Logic Building Process

Basics of Computer Language

Presented by

Day 1-:

Algorithm & Flowchart

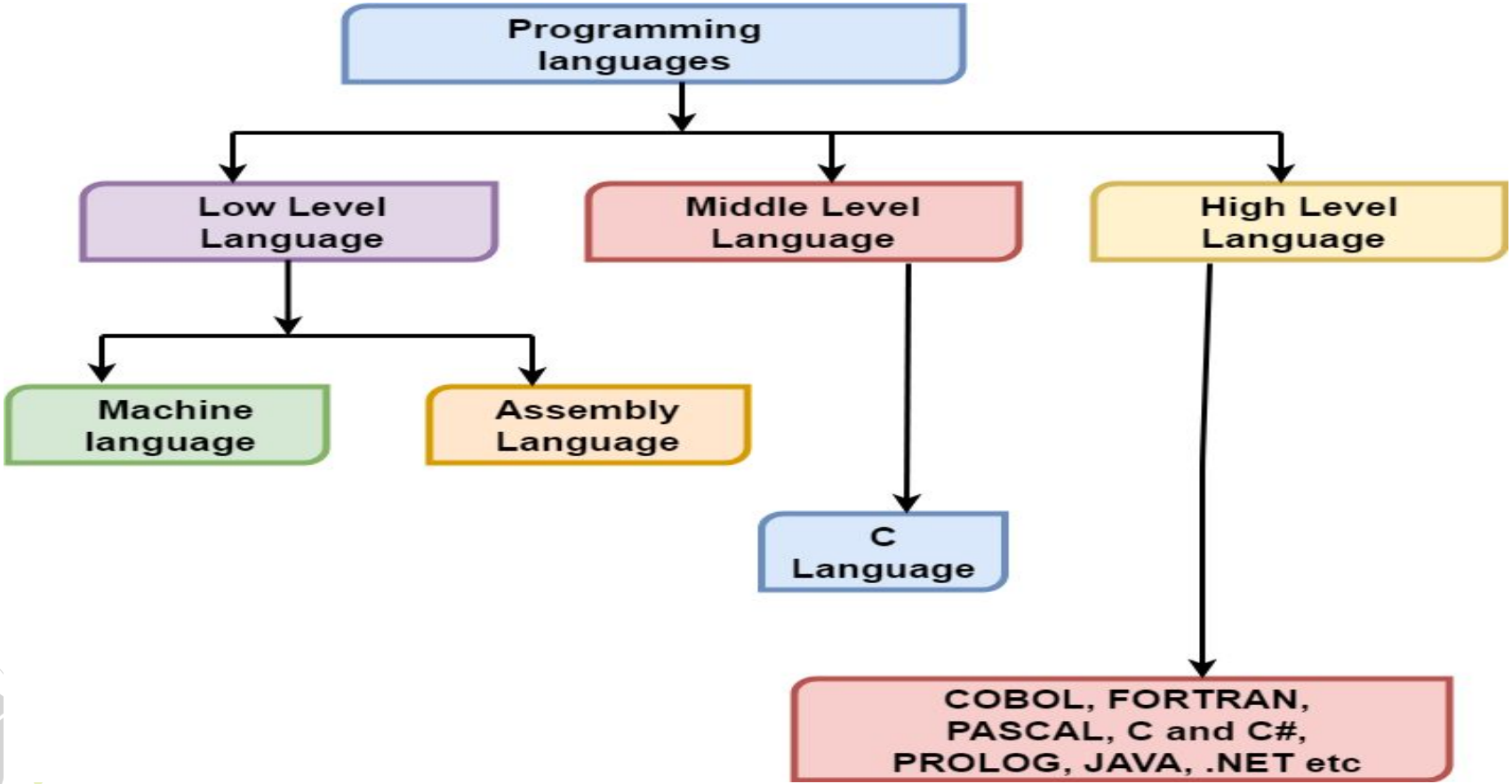
What is computer language ?

