unit - 1

Introduction TO Data Structure. (M-4)

Q.1 Define Keywords, constants.

→ Keywords.

Keywords are also known as the reserved words because the meaning of these words are already pre-defined by the compiler.

- Example :- int, char, float, double.
- There are total 32 keyword in C.

• Constants:

- constants are the value given to the identifier that do not change their values throughout the execution of the programm.
- constants are declare only by using keyword constant.
- example :- constant int age = 20;

Q.2 Define variable, How to declare variable?

→ The meaningful name given to the memory location where the value is store is known as variable.

- Syntax for declaring a variable.

  data-type var-name;

- example :- int x = 10

Q.3 What is datatype ? Give example.

→ The datatype specifies the size & type of information the variable will store.

- Example :-

1. int.

This is used for declaring non-decimal integer type variable. The datasize of integer is 2 byte

2. char.

This is used for declaring alphabets & character the datasize of char is 1 byte / byte

3. Float.

This is used for declaring fractional number In this we can take 6 digit after decimal point. The datasize of Float is 4 bytes.

4. Double.

This is also used for declaring fractional numbers. In this we can take upto 10 digits after decimal point the datasize of double is 8 bytes.

Q.4 What is an Array? How to declare array? how to intialize an array?

→ Array is a collection of multiple data of the some data type.

- Syntax.

- Datatype array name [array size];

- Declaration & initialization of array;

```
# include<stdio.h>
#include<conio.h>
  void main()
{
int a[5] = {1,2,3,4,5};
printf ("%d", a[0]);
for(int i=0; i<5; i++)
{
printf ("%d\n", a[i]);
}
getch();
}
```

Write a C program to check wheather the given number is even or odd.

```
#include <stdio.h>
#include <conio.h>
 void main ()
{
 int n;
clrscr ();
printf (" enter the number:");
scanf (" %d ", &n);
if (n %2 ==0)
{
printf (" number is even ");
 }
else
{
printf (" number is odd");
}
getch ();
}
```

output:

Enter the number ; 16

Number is even.

Date
Page

★ For loop :

- example

```
for (i=0 ; i<=10 ; i= i+2)
{
  printf("%d\n",i);
}
```

- output.

0
2
4
6
8
10

★ While loop. ;

- Example

```
int i=0
while (i<5)
{
 printf("%d\n",i);
 i++ ;
}
```

- output.

0
1
2
3
4

★ Do While. ;

- Example

```
int i=0
do
{
 printf("%d\n",i);
 i++;
} while (i<5)
```

- output.

0
1
2
3
4

1) Data :

- Data is collection of numbers, alphabets, symbols to combnie to represent information.

- computer takes row data as input & after processing of it, it produces refined data as output.

2) Atomic data :

It is a non decomposible entity.

- Example.

- An integer value.
  523

- character value.
  A

- Cannot be further divided.

- If we further divided 523 5,2,3 then the meaning will lost.

3) Composite :

It is composition of several atomic data, Hence it can be further divided into data.

- Example :

DOB : 08/02/2007
- can we seper

- 1st gives date of birth.
- 2nd gives month of birth.
- 3rd gives year of birth.

★ Data structure.

- It is a particular way of organising data in computer memory so that memory can be used efficiently. (easily)

- Data structure deals with the representation of data considering not only elements store but also relation between each other.

- processing and accessing should be efficient.

* Data structure mainly specifies the follows 4 things:

1. Organization of data.
2. Accessing method.
3. Degree of Association.
4. processing methods.

* Derived datatypes:

1) Array - It is collection of homogeneus elemen

ex:- int a[6];

a[0] a[1] a[2] a[3] a[4] a[5]

