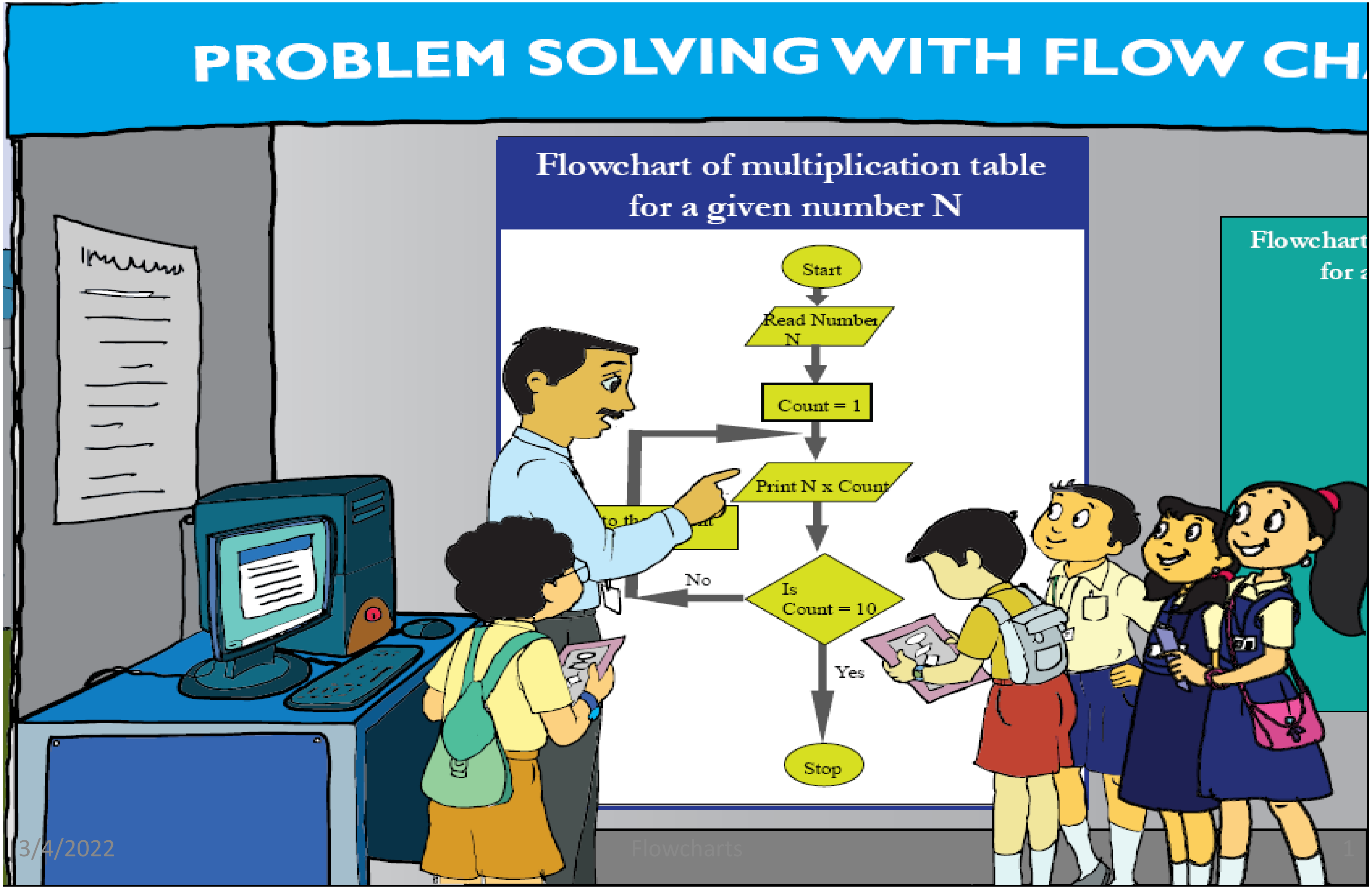
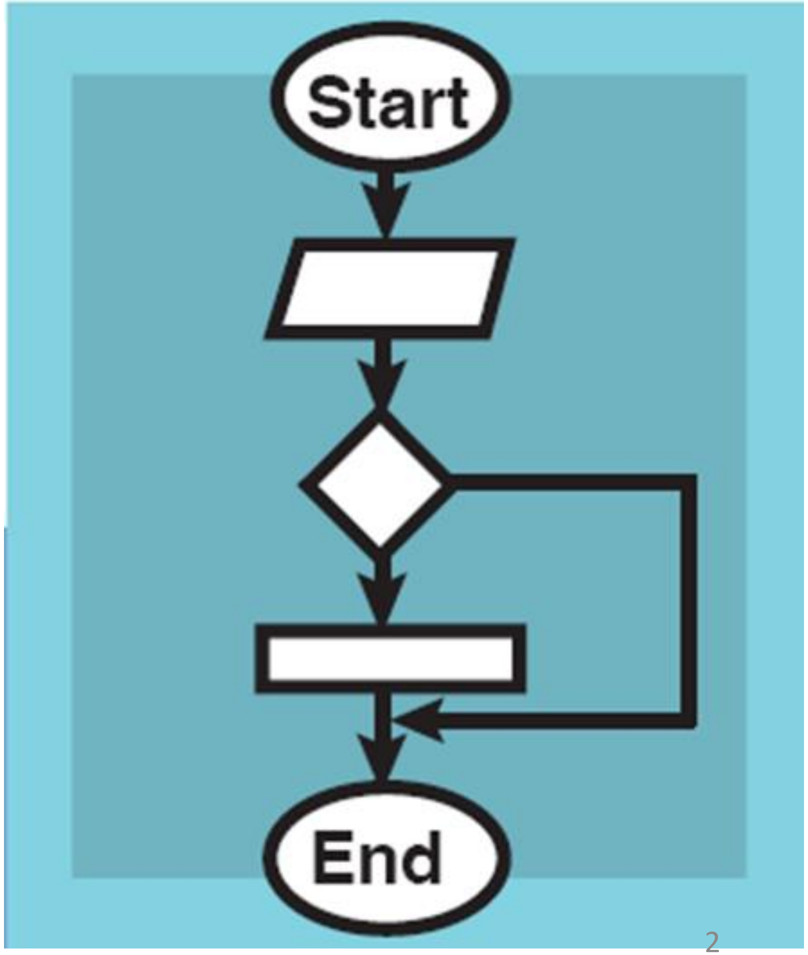
FLOW CHARTS


PROBLEM SOLVING WITH FLOW CH
Flowchart of multiplication table for a given number N
Start
Read Number N
Count = 1
Print N x Count
to th
nt
No
Is Count = 10
Yes
Stop
Flowchart
for a

In this lesson, you will learn:

- To draw a flowchart for problem solving.
- Some uses of flowchart



What is a flowchart?

- A flowchart is a picture (graphical representation) of the problem solving process.
- A flowchart gives a step-by-step procedure for solution of a problem.

TYPES OF FLOWCHARTS

Computer professionals use two types of flowcharts :

Program Flowcharts

System Flowcharts

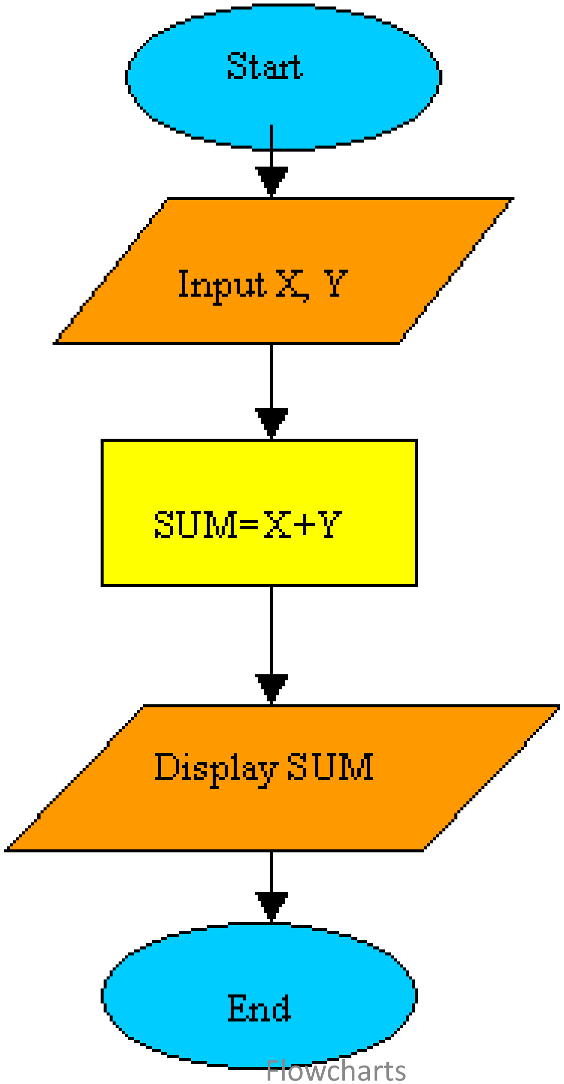
Program Flowcharts

These are used by programmers. A program flowchart shows the program structure, logic flow and operations performed. It also forms an important part of the documentation of the system. It broadly includes the following:

- Program Structure.
- Program Logic.
- Data Inputs at various stages.
- Data Processing
- Computations and Calculations.
- Conditions on which decisions are based.
- Branching & Looping Sequences.
- Results., Various Outputs.