Computer is an electronic device which works on electronic signals so computer and that's the reason computer can understands electrical signals based language called binary language

Decimal number	Binary number
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Decimal	binary
2	10
+ 2	+ 10
---	---
4	100

Categories of Programming Languages



High level languages vs Low level languages

S.NO	HIGH LEVEL LANGUAGE	LOW LEVEL LANGUAGE
1.	It is programmer friendly language.	It is a machine friendly language.
2.	High level language is less memory efficient.	Low level language is high memory efficient.
3.	It is easy to understand.	It is tough to understand.
4.	It is simple to debug.	It is complex to debug comparatively.
5.	It is simple to maintain.	It is complex to maintain comparatively.
6.	It is portable.	It is non-portable.
7.	It can run on any platform.	It is machine-dependent.
8.	It needs compiler or interpreter for translation.	It needs assembler for translation.
9.	It is used widely for programming.	It is not commonly used now-a-days in programming.

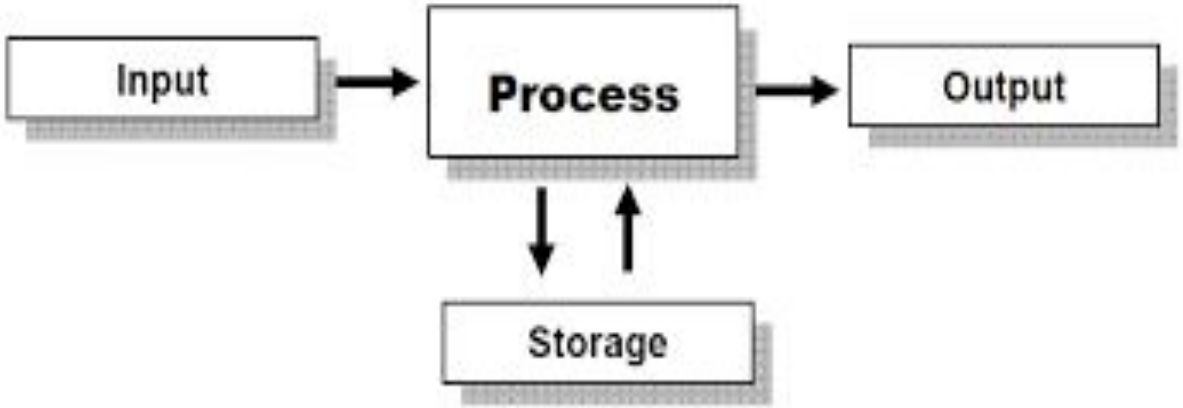
High level languages vs Low level languages

Type of Language	Example Language	Description	Example Instruction
High-level Language	Python, Visual Basic, Java, C++	Independent of hardware (portable). Translated using either a compiler or interpreter. One statement translates into many machine code instructions.	payRate = 7.38 Hours = 37.5 Salary = payRate * Hours
Low-level Language	Assembly Language	Translated using an assembler. One statement translates into one machine code instruction.	LDA181 ADD93 STO185
	Machine Code	Executable binary code produced either by a compiler, interpreter or assembler.	10101000110101010100100101010101

Machine Level Language vs Assembly Level Language

Parameters	Machine Level Language	Assembly Level Language
Hierarchy Level	It is at the lowest level in the hierarchy and has zero abstraction level from the hardware.	It is above the machine level language in the hierarchy and so has less abstraction level from the hardware.
Learning Curve	It is hard to understand by Humans.	It is easy to learn and maintain.
Written as	It is written in binary that is 0 or 1.	It is written in simple English and is easy to understand.
Generation	It is a first-generation programming language.	It is the second-generation programming language.
Requirement for Translator/Assembler	The machine code is executed directly so no translator is required.	It requires an assembler to convert assembly language to machine code.