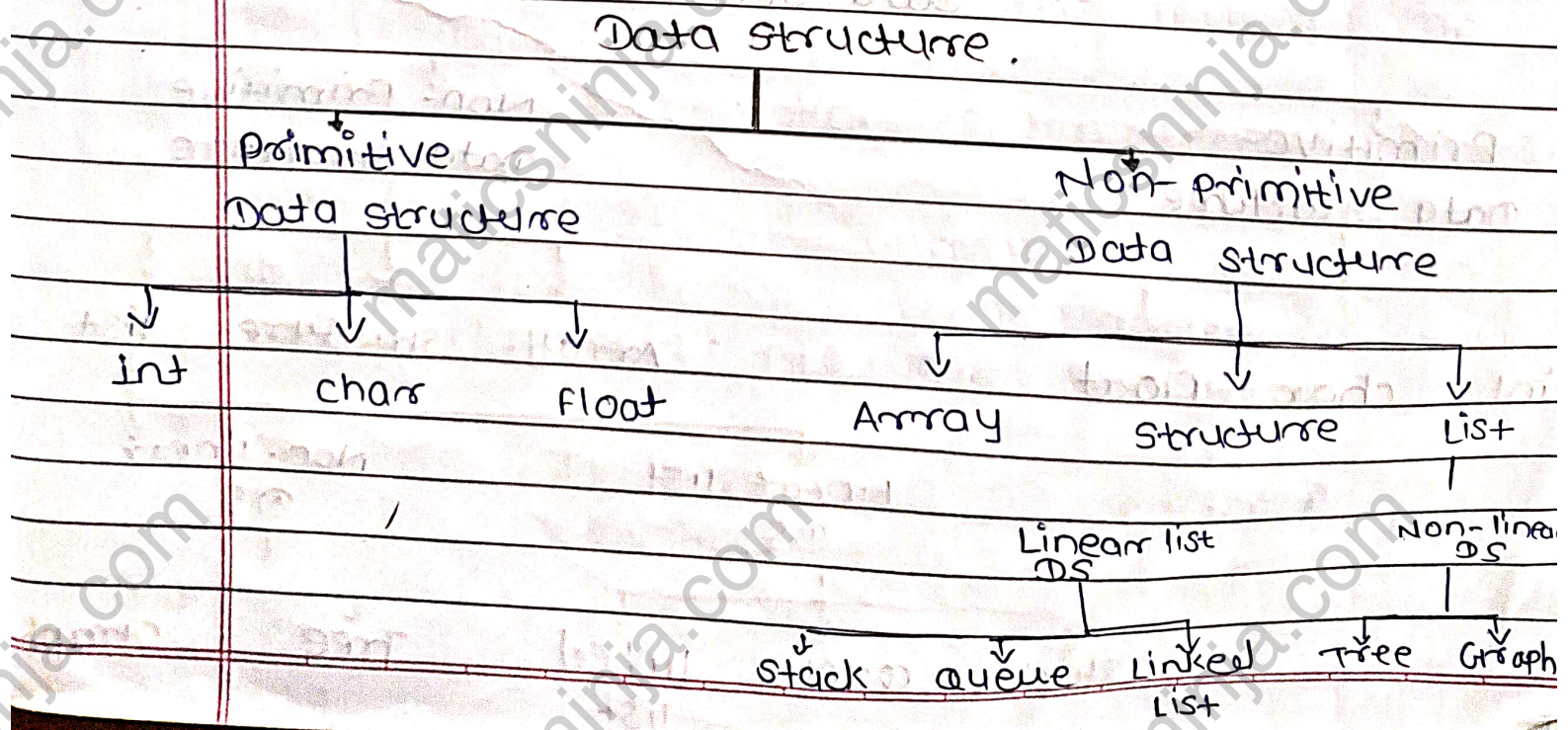
2) Structure - It is collection of heterogeneous element.

ex:- struct student.
{
int roll no;
char name [10];
char Address [120];
}; struct student s1;
s1. roll no;
s1. name;
s1, address;

3) union:- Major different in structure and union is in terms of storage.