Program Flowcharts

Flowchart SUM: To display total of two numbers



System Flowcharts

System flowcharts are used by system analyst to show various

- processes, sub systems, outputs and operations on data in a system.
- *In this course material we will be discussing program flowcharts only*

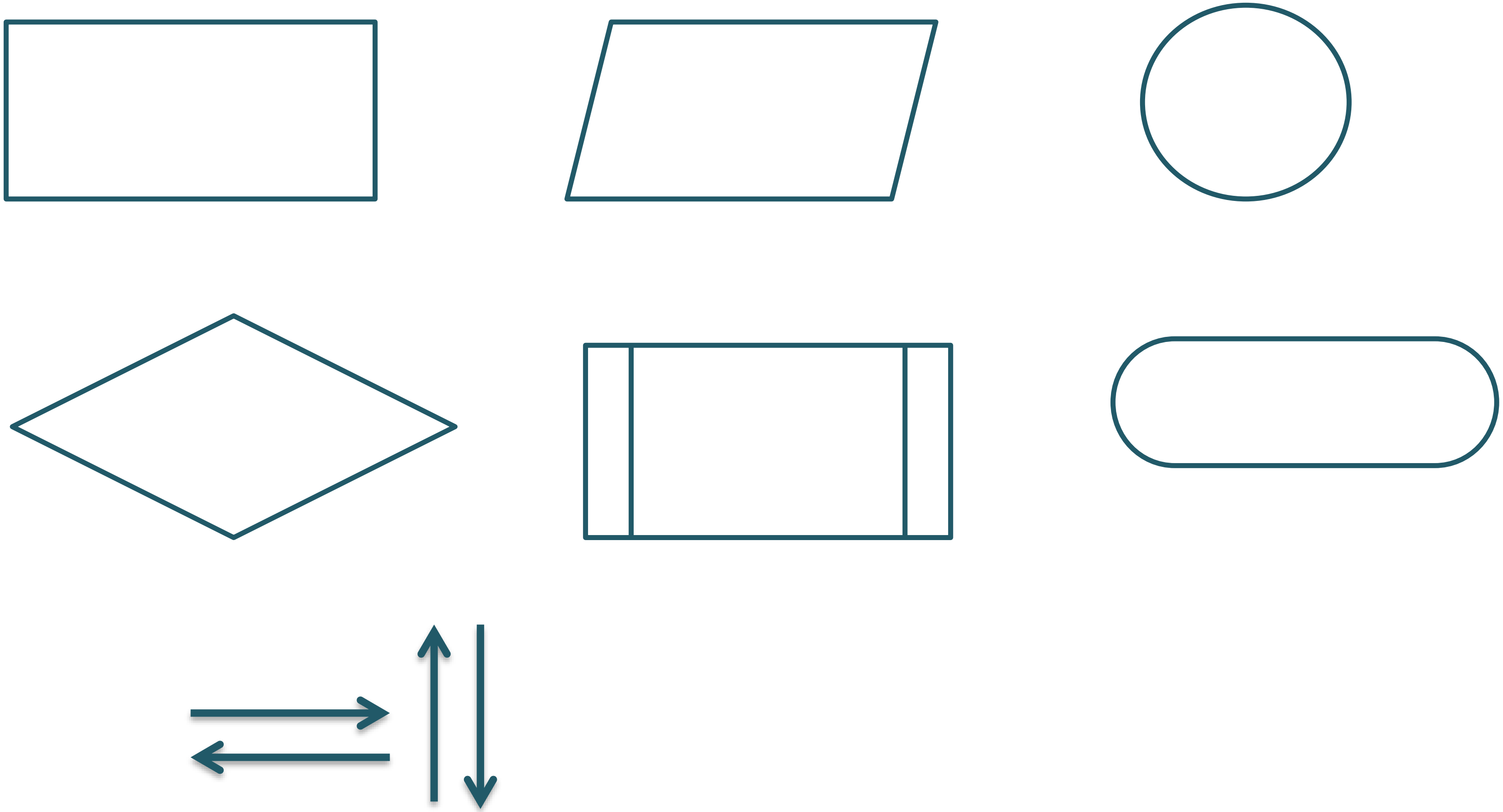
Uses of a flowchart

- To specify the method of solving a problem.
- To plan the sequence of a computer program.
- Communicate ideas, solutions.

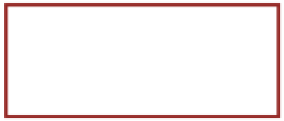
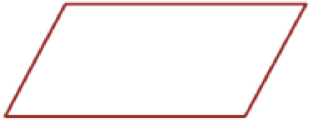
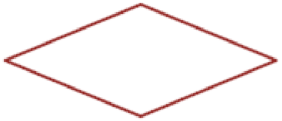
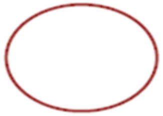
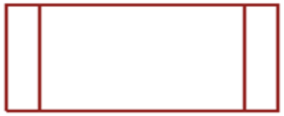
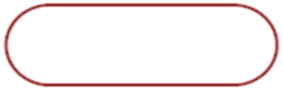
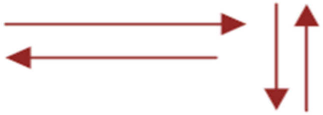
Elements of a flowchart

- Various geometrical shaped elements represent the steps of the solution.
- The elements are connected by directional arrows to show the flow of the solution.

Flowcharting Symbols

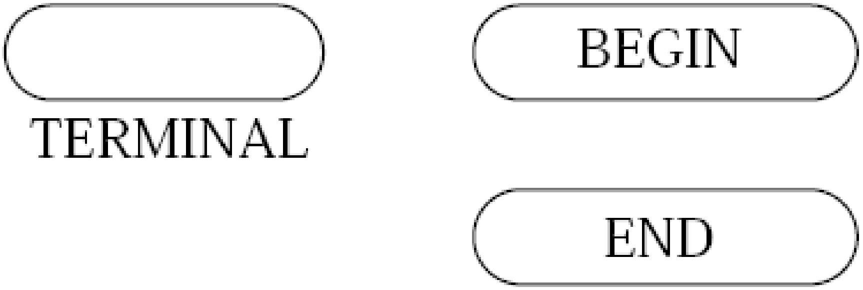


Flowcharting Symbols

Symbol	Name
	Process
	input/output
	Decision
	Connector
	Predefined Process
	Terminal
	Flow Lines

Terminal

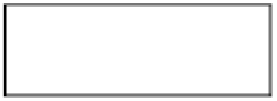
- Every flowchart has a unique starting point and an ending point.
- The flowchart begins at the start terminator and ends at the stop terminator.
- The Starting Point is indicated with the word START inside the terminator symbol.
- The Ending Point is indicated with the word STOP inside the terminator symbol.
- *There can be only one START and one STOP terminator in your entire flowchart.*
- In case a program logic involves a pause, it is also indicated with the terminal symbol.



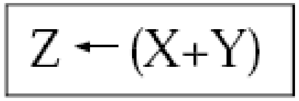
SYMBOL	EXAMPLE	
TERMINAL	BEGIN	START ACTION HERE.
	END	STOP ACTION HERE.

Processing

- A process symbol is used to represent arithmetic and data movement instructions in the flowchart.
- All arithmetic processes of addition, subtraction, multiplication and division are indicated in the process symbol.
- The logical process of data movement form one memory location to another is also represented in the process box.
- If there are more than one process instructions to be executed sequentially, they can be placed in the same process box, one below the other in the sequence in which they are to be executed.



PROCESSING

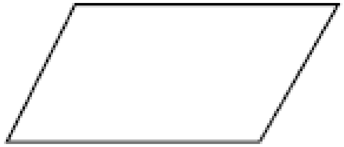


$Z \leftarrow (X+Y)$

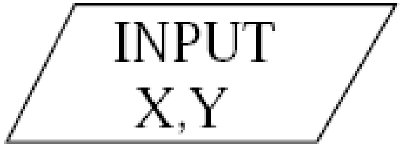
ADD THE VALUE CONTAINED
IN Y TO THE VALUE
CONTAINED IN X AND
PLACE THE RESULT IN Z

Input/output

- This symbol is used to denote any input/output function in the program.
- Thus if there is any input to the program via an input device, like a keyboard, tape, card reader etc. It will be indicated in the flowchart with the help of the Input/output symbol.
- Similarly, all output instructions, for output to devices like printers, plotters, magnetic tapes, disk, monitors etc. are indicated in the Input/output symbol.

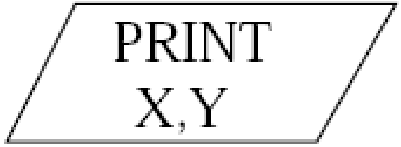


INPUT/
OUTPUT



INPUT
X,Y

TAKE TWO VALUES FROM AN EXTERNAL SOURCE AND ASSIGN THEM TO X & Y ON A TERMINAL/SCREEN

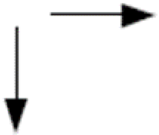


PRINT
X,Y

WRITE THE VALUES CONTAINED IN X & Y ON A TERMINAL/SCREEN

Flow lines

- Flow lines are solid lines with arrowheads are used to indicate the flow of operation, that is, the exact sequence in which the instructions are to be executed.
- The normal flow of flowchart is from top to bottom and left to right.
- However, as a good practice and in order to avoid confusion, flow lines are usually drawn with an arrowhead at the point of entry to a symbol.
- Good practice also dictates that flow lines should not cross each other and that such intersections should be avoided whenever possible.



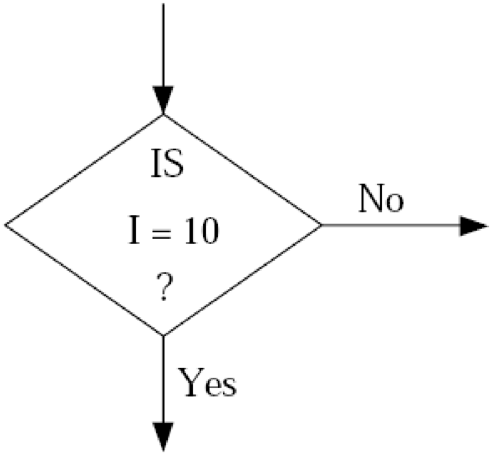
FLOW LINES

THE ARROWS INDICATE THE ROUTES FOR SYSTEMATIC SOLUTION OF THE PROBLEM.

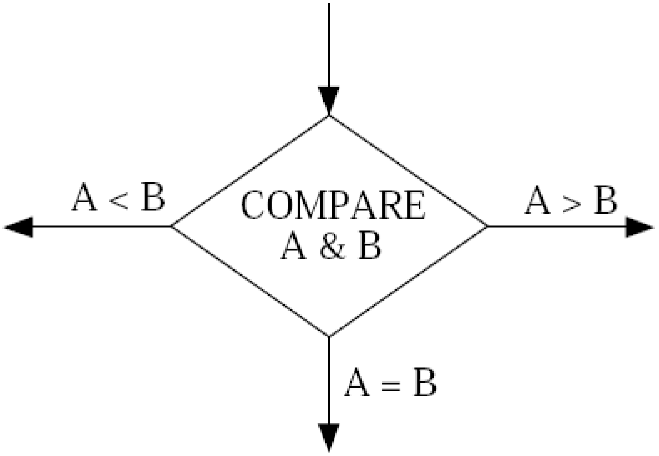
Decision

- The decision symbol is used in a flowchart to indicate a point at which a decision has to be made and a branch to one of two or more alternative points is possible.
- Moreover, the condition upon which each of the possible exit paths will be executed, should be identified and all the possible paths should be accounted for.
- During execution, the appropriate path is followed depending upon the result of the decision.

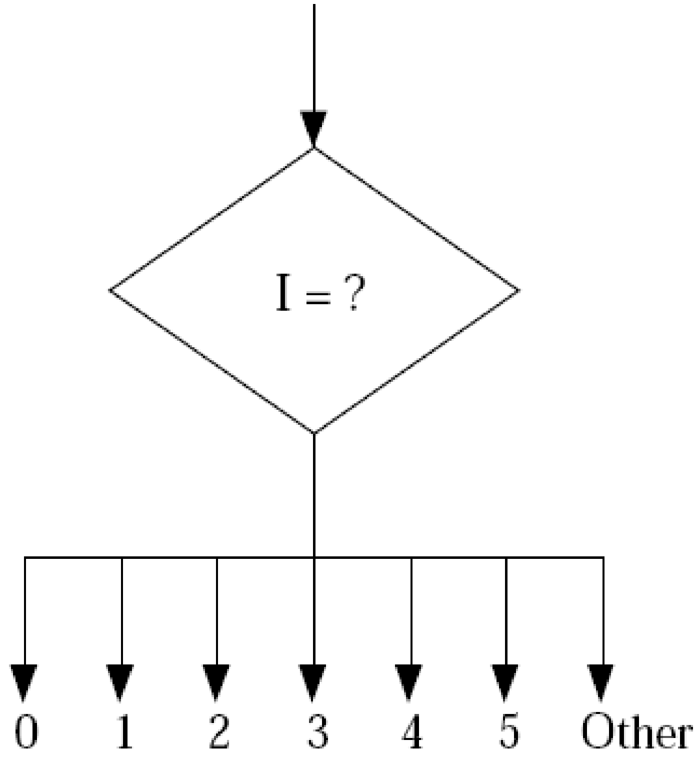
- Below figure shows three different ways in which a decision symbol can be used.