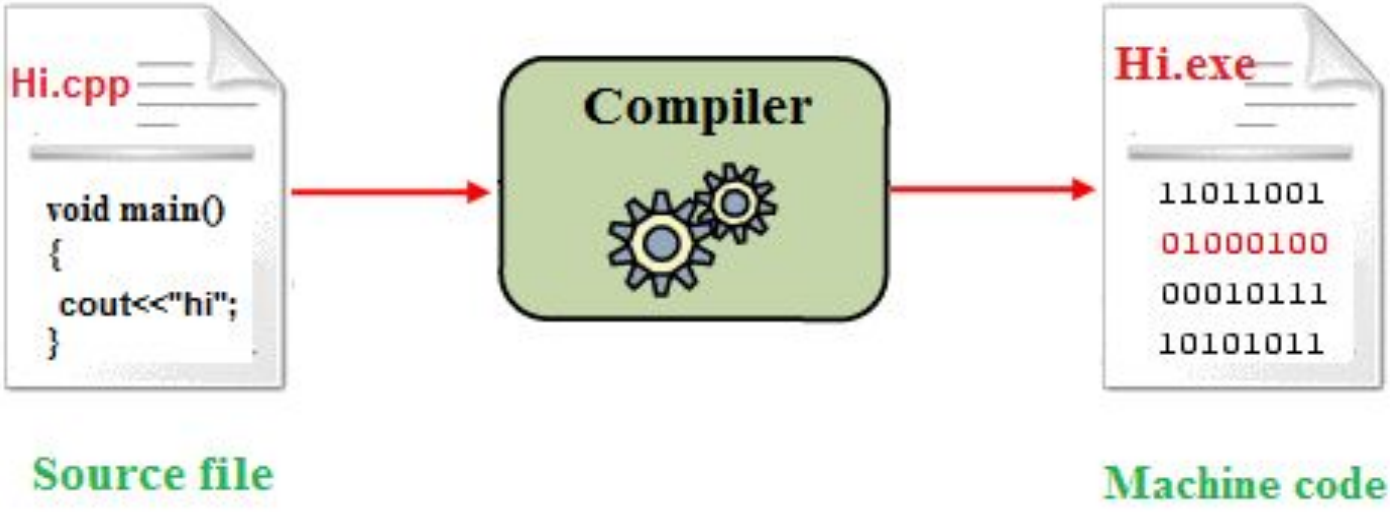
Computer Instructions components



Programming Language

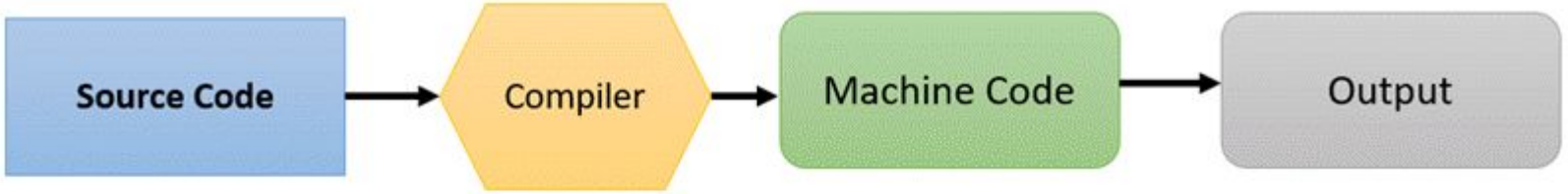
→ To give an instruction to computer to produce some out come we need programming language.

→ A programming language is a computer language that is used by programmers (developers) to communicate with computers. It is a set of instructions written in any specific language (C, C++, Java, Python) to perform a specific task



Computer process

How Compiler Works



How Interpreter Works

Compiler vs Interpreter

Sr No	Compiler	Interpreter
1	Compiler Takes **Entire** program as input	Interpreter Takes **Single** instruction as input .
2	Intermediate Object Code is **Generated**	**No** Intermediate Object Code is **Generated**
3	Conditional Control Statements are Executes **faster**	Conditional Control Statements are Executes **slower**
4	**Memory Requirement** : **More** (Since Object Code is Generated)	**Memory Requirement** is **Less**
5	Program need not be **compiled** every time	Every time higher level program is converted into lower level program
6	**Errors** are displayed after **entire program** is checked	**Errors** are displayed for **every instruction** interpreted (if any)
7	**Example** : C Compiler	**Example** : BASIC

What is instructions in computer ?

→ Computer can performs logical and arithmetic operations.

→ So whatever we want from the computer we have to encode in form of Logical or Arithmetic based statements

→ The instruction are given order to the process for some output.

→ Set of instructions known as computer **Program**

What is Program ?

→ Program is set of instructions written in computer language to solve business solutions.

→ Instructions are set of operations created using programming language syntax and business logic.

→ To ask computer to solve real world problems we need to write instructions in nitty gritty form to do this we can take help of reasoning and logic building skills.

Logic Building

Is a process to understand and solve real life business problems.

To find optimal solution for business problem we can take help of

- Algorithm
- Flowchart

Algorithm vs Program

1. Design
2. Domain knowledge
3. Any Language
4. Hardware & Software Independent
5. Analysis

1. Implementation
2. Programmer
3. Programming Languages
4. Hardware & Software dependent
5. Testing

What is Algorithm ?