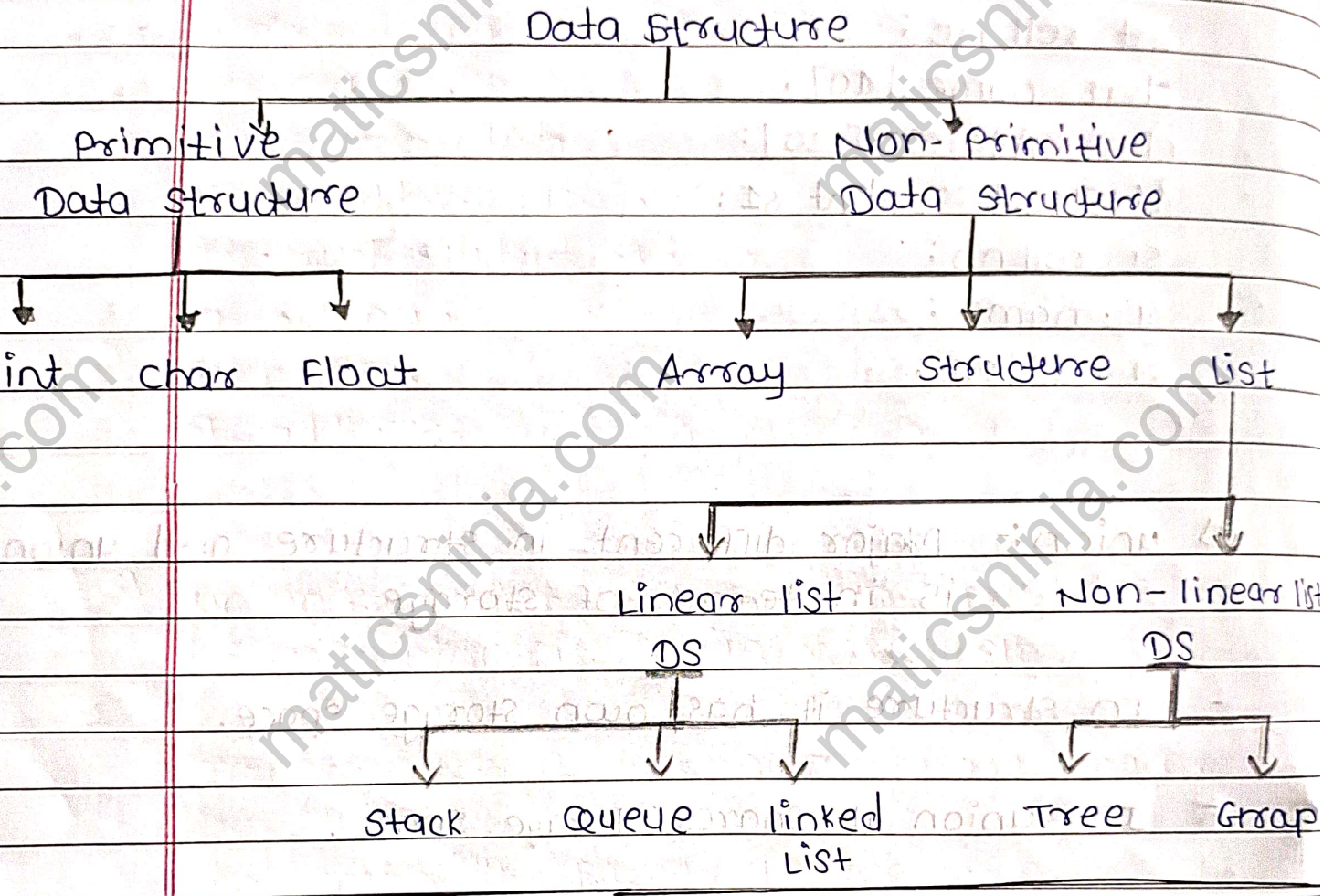
- In structure it has own storage space.
- In union it shares storage space.