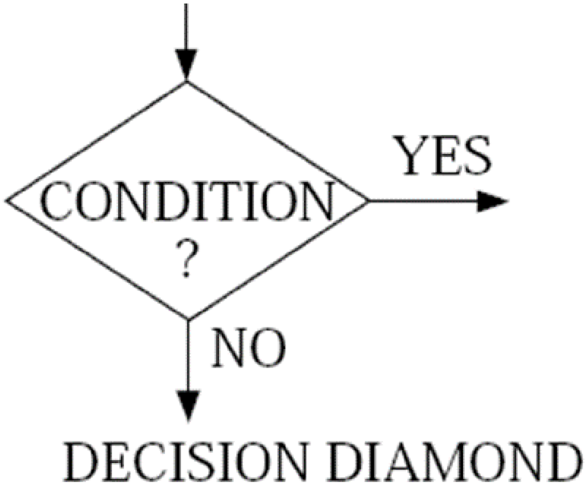
(a) Two-way branch



(b) Three-way branch



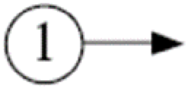
(c) Multiple-way branch



IF CONDITION IS SATISFIED THEN YES PATH IS TO BE FOLLOWED OTHERWISE NO ROUTE IS TO BE TAKEN.

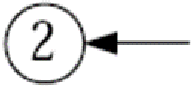
Connector

- The connector represents entry from or exit to another part of the flowchart.
- A connector symbol is indicated by a circle and a letter or a digit is placed in the circle.
- This letter or digit indicates a link.
- A pair of such identically labelled connectors are used to indicate a continued flow in situations where flowcharts are complex or spread over more than one page
- Thus a connector indicates an exit from some section in the flowchart and an entry into another section of the flowchart.



ENTRY CONNECTOR

AN ENTRY IN THE FLOW CHART
IS MADE AT THE CONNECTING
POINT MARKED ①



CONNECTOR
(TRANSFER)

A TRANSFER OF PROBLEM
SOLUTION IS MADE AT THE
CONNECTOR POINT ②
IN THE FLOW CHART.

While using the flowchart symbols following points have to be kept in mind:

- The shape of the symbol is important and must not be changed.

- The size can be changed as required.

- The symbol must be immediately recognizable.

- The details inside the symbol must be clearly legible.

- The flow lines, as far as possible must not cross.

Advantages of Flow Diagrams

- Developing the program logic and sequence
- A flowchart being a pictorial representation of a program, makes it easier for the programmer to explain the logic of the program to others rather than a program
- It shows the execution of logical steps without the syntax and language complexities of a program.
- Flowcharts provide a strong documentation in the overall documentation of the software system.
- Once the flowchart is complete, it becomes very easy for programmers to write the program from the starting point to the ending point
- A flowchart is very helpful in the process of debugging a program.

Drawing a flowchart

In developing the flowcharts following points have to be considered:

- Defining the problem.

- Identify the various steps required to form a solution.

- Determine the required input and output parameters.

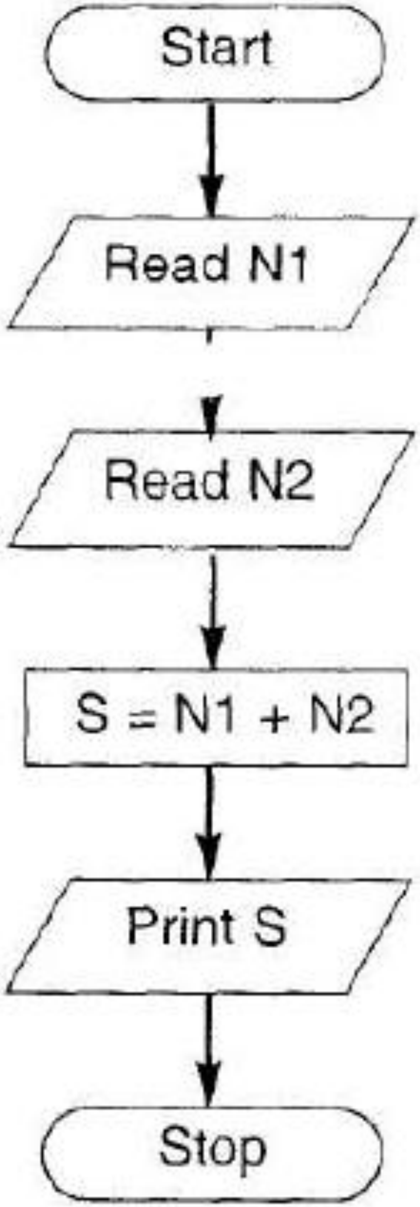
- Get expected input data values and output result.

- Determine the various computations and decisions involved.

- **Example** : Prepare a flowchart to add two numbers.

The steps are :

1. Start.
2. Get two numbers N1 and N2.
3. Add them.
4. Print the result.
5. Stop.

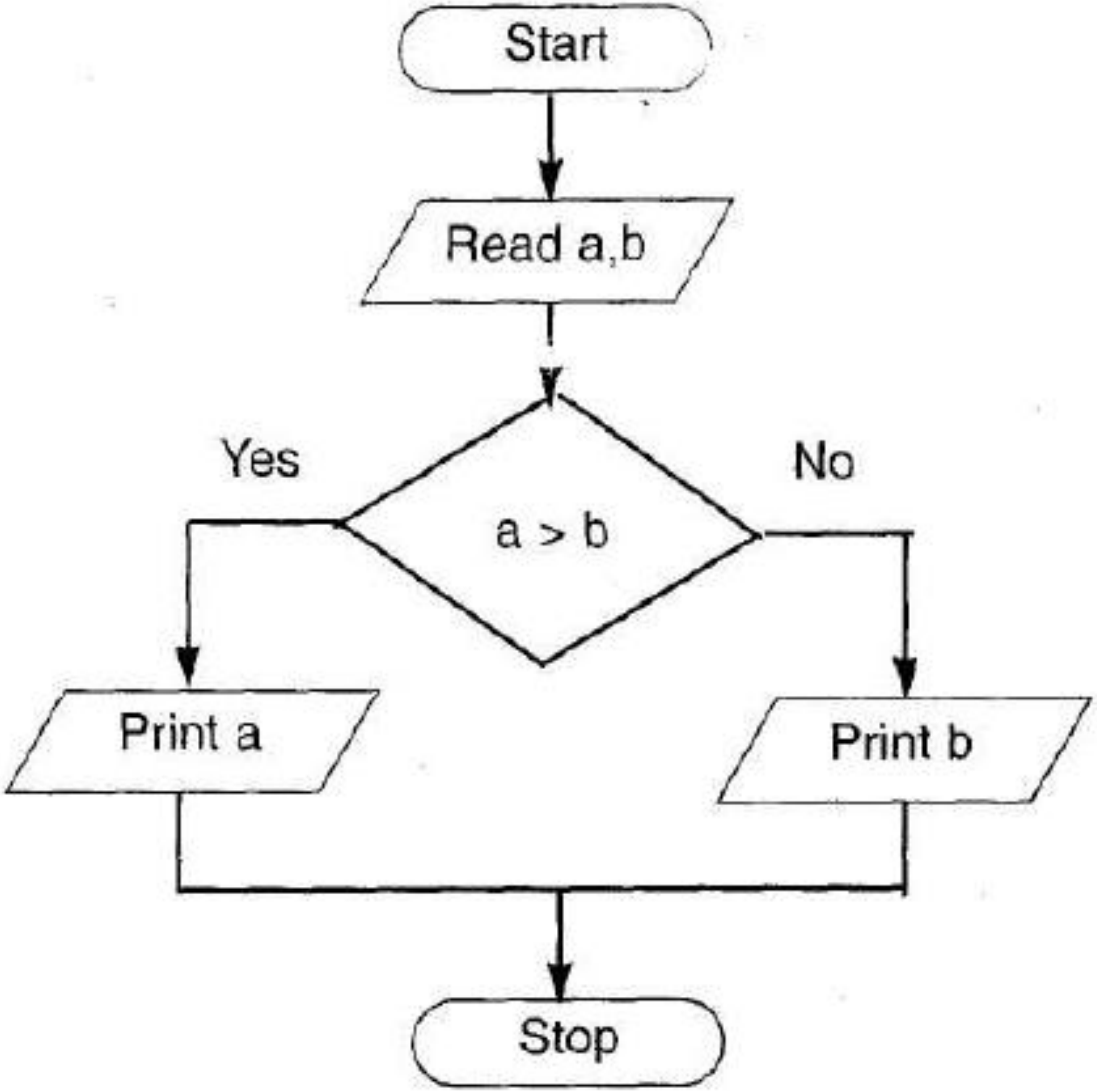

Start
Read N1
Read N2
S = N1 + N2
Print S
Stop

Example : Prepare a flowchart to determine the greatest of two numbers

To prepare a flowchart to determine the greatest of two numbers.