Algorithm is step by step solution written in human readable format it always follow given steps.

1) Start : indicates problem solution process starts
2) Declare : instruct computer to allocate memory for some processing data
3) Input : accept values from user
4) Process : perform logic and arithmetic operations on data
5) Output : process output
6) Stop : stop process

Problem statement : write an algorithm to find simple interest

1) Start
2) Declare
3) Input
4) Process
5) Output
6) stop

1) Start
2) Declare amount,rate,year,si as float # variable
3) Input amount,rate,year
4) si <- amount*rate*year/100
5) print(si)
6) stop

Variables & Datatypes

Variable : - used to store some value into computer memory

eg:- amount,rate,year,si are variable

Datatype :-

→ It instruct computer to to allocate memory for given data format

→ **It tells computer about data format and respected format memory requirements .**

→ **Datatype have two category**

- Numeric datatype : integer,float,boolean
- Character based datatype : string

eg:- Float is datatype which stores decimal points values.

Variable and Data type

Variable can have name,data type and value

- Input variable x=10 # x is name 10 is value and data type is integer
- Output variable # used to store some processed output
- Constants # value can not change once we assign like
  PI=3.14

Example -:

Formula : PI x (rxr)

1) Start
2) Declare r,area as float # variable declare
3) PI = 3.14 # constant variable declaration
4) Input ← r # input variable values
5) area = PI * (r*r) # process and store output into output variable
6) Print area # prints output
7) Stop # stop

Statements syntax

- Statements can have arithmetic and logical process for that we required Operators for Arithmetic Operations we have operators like

Operators	Meaning	Example	Result
+	Addition	4+2	6
-	Subtraction	4-2	2
*	Multiplication	4*2	8
/	Division	4/2	2
%	Modulus operator to get remainder in integer division	5%2	1

Arithmetic statement
S = A + B

A is operand
+ is Operator
B is Operand

Type of executable statements

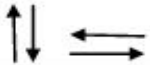
1) Sequential statements
2) Conditional or decision statements
3) Iterative or looping statements.

Before we move over other executable statements let's have a look at **FlowCharts**

What is Flow Chart ?

→ Flow chart is pictorial representation for problem statements .
→ It uses symbols to represent executable process and statements
→ It also give clear vision to problem solutions.

Flowchart symbols

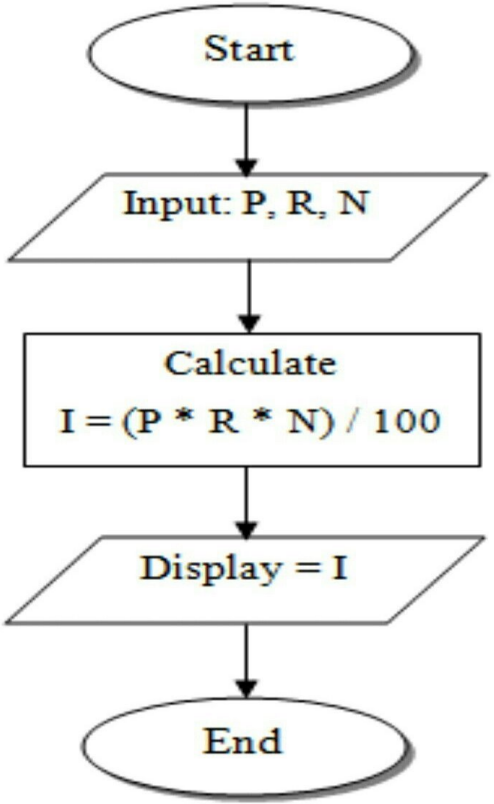
Symbol	Name	Function
	Process	Indicates any type of internal operation inside the Processor or Memory
	input/output	Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results
	Decision	Used to ask a question that can be answered in a binary format (Yes/No, True/False)
	Connector	Allows the flowchart to be drawn without intersecting lines or without a reverse flow.
	Predefined Process	Used to invoke a subroutine or an Interrupt program.
	Terminal	Indicates the starting or ending of the program, process, or interrupt program
	Flow Lines	Shows direction of flow.