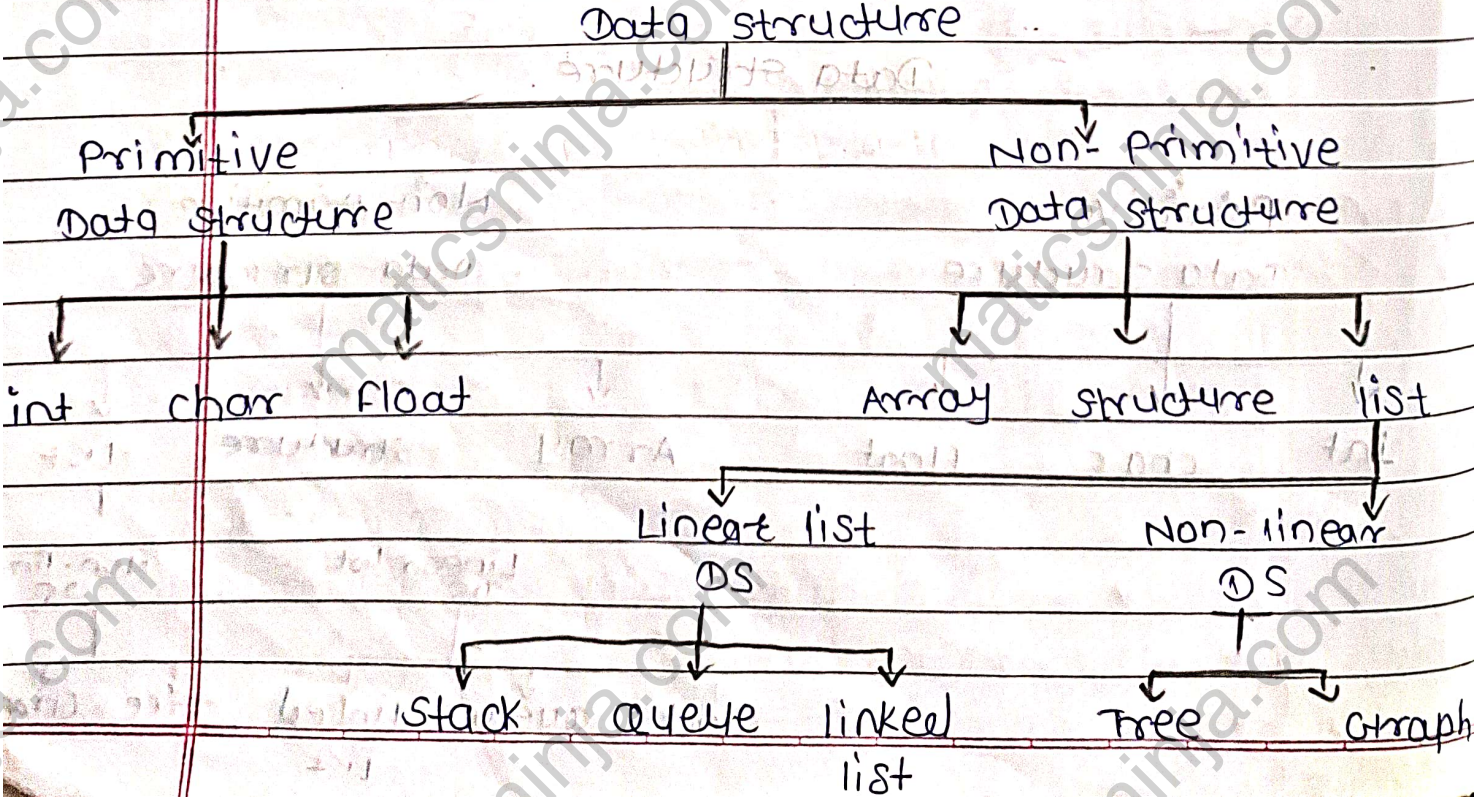
★ classification of data structure.



Imp

H.W 1) Draw the classification of datatype (2 times).





Data :- Data is a collection of information but in Row form

Information :- When data is process it becomes information.

H.W 2) Define concept of data structure.

- It is a particular way of organising data in computer memory so that memory can be used efficiently.

- Data structure deals with the representation of data considering not only elements store but also relation between each other.

- Processing and accessing should be efficient.

* Data structure mainly specifies the follows 4 things:

1) organization of data.
2) Accessing method.
3) Degree of Association.
4) processing methods.

* Derived datatypes.

1) Array :- It is collection of homogeneous element.

ex:- int a[6];

a[0] a[1] a[2] a[3] a[4] a[5]

2] Structure - It is collection of heterogeneous element.

ex:-

```
struct student.
{
int roll no;
char name [10];
Char Address [120];
}; struct student s1;
s1 roll no;
s1 name;
s1 address;
```

(3) union :- Major different in structure and union is in terms of storage.

- In structure it has own storage space.
- In union it shares storage space.

★ operation of data structure.

1. Traversing
2. Insertion
3. Deletion
4. sorting
5. Searching
6. Merching

RAM - Random Access Memory

Main Memory → RAM

pointer - ~~to~~ int address store करने

Date
Page

* Define concept of data structure.

Data structure :- Data structure is particular ways of organising Data in a computer memory can be used efficiency.

They are divided into two parts:

1. Primitive Data structure.
2. Non- Primitive Data Structure.

1. primitive Data Structure.

primitive data types are a set of basis data types form which all other data types are constructed.

Ex:- int, char, float.

int :- It is used for declering non-decimal integer type.

char :- This is used for decimal alphabet & character.

Float :- This is used for declaring fractional number.

2. Non-primitive data structure :-

Non-primitive data structure is a data structure that allows to store multiple data type values.

ex:- Array, structure, list.

Array :- Array is a collection of ~~heterogenous~~ homogenous element.

list :- list is a data structure that stores element in an ordered & sequential manner.

list are divided into two parts.