Here we use the decision symbol. We also combine the two reads for numbers A and B in one box. The steps are :

1. Start
2. Get two number A and B.
3. If A > B then print A else print B.
4. Stop.

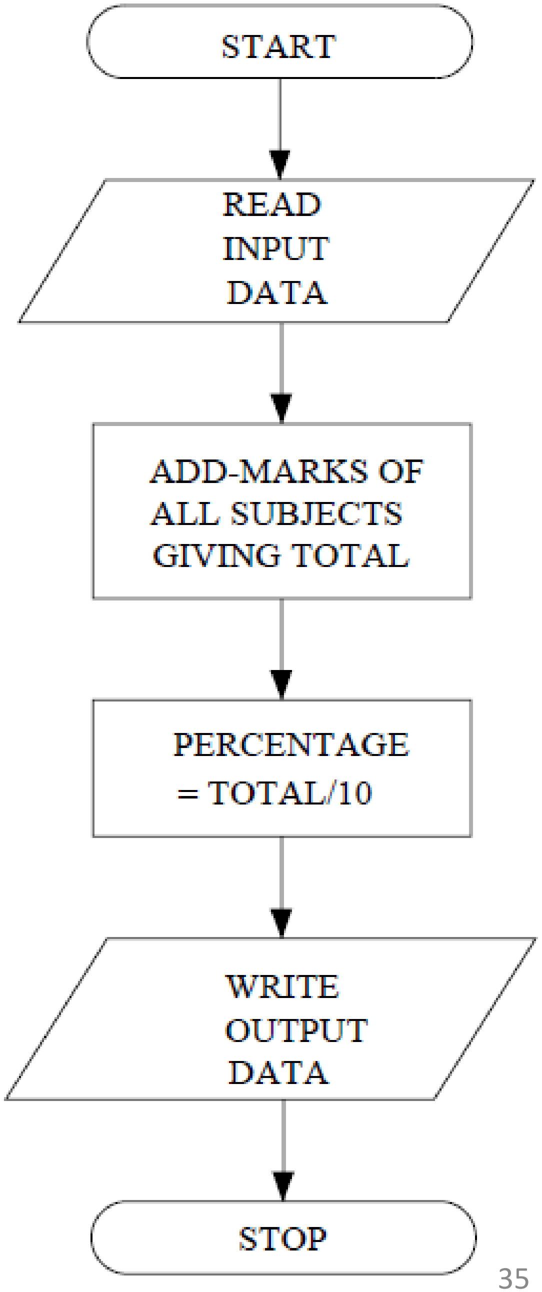

Start
Read a,b
a > b
Yes
No
Print a
Print b
Stop

- Write a flow chart to obtain the average of three numbers
- If you are given hours and rate,

  pay = hours * rate

  Write a flow chart to obtain the pay

Draw a flowchart for adding marks in ten subjects obtained by a student in an examination. The output should print the percentage of marks of the student in the examination

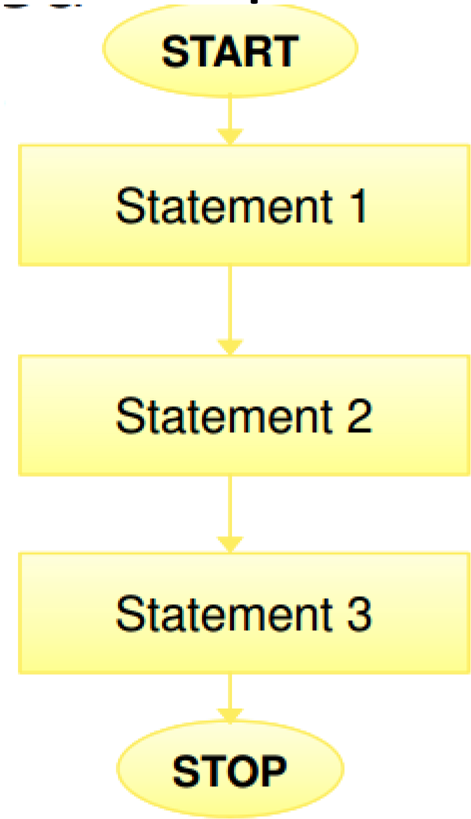


Flowcharting Techniques

- Flowcharts For Computations (Sequence)
- Flowcharts For Decision Making(Selection)
- Flowcharts For Loops (Iteration)

Flow charts for computations

- SEQUENCE is a linear progression where one task is performed sequentially after another



Flow charts for computations

The general format of the flowcharting steps for computations is :

- ✓Create variables used in calculations and read operation.
- ✓Get required data input using variables.
- ✓Perform the necessary calculations.
- ✓Print the result.

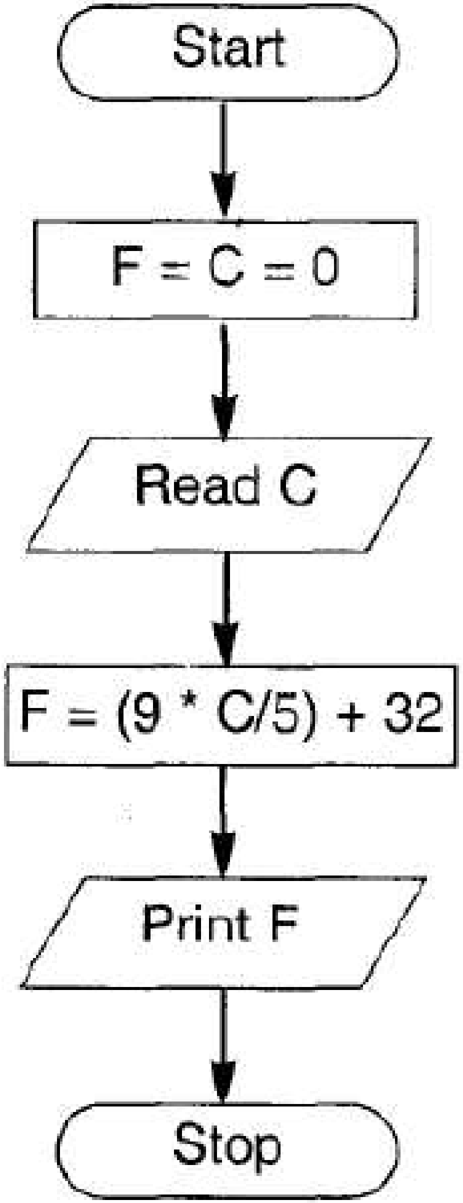
The following examples show the usage of flowcharts in computations

Example :

Flowchart for a program that converts temperature in degrees Celsius to degrees Fahrenheit

First let us write the steps involved in this computation technique.

1. Start.
2. Create variables F and C (for temperature in Fahrenheit and Celsius).
3. Read degrees Celsius into C.
4. Compute the degrees Fahrenheit into F.
5. Print result (F).
6. Stop.


Start
F = C = 0
Read C
F = (9 * C/5) + 32
Print F
Stop

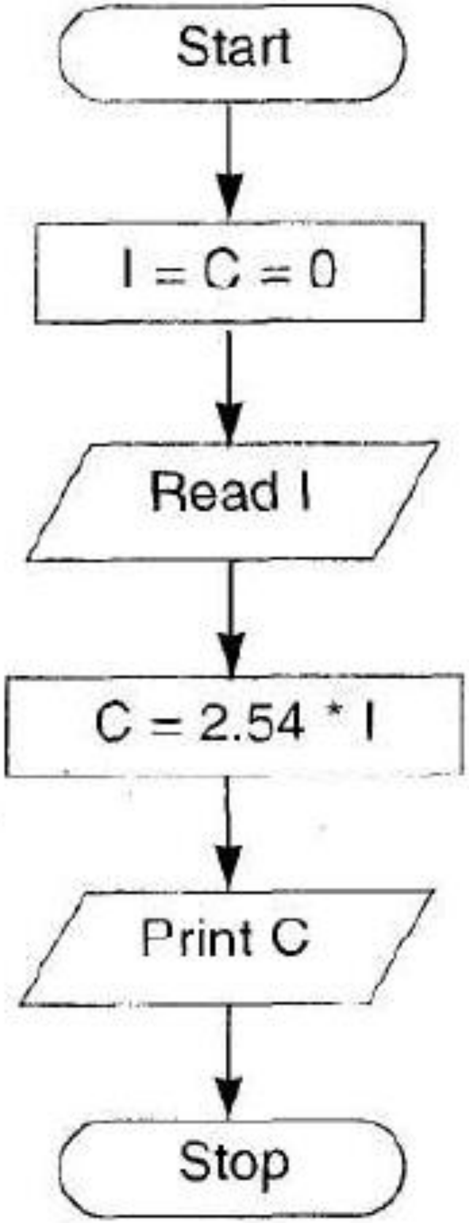
- **Example** : Flowchart for a program that converts inches to centimetres

$$F = (9 * C/5) + 32$$

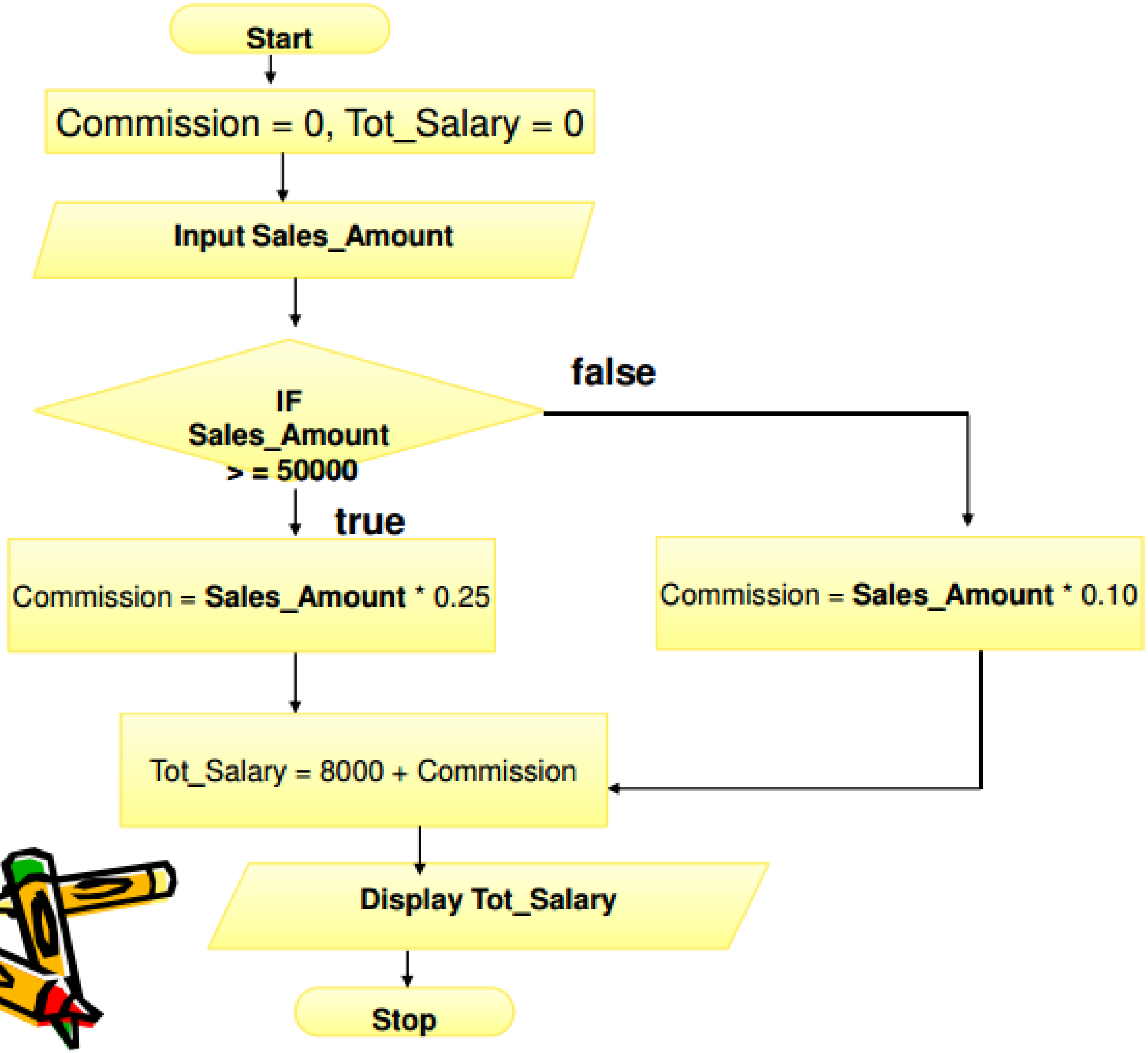
- **Example:** Flowchart to get marks for 3 subjects and declare the result. If the marks >= 35 in all the subjects the student passes else fails.