Tools for Flowchart making

https://www.smartdraw.com/flowchart/flowchart-maker.htm

We can also use paint as a beginners

Write a flowchart to find simple interest



Note : perform all sequential algorithm with flowchart

Exercise (sequential statements)

1) Calculate fahrenheit to celsius
2) Calculate compound interest
3) Calculate BMI
4) Convert meter to centimeter
5) Calculate tax on salary
6) Calculate area of triangle
7) Calculate area of rectangle
8) Calculate square of user entered number
9) Calculate cube of user entered number
10) Calculate gross salary

Day 2-:

Conditional Statements

Decision Making statements

→ Statements which perform decision making on based of some conditions known as decision making statements let's have a look

→ We have two value 2 and 4 i want to get value which is maximum out of two

→ So i have to perform some condition like greater than ,less than etc....

→ To perform condition or to check relativity between two values we have relational or conditional operators which returns True or False using those operators executable statements will make decision

Comparison Operators

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
! =	Not equal to	5! =2	True

Logical Operator

To collaborate more than one decision making process we use logical operators

I,e

1. **And**
2. **Or**
3. **Not**

And operator Evaluates True when Every Condition is Evaluated as True.

Expression	Evaluates to
True and True	True
True and False	False
False and True	False
False and False	False

Or operator Evaluates True when Any one Condition is Evaluated as True.

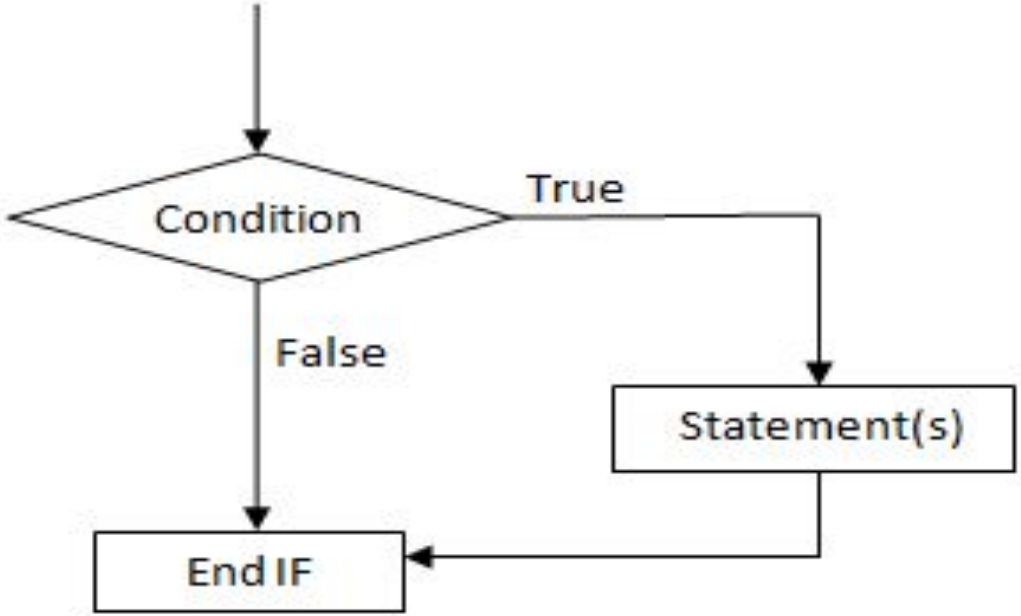
Expression	Evaluates to
True or True	True
True or False	True
False or True	True
False or False	False

