nos A and B

Step 2 : if  $A = B$

then print "Both are Equal"  
else

print "Not Equal".

Step 3 : End

### (8) REPETITION

Q Write an algo to print the sum of 10 natural nos.

Ans

Step 1 : Set  $I = 1$ ,  $N = 10$ ,  $sum = 0$

Step 2 : Repeat Steps 3 and 4  
while  $I \leq N$

Step 3 : Set  $sum = sum + I$

Step 4 : Set  $I = I + 1$

Step 5 : Print  $sum$

Step 6 : End

Q Write an algo for swapping two numbers

Q2 To find out larger of 3 nos

Q3 To find whether a no is even or not.

Q4 To add all even nos from 1 to 100

## II Flow chart

Pictorial / Graphical representation of a problem to be solved (Algorithm)

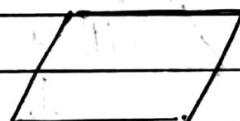
Q. What is the significance of using flowcharts in an algorithm.

Q. Explain the advantages and disadvantages of using flowchart in an algorithm.

### Basic Flowchart Symbols



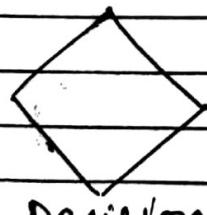
Start or  
End / terminal



Input /  
Output



Programming



Decision

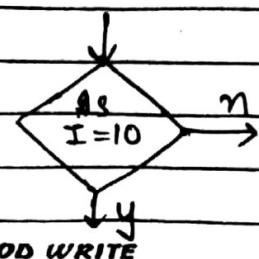


Arrows for  
flow



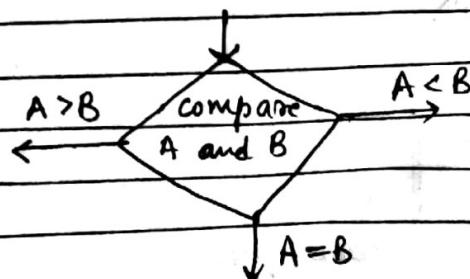
connectors

→ 2 way branching.

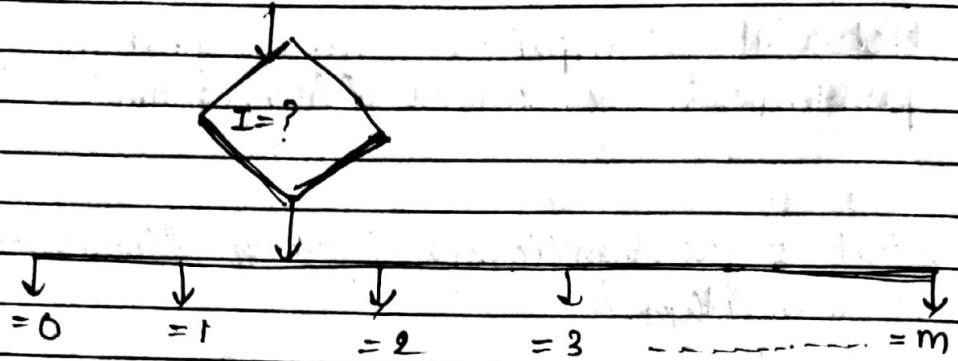


GOOD WRITE

→ 3 way branching

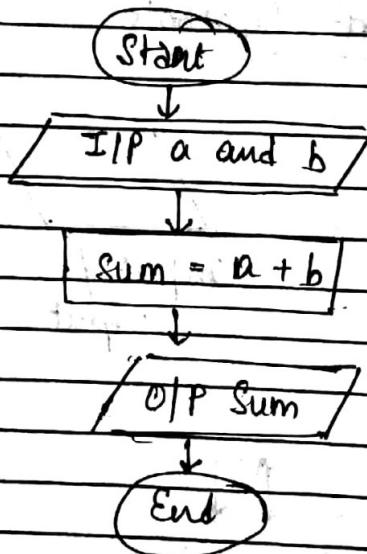


⇒ Multi-way branching



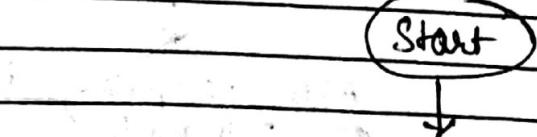
SPPU Q Write an algorithm using flow chart to display sum of 2 no.s.