Example : Flowchart for a program that converts inches to centimetres. First let us write the steps involved in this computation technique.

1. Start.
2. Create variables C and I (for Centimetres and Inches respectively).
2. Read value of Inches into I
3. Compute the Inches into C.
4. Print result (C).
5. Stop.


Start
I = C = 0
Read I
C = 2.54 * I
Print C
Stop

- A company pays a basic salary of Rs. 8000/- to the salesmen. If a salesman does sales of Rs. 50,000/- or above, he is given a 25% commission. Otherwise only 10%. Input the sales done by a salesman and calculate his salary for the month.



Start

Commission = 0, Tot_Salary = 0

Input Sales_Amount

IF Sales_Amount > = 50000

false

true

Commission = **Sales_Amount** * 0.25

Commission = **Sales_Amount** * 0.10

Tot_Salary = 8000 + Commission

Display Tot_Salary

Stop

Flow charts for decision making

Computers are used extensively for performing various types of analysis. The decision symbol is used in flowcharts to indicate it.

The general format of steps for flowcharting is as follows:

- Perform the test of the condition.

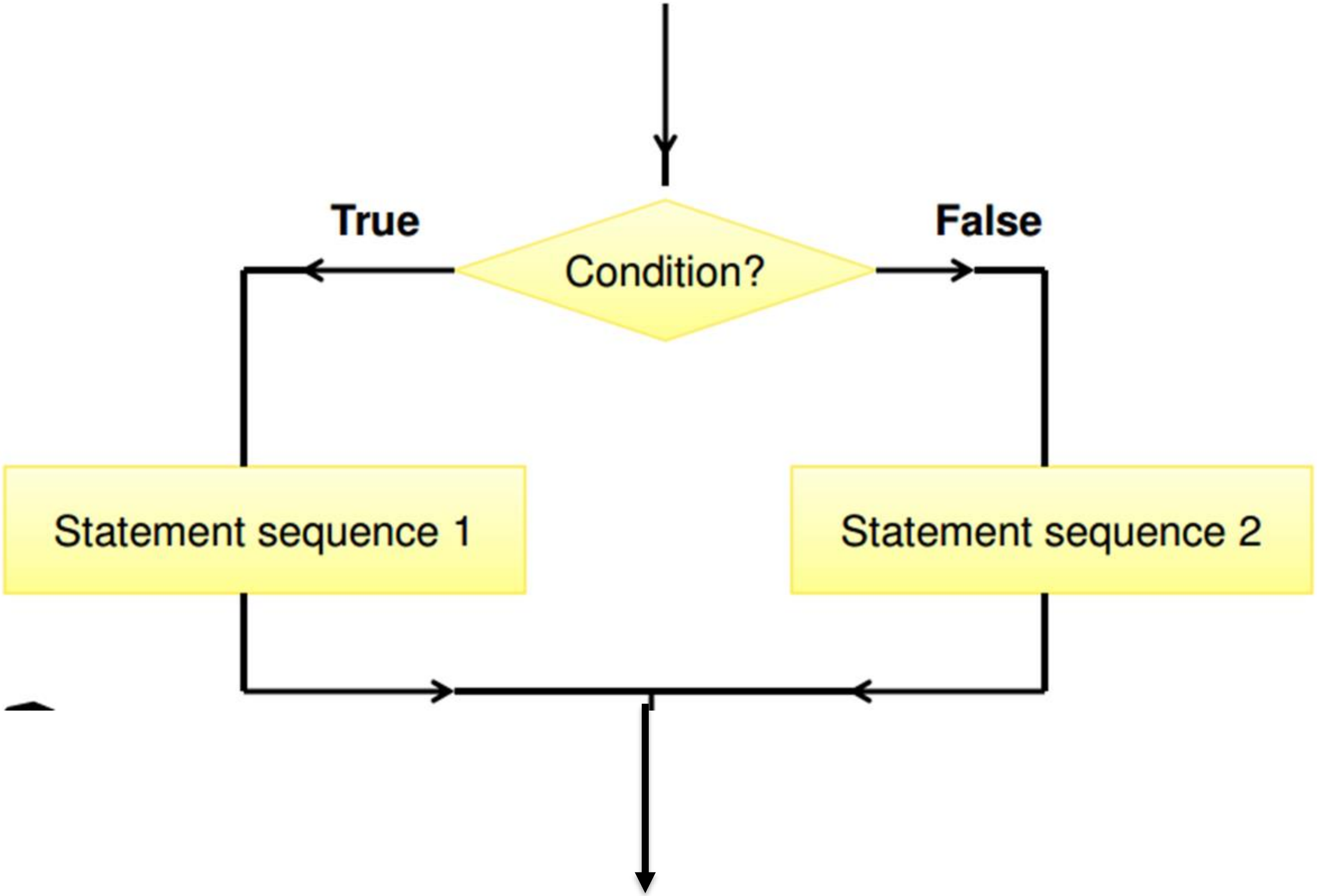
- If condition evaluates true branch to Yes steps.

- If condition evaluates false branch to No steps.

Flow charts for decision making

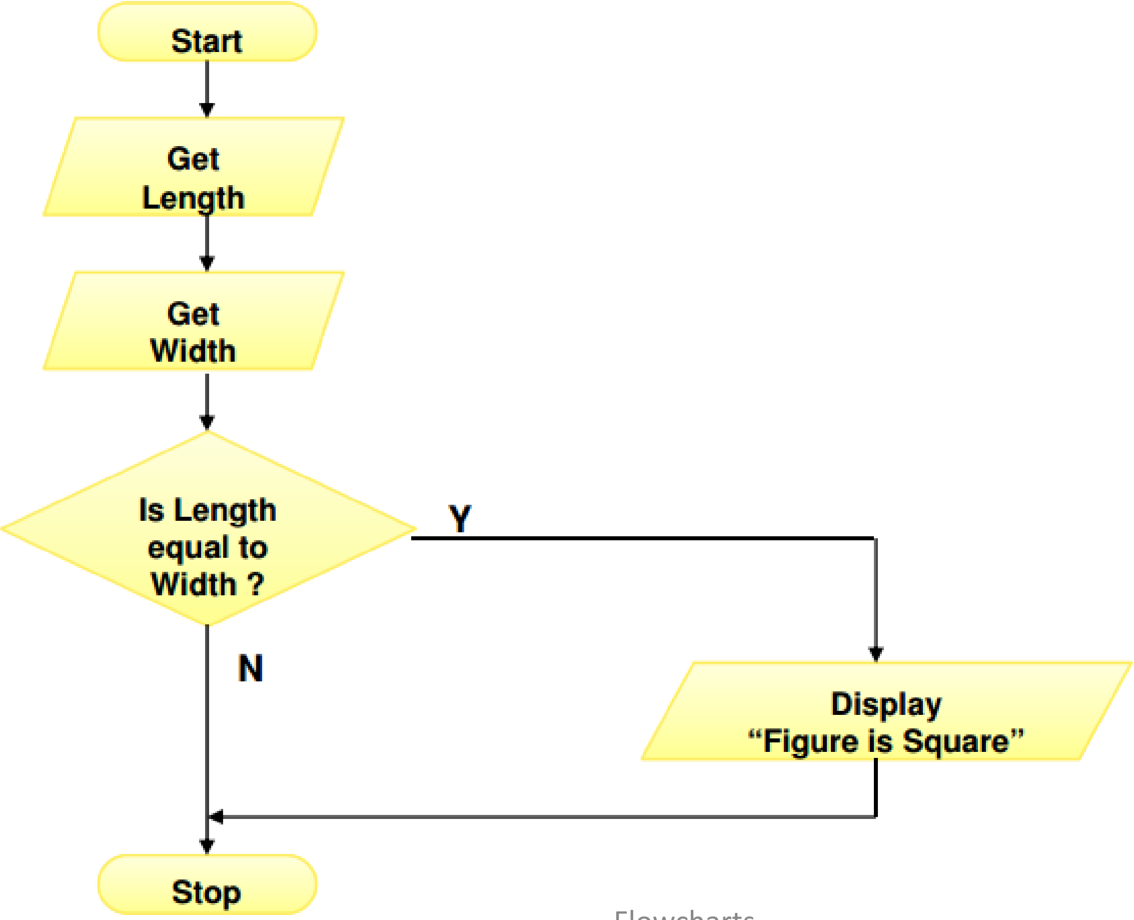
SELECTION - there may be alternative steps that could be taken subject to a particular condition

- IF-THEN-ELSE is a decision (selection) in which a choice is made between two alternative courses of action


True
Condition?
False
Statement sequence 1
Statement sequence 2

Ex:

- Input the length and width of a quadrilateral and state whether it is a square



Ex:

- Input the length and width of a quadrilateral and state whether it is a square or a rectangle.

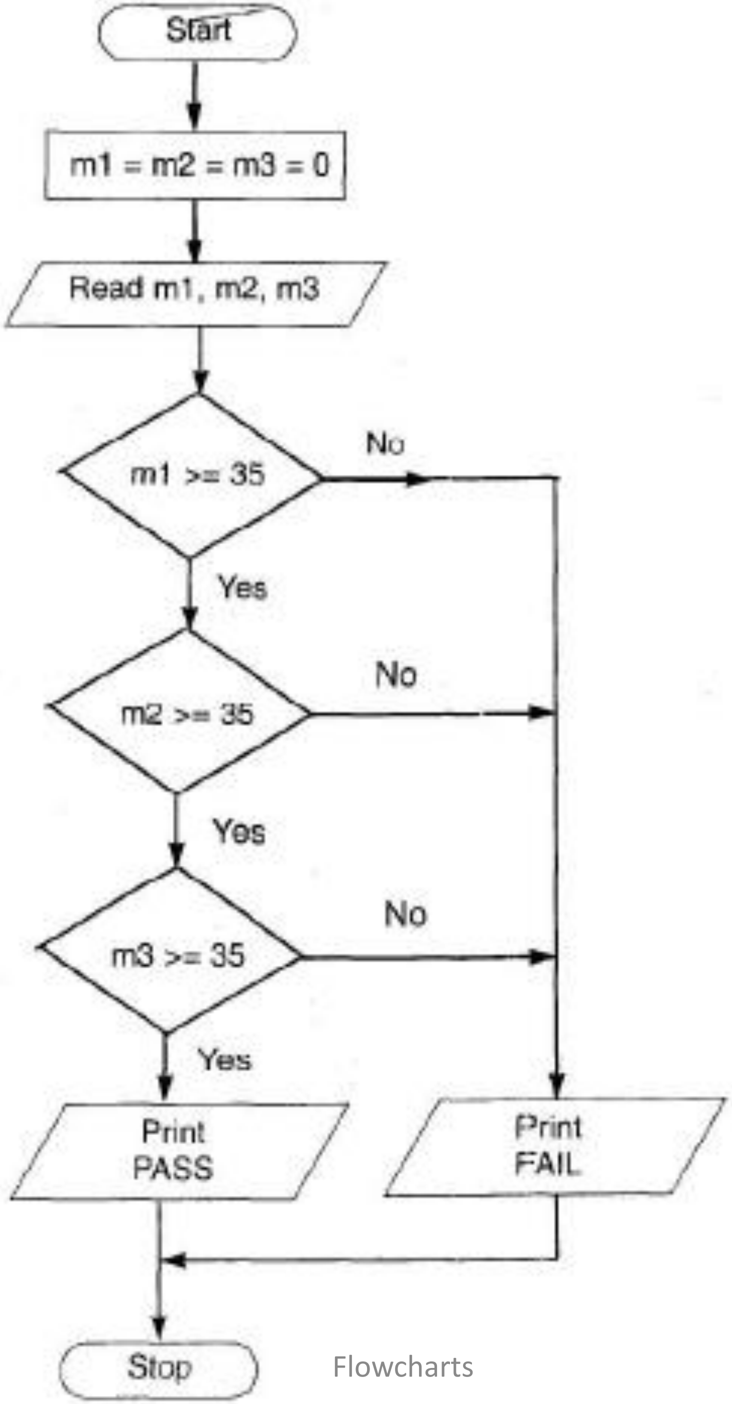
- *Note:* Most programming languages have commands for performing test and branching the exact commands and syntax depends on the language used. Some of the conditional constructs available in programming languages for implementing decision making in programs are as follows:

  - If
  - If - else – end if
  - If – else if – end if
  - Do case – end case.
  - Switch.

Example: Flowchart to get marks for 3 subjects and declare the result. If the marks >= 35 in all the subjects the student passes else fails.

The steps involved in this process are :

1. Start.
2. Create variables m1, m2, m3.
3. Read marks of three subjects m1, m2, m3.
4. If m1 >= 35 go to step 5 else go to step 7
5. If m2 >= 35 go to step 6 else go to step 7
6. If m3 >= 35 print Pass. Go to step 8
7. Print fail
8. Stop


Start
m1 = m2 = m3 = 0
Read m1, m2, m3
m1 >= 35
No
Yes
m2 >= 35
No
Yes
m3 >= 35
No
Yes
Print PASS
Print FAIL
Stop

An alternative method is the one in which you can combine all the conditions with the AND operator. The steps then would be :

1. Start
2. Create variables m1, m2, m3.
3. Read marks of three subjects into m1, m2 and m3.
4. If m1 >= 35 and m2 >= 35 and m3 >= 35 print Pass. Otherwise go to step 5.
5. Print Fail.
6. Stop

Flow charts for loops

Looping refers to the repeated use of one or more steps. i.e. the statement or block of statements within the loop are executed repeatedly.

There are two types of loops.

1. Fixed Loop - The operations are repeated a fixed number of times. In this case, the values of the variables within the loop have no effect on the number of times the loop is to be executed.
2. Variable Loop - The operations are repeated until a specific condition is met. Here , the number of times the loop is repeated can vary.

Flow charts for loops

The loop process in general includes :

- Setting and initialising a counter

- execution of operations

- testing the completion of operations

- incrementing the counter

The test could either be to determine whether the loop has executed the specified number of times, or whether a specified condition has been met.

Programming considerations :

Most of the programming languages have a number of loop constructs for efficiently handling repetitive statements in a program. These include :

- do-while loop
- while loop
- for loop
- for-next loop

In most of the looping situations, we make use of counters. In situations where the loop is to be repeated on the basis of conditions, relational operators are used to check the conditions.

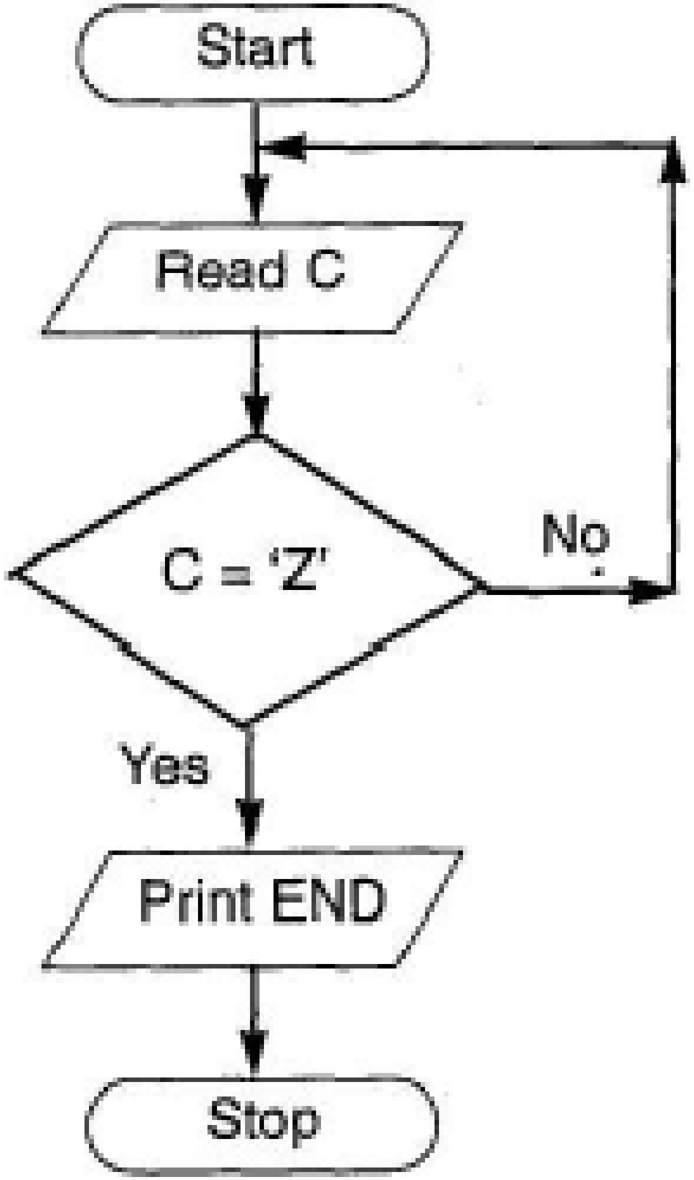
- **Example : C**heck whether character read from keyboard is Z. If it is Z then print END, else read another character.

This example shows the test which executes till a particular condition is satisfied.

The steps are :

1. Start
2. Create variable C
3. Read C,
4. Check if C = ‘Z’. If no go to step 3.
5. Print END
6. Stop

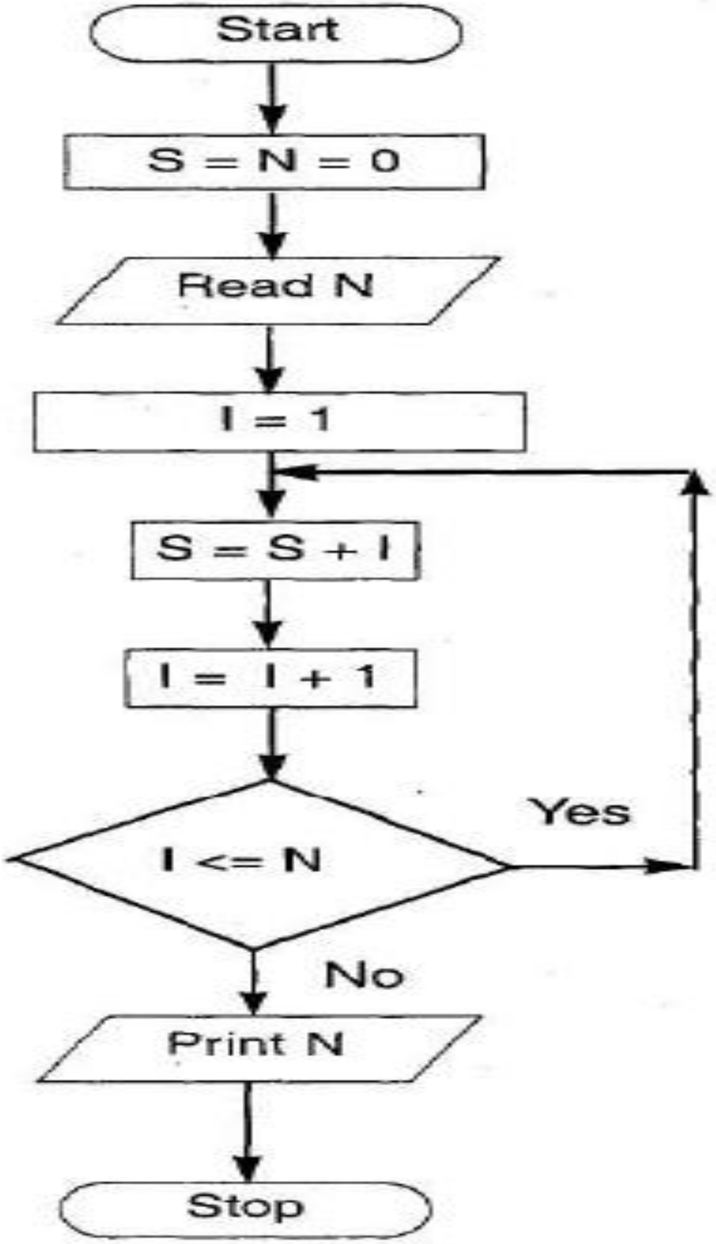
The flowchart is shown below figure


Start
Read C
C = 'Z'
No
Yes
Print END
Stop

- **Example :** To find the sum of first N numbers write a flow chart using a loop for a specific number of times.

The steps are :

1. Start
2. Create variables S , N, I
3. Read N
4. Set S (sum) to 0
5. Set counter (I) to 1.
6. S = S + I
7. Increment I by 1.
8. Check if I is less than or equal to N. If no, go to step 6.
9. Print S
10. Stop

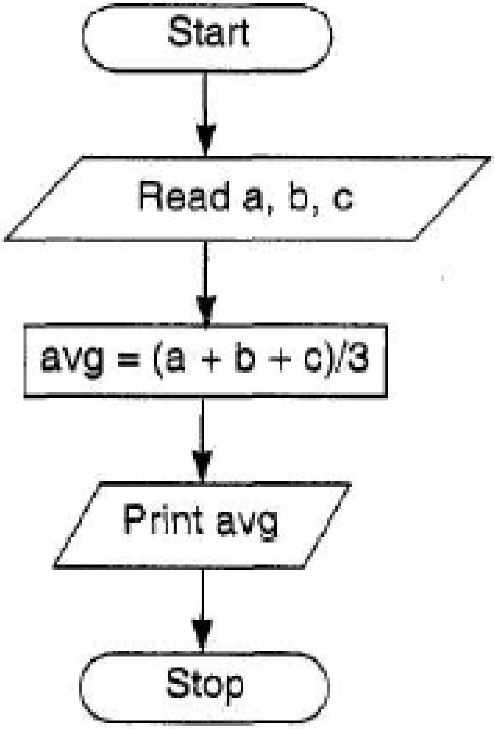

Start
S = N = 0
Read N
I = 1
S = S + I
I = I + 1
I <= N
Yes
No
Print N
Stop

Exercises

1. a) To find average of three numbers a, b ,c.