Type of IF Condition

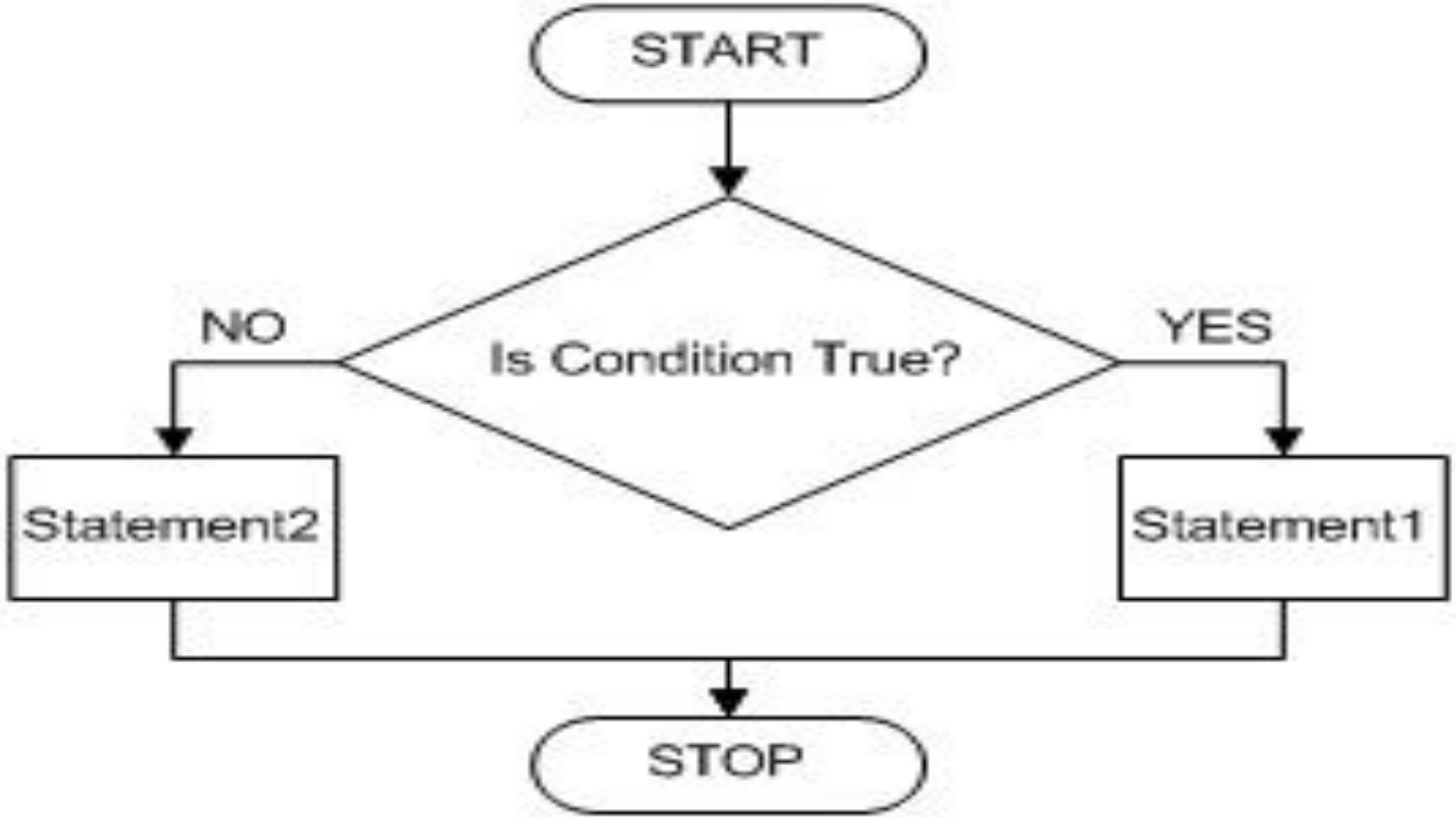
1) IF
2) IF ELSE
3) Nested IF
4) ELSE IF ladder