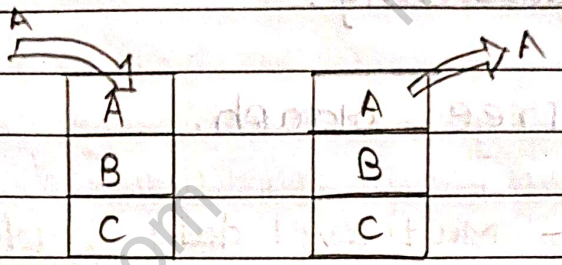
1. Linear Data structure.
2. Non-Linear Data structure.

1. Linear data structure.

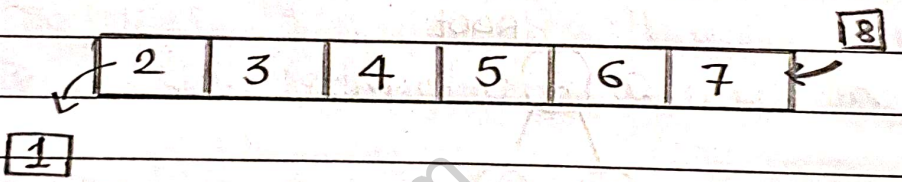
A linear data structure is a type of data structure that stores the data linearly or sequentially

ex:- Stack, Queue, linked list.

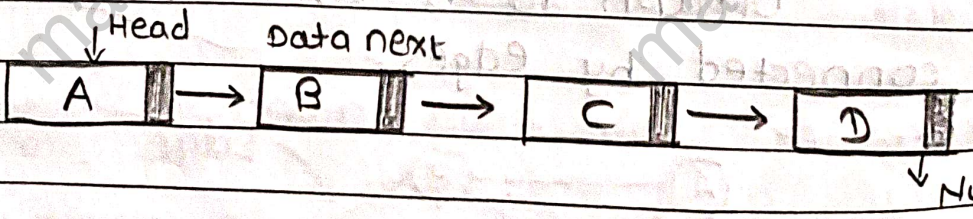
- **Stack:-** A stack is a linear data structure that follows the last - In - first - out (LIFO) principle, It behaves like a stack of plates where the last plate added is the first one to be removed.



- **Queue :-** A queue is a fundamental data structure that follows the first - In - First out [FIFO], Where the first one element added to the queue is the first one to be removed.



- **linked list :-** A linked list is a fundamental data structure, It consist of nodes where each node contain data & a reference (link) to the next node in the sequence.



Linear DS:- A DS is called Linear if all of its elements are arranged in the linear order.

Non-linear DS:- This DS does not form a sequanes, i.e each iteam or element. two or more iteam in non-linear arrangended

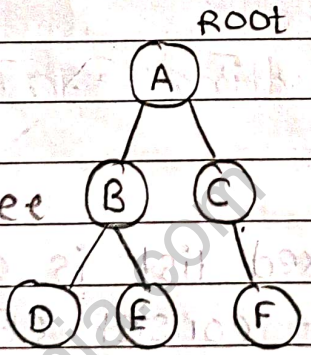
2) Non-linear data structure.

It is a form of data structure where are data elements don't stay, arranged linearly or sequentically.

ex:- Tree, Graph.

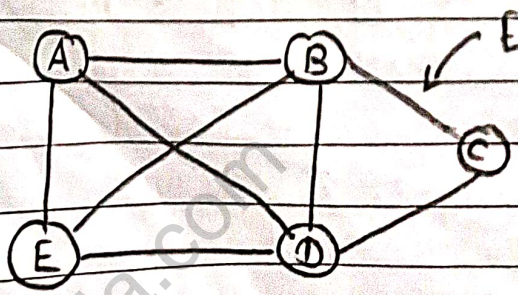
① Tree:- Multi level data structure.

Tree data structure is a hierarchical structure that is used to represent & organize data in a way that is easy to navigate & search.



② Graph:-

Graph is a collection of nodes connected by edges.



Date 23/12/21
Page

* operation of Data structure.

1. Traversing.
2. Insertion
3. Deletion
4. Sorting
5. Searching
6. Merching.

1. Traversing.

- Traversing a data structure is arising each data and arising only once.

2. Insertion

- Insertion is adding a new data in data structur

3. Searching.

- The searching to find location of data in within the data structure.

4. Deletion.

Deletion removing data and and data structure.

5. Sorting.

5. Sorting.

- Sorting arranging of data in logical data.