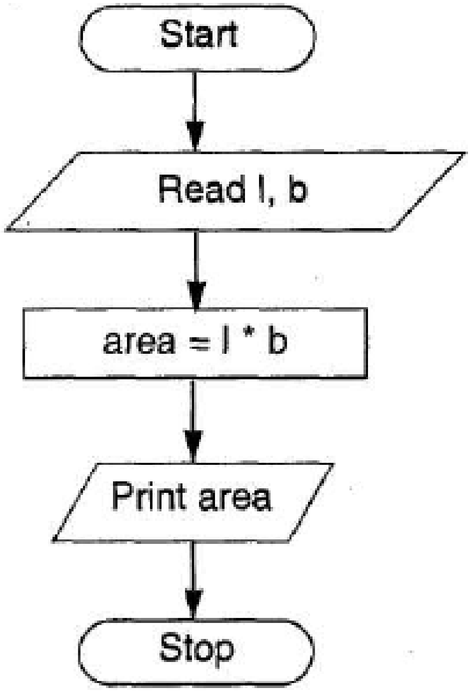
b) To find area of a rectangle whose length and breadth area read.

2. Flowchart for printing the summation of first five odd numbers



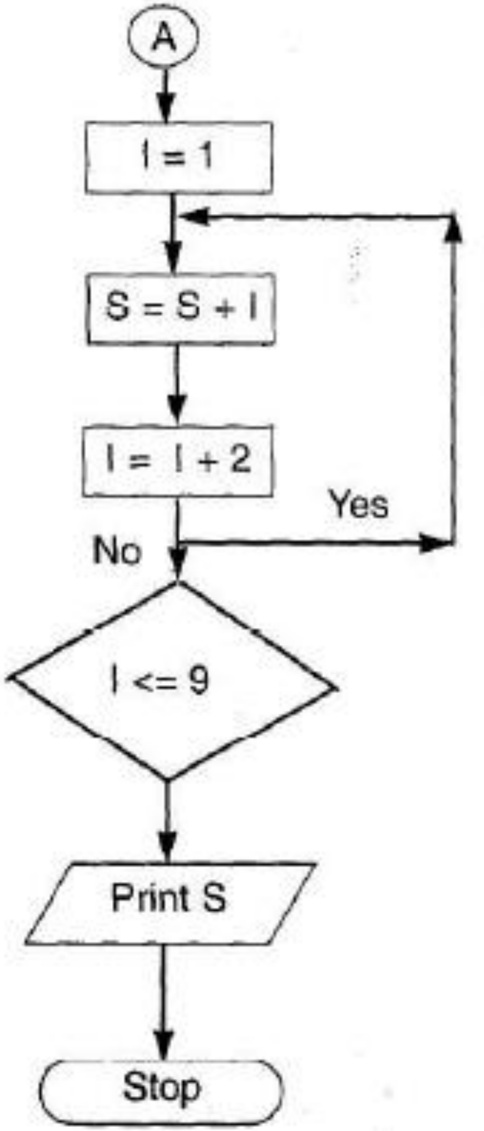
The steps are :

1. Start
2. Read numbers a, b, c
3. Compute the average as (a + b + c)/3
4. Print average
5. Stop.

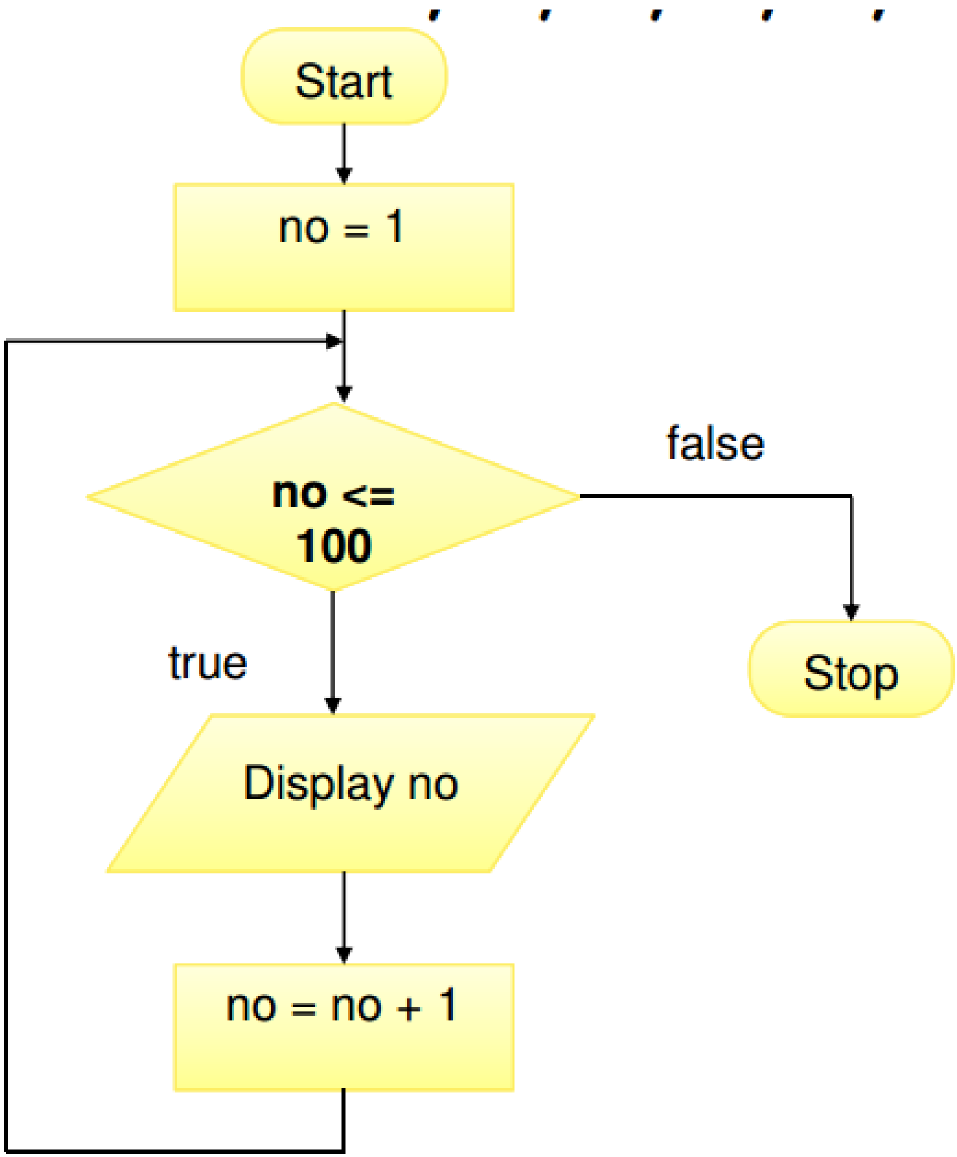


The steps are :

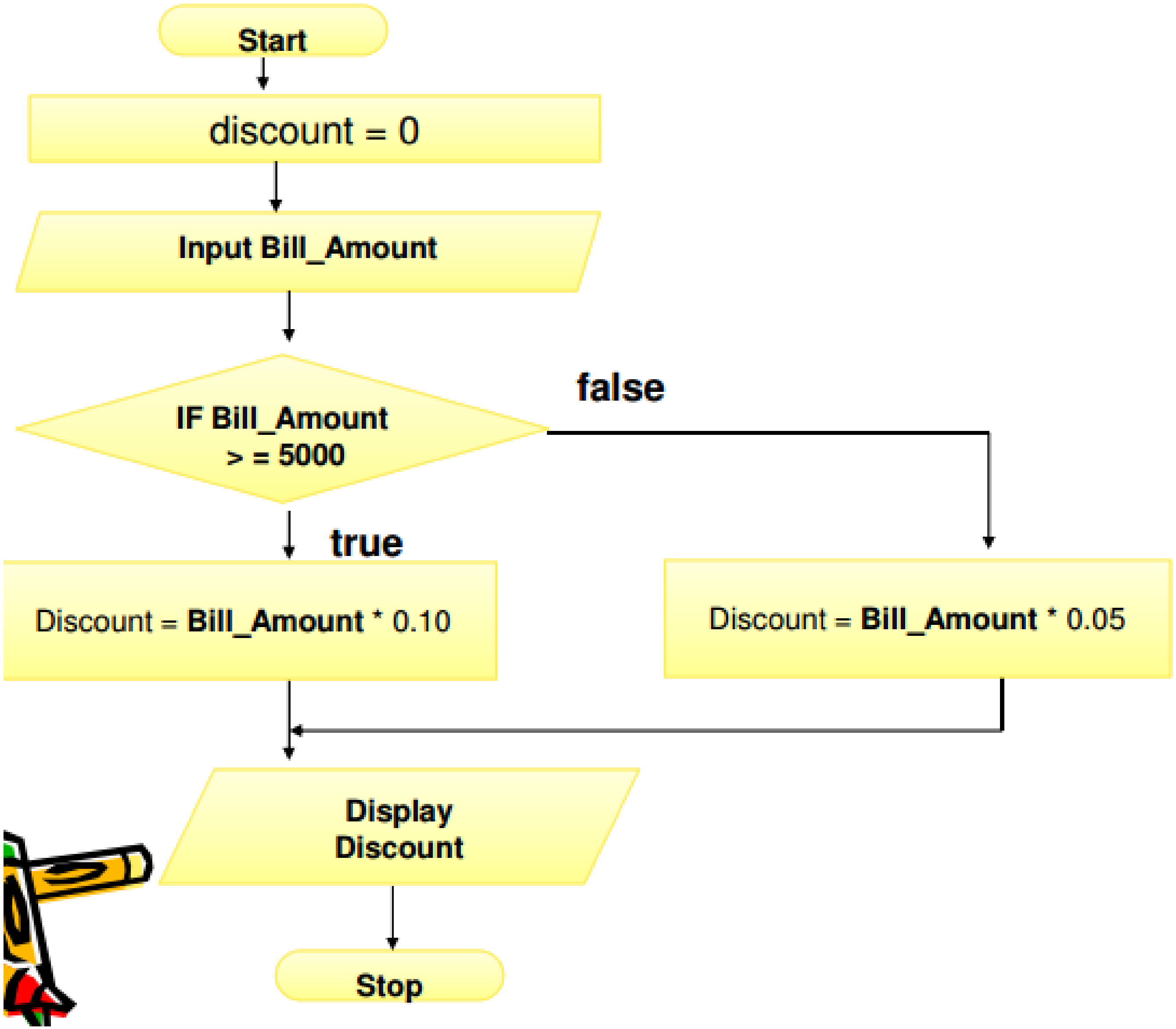
1. Start
2. Read length l and breadth b
3. Compute the area as l * b
4. Print area
5. Stop.