Ans:

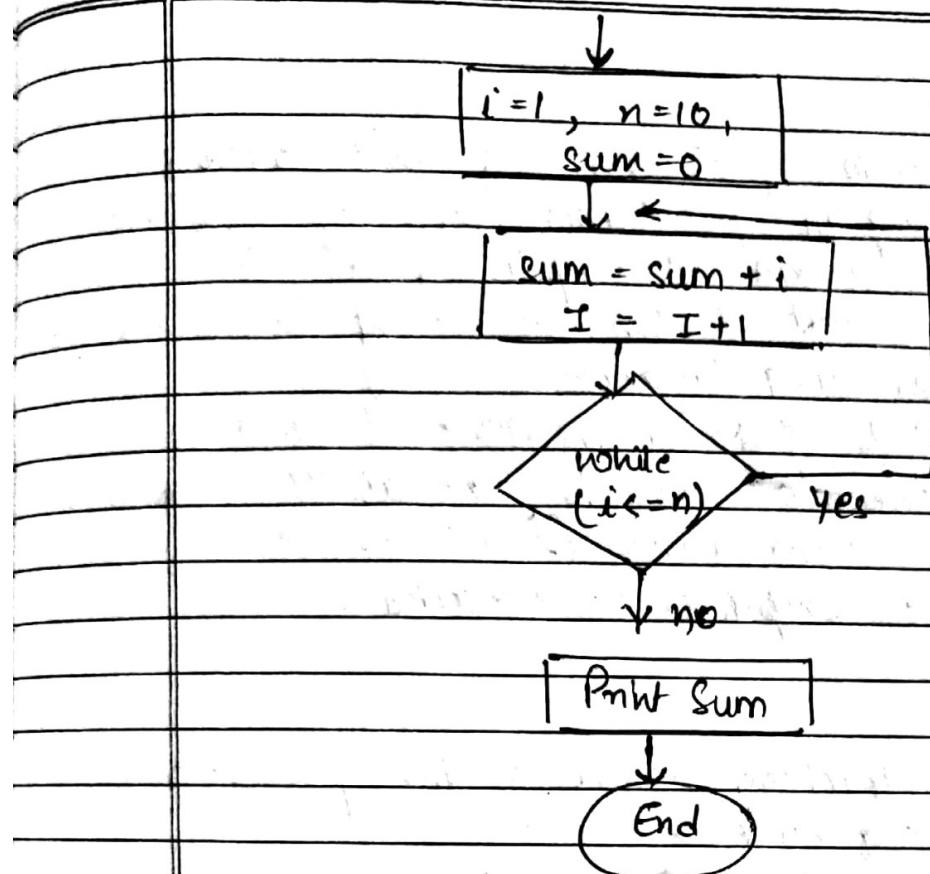


Q Algo using flow chart to find sum of first 10 natural no.s

Ans:

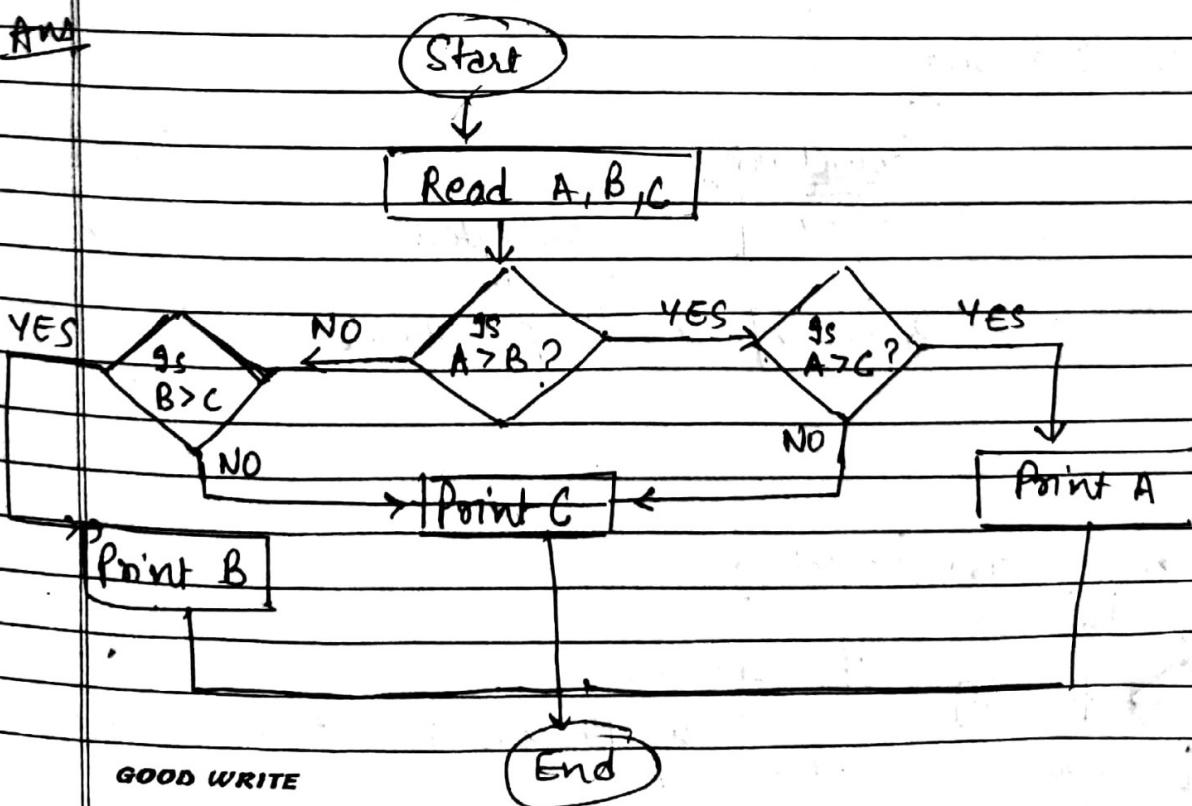


GOOD WRITE



Q Using flowchart, find max of 3 no's

Ans



### III Pseudo code

- It is a program planning tool that allows programmers to plan programs logic by writing programs instruction in simplified English language so that it can be easily converted to any programming language.
- Also called PDL (Program Design Language)
- Pseudo code doesn't include details like variable declaration subroutines (functions) etc.  
It cannot be compiled or executed.

The control structures used in the pseudo code are :

1. Sequence
2. Selection (Decision)
3. Iteration (Repetition)

#### 1. Sequence

##### Flowchart

Process 1

↓  
Process 2

converted

to pseudo code

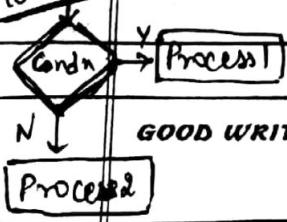
##### Pseudo code

Process 1

Process 2

#### 2. Decision (Selection)

##### Flowchart



If condition

then Process 1

else

then Process 2

} Pseudo code

## 3. Repetition (Iteration)

Flowchart

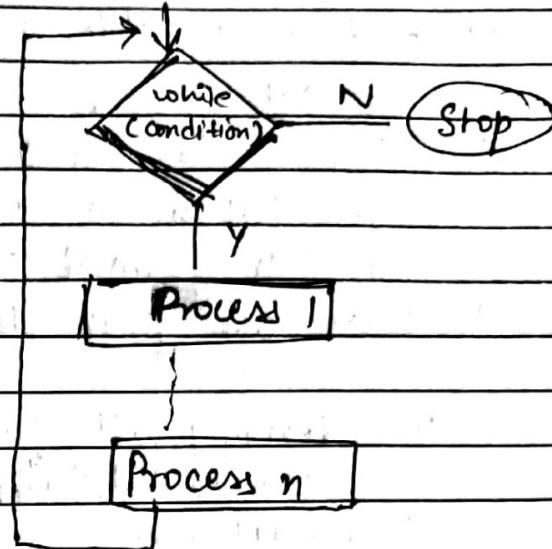
~~Pseudo code of while (condition)~~

Process 1

;

Process n

end



Q Print the sum of two no's

- Read the values of A & B
- Compute sum by adding A and B.
- Print the sum
- Stop / End

Sum of first <sup>10 natural</sup> numbers

Initialise I to 1 and sum to zero

while ( $I \leq 10$ )

Add I to sum and store

to sum

Increment I

Print sum

end

Print final sum

## Decision Table

- A decision table is a powerful tool to debug and prevent errors.
- It represents conditions and the respective action to be taken to address them in a structured tabular format.
- There are four steps for creating a decision table.

1. Identify all possible conditions to be addressed.
2. Determine ~~absence~~<sup>actions</sup> for all possible specified conditions.
3. Create maximum possible rules (that depends upon the conditions).  
If we have  $n$  conditions, then possible rules are  $2^n$ .
4. Define action for each rule.

Figure

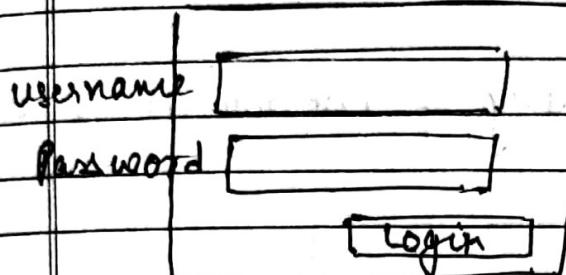
Condition / Actions	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>	R <sub>5</sub>	R <sub>6</sub>	R <sub>7</sub>	R <sub>8</sub>	R <sub>9</sub>	R <sub>10</sub>	R <sub>11</sub>	R <sub>12</sub>	R <sub>13</sub>	R <sub>14</sub>	R <sub>15</sub>	R <sub>16</sub>	R <sub>17</sub>	R <sub>18</sub>	R <sub>19</sub>	R <sub>20</sub>	
Conditions	C <sub>1</sub>																				
Actions	A <sub>1</sub>																				
Stub	A <sub>2</sub>																				
!																					
C <sub>n</sub>	A <sub>n</sub>																				

eg

Decision Table for login screen.

T/F = True / False

E/W = Error / webpage



Condition / Action	Rules.			
	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>
c <sub>1</sub> Username (T/F)	F	F	T	T
c <sub>2</sub> Password (T/F)	T	F	F	T
A <sub>1</sub> Output (E/W)	E	E	E	W