6. Merching.

Merching is combining of two similar data structure.

---

* Algorithum.

- In algorithum is set of step to require a solve the problem

* properties of Algorithum.

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

1. Input

- Input data applied externally.

2. output

- output that is result of programm.

3. Finitness.

- In every case algorithum terminast after finitneness of step.

4. Definiteness.

- The step be clear and unambiguous.

5. Effectiveness.

- Algorithum is should be written basic instruction it should be visible to converd the alogrithum into programm.

* Alogrithum for addition of 2 Number.

Step 1 :- Start.

Step 2 :- Declare three variable, num1, num2, sum

Step 3 :- Read the first number.

Step 4 :- Read the second number.

Step 5 :- Add 2 number & store result into sum.

Step 6 :- Print the result.

Step 7 :- Stop.

* Flowchart.

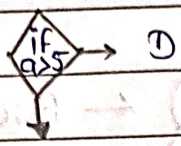
- The pictural representation of your problem is called as Flowchart.

* Symbol use in Flowchart.

1. Termind symbol (Start / Stop).

② Assignment statement → 

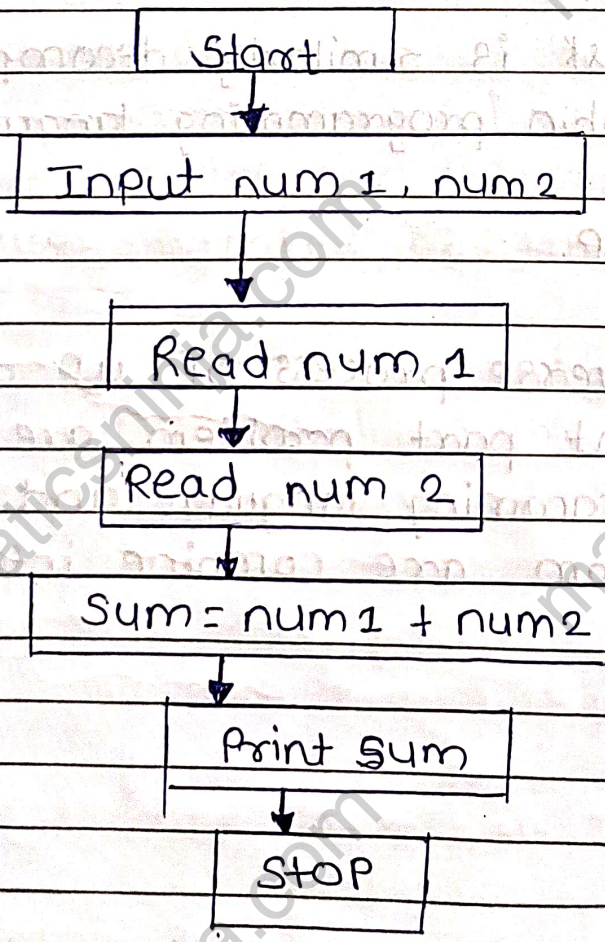
③ Print → 

④ Decision making → 

⑤ Flow Indication → 

⑥ connector → 

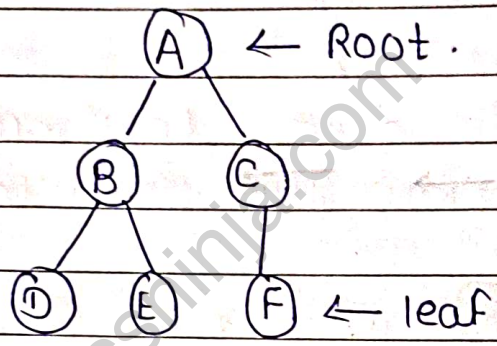
★ Flowchart of addition at two numbers.



Date 24/7/24
Page

* Different Approaches for disigning of Algorith

1. Top-down.
2. Bottom-up.



1. Top down Approach.

- A programmer tries to partation solution into subtask
- Each subtask is similarly decompost all task express within programming language.

2. Bottom-up.

This is reverse process of up down approoach the different part problem are 1st solved using programming language and then peaces of programm are combine into a computer programm.

* Analyse Algorithum.

Algorithum can be Analyse according to two factors :-

1) space complexity.
2) Time complexity.

1. space complexity.

The space ~~com~~ required for an algorithum is called as space complexity.

2. Time complexity.

The time require for exicusion of an algorithum is called as time complexity.