A
I = 1
S = S + I
I = I + 2
Yes
No
I <= 9
Print S
Stop

- Display the numbers 1, 2, 3, 4, 5, ….., 100


Start
no = 1
no <=
100
false
true
Stop
Display no
no = no + 1

- A company gives discounts for the total bill paid by the customers. If the Bill amount is above Rs. 5000/-, a discount of 10 % is given. Otherwise 5% is given. Input the Bill amount and calculate the discount amount.



Start

discount = 0

Input Bill_Amount

IF Bill_Amount > = 5000

false

true

Discount = Bill_Amount * 0.10

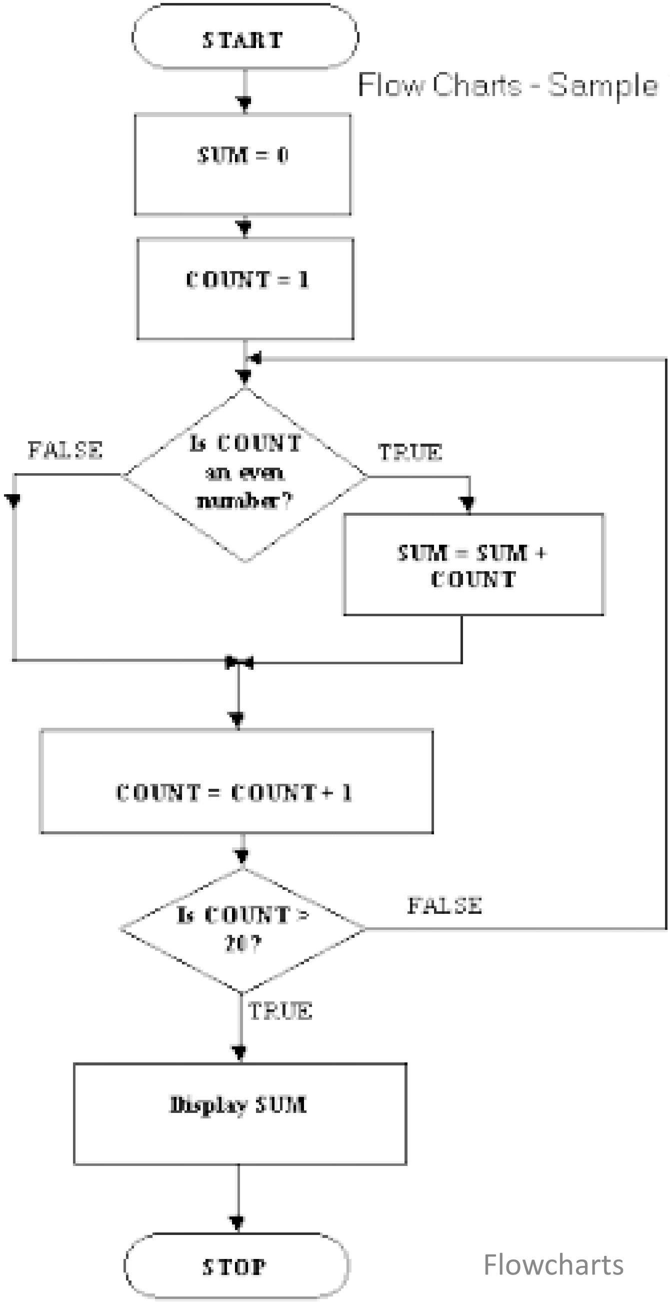
Discount = Bill_Amount * 0.05

Display Discount

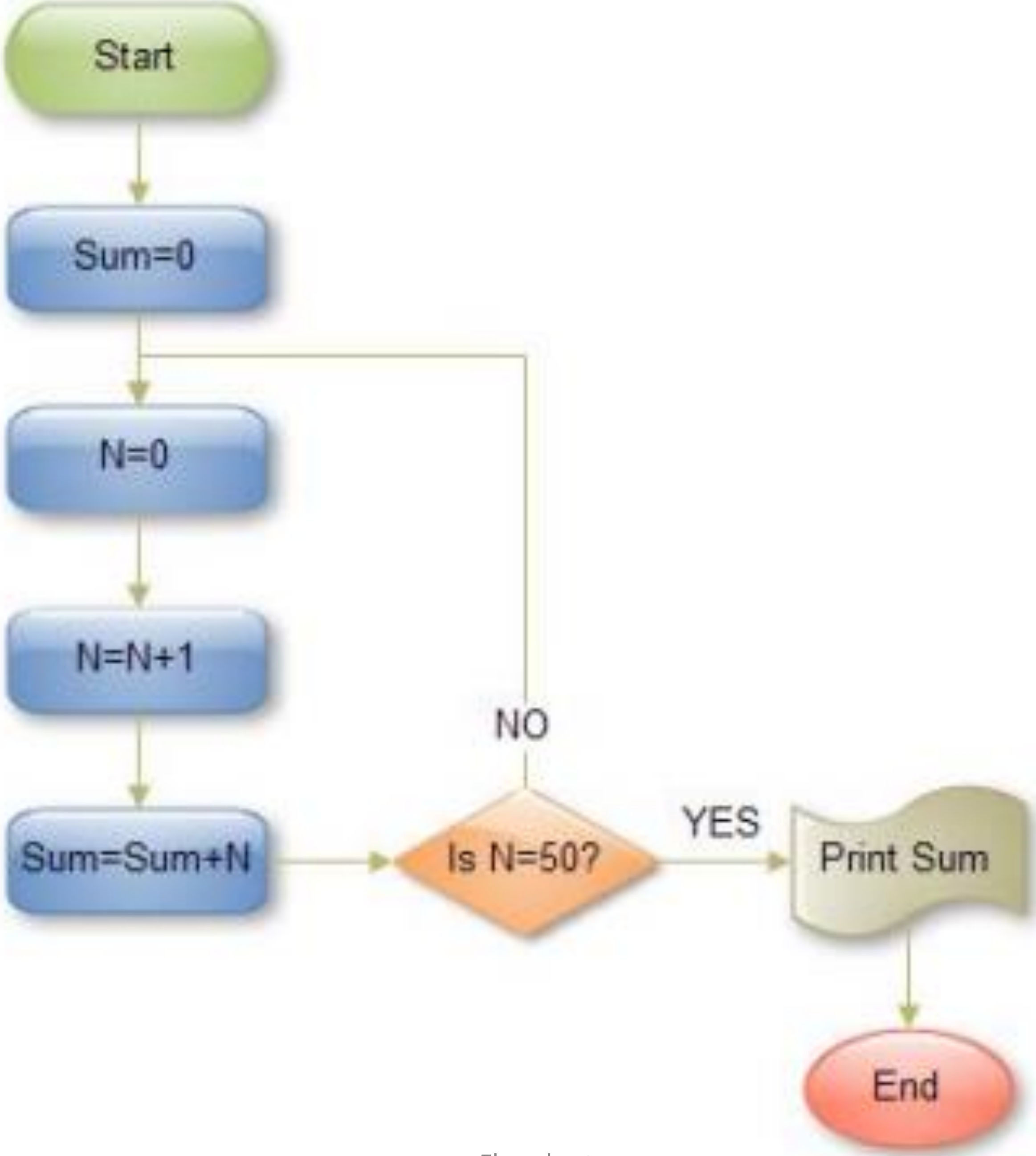
Stop

- The algorithm sums all the even numbers between 1 and 20 inclusive and then displays the sum. It uses a repeat loop and contains a null else within the repeat loop.

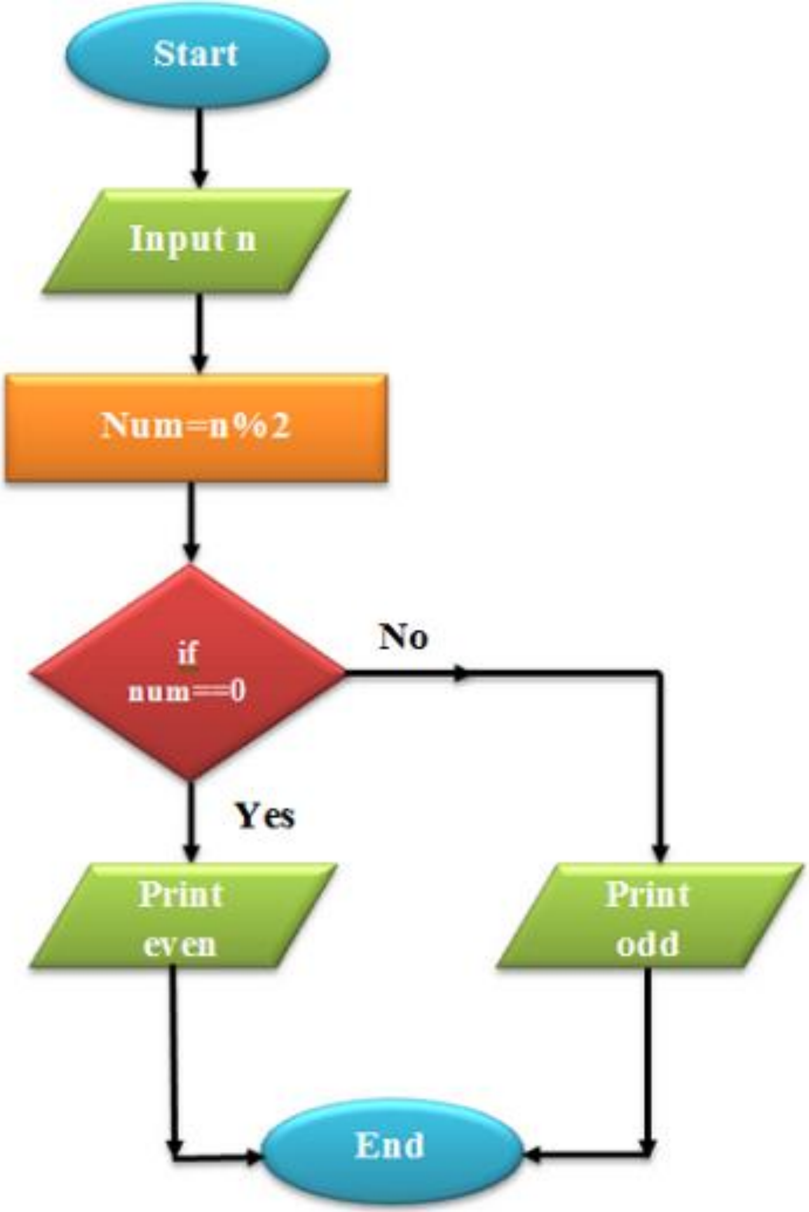
Flow Charts - Sample 1



- Draw a flowchart to find the sum of first 50 natural numbers


Start
Sum=0
N=0
N=N+1
Sum=Sum+N
Is N=50?
NO
YES
Print Sum
End

- flow chart to find whether a number is odd or even.


Start
Input n
Num=n%2
if num==0
No
Yes
Print even
Print odd
End

Thank You