Explain each with flowchart

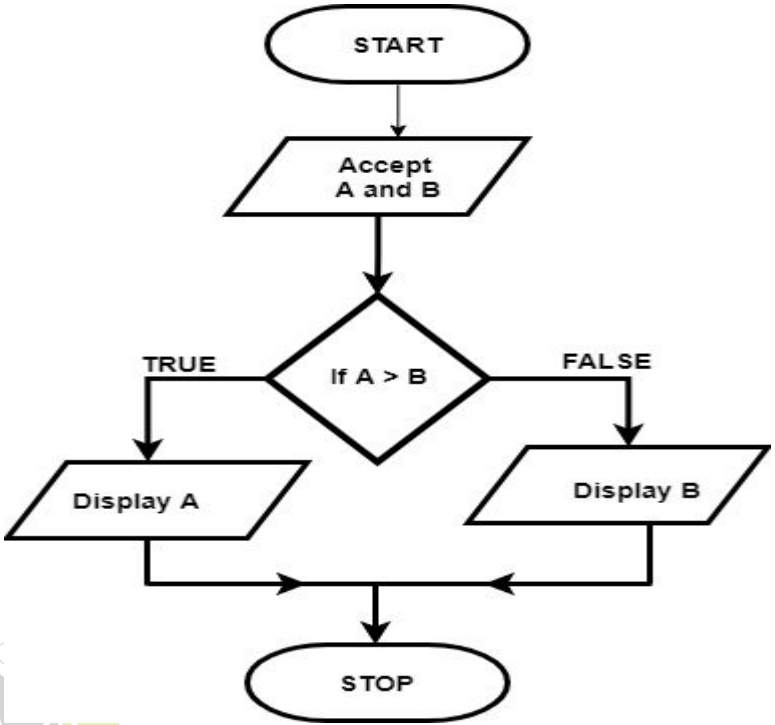
1. IF condition



2. IF ELSE Condition



Write a flowchart to find maximum out of two variable

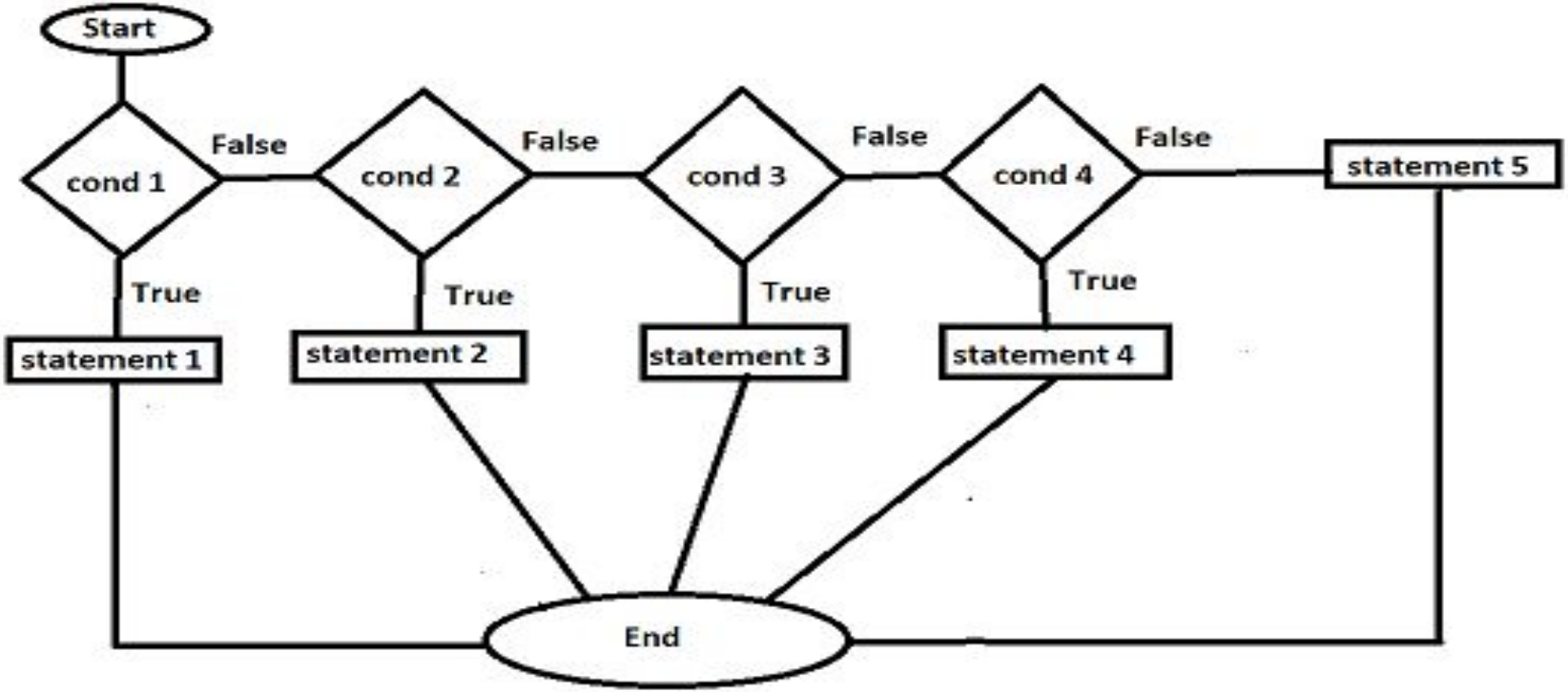


IF condition used to make decision

1) Start
2) Let A <- input
3) Let B <- input

4) If A > B then
   Print A is greater
   Else
   Print B is greater
1) stop

3. ELSE IF Ladder



Flowchart to given a marks print grade

Constraints -:

- 75 & above -: A grade
- 60-74 -: B grade
- 35-59 -: C grade
- Below 35 -: Fail

Let mark <- input

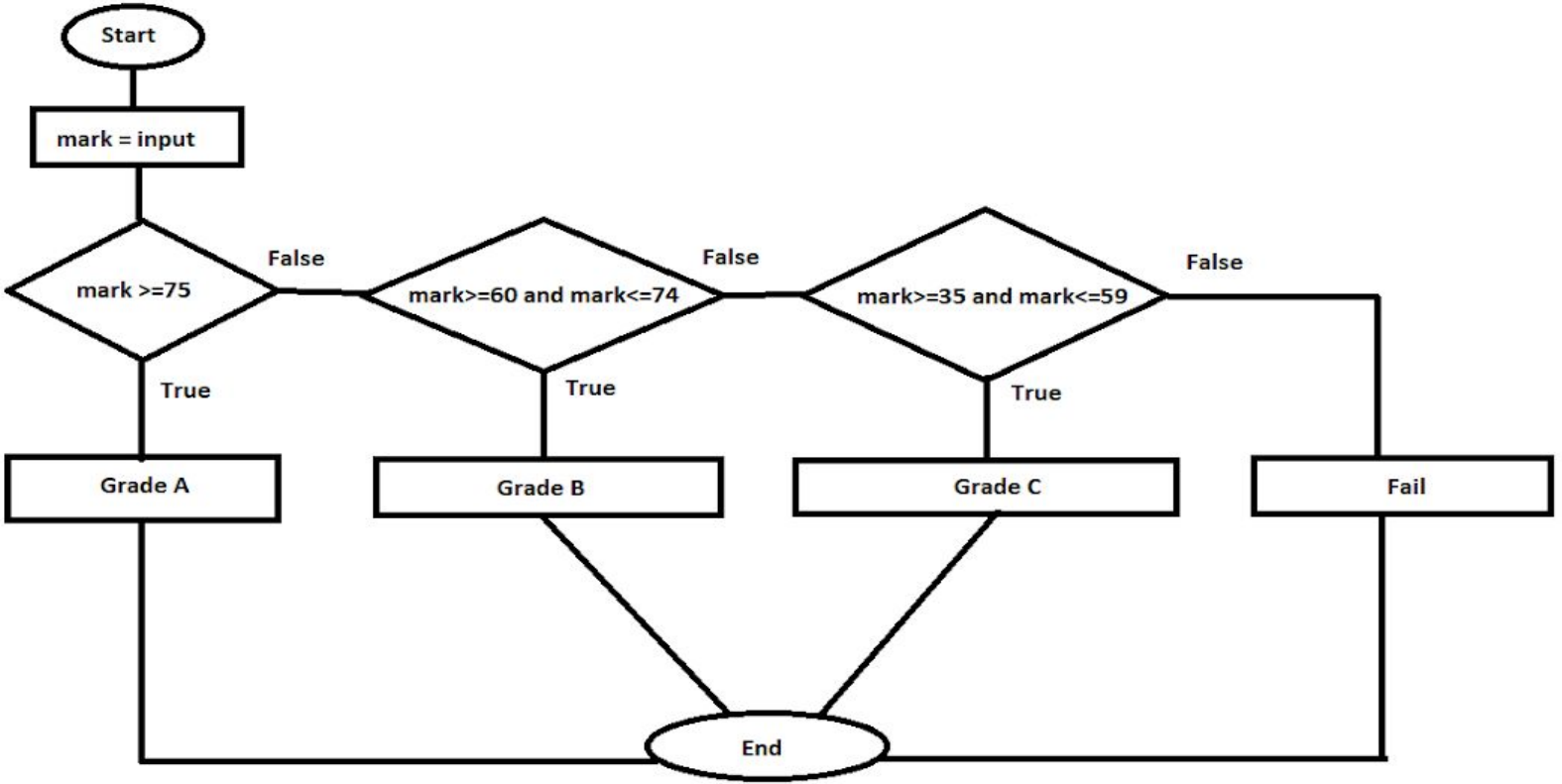
If (mark>=75)
Then print A grade

Elseif (mark>=60 and mark<=74)
Then print B grade

Elseif (mark>=35 and mark<=59)
Then print C grade

Else
Print Fail

Flowchart to given a marks print grade



Divisibility

→ A number is exactly divisible by some other number if it gives 0 as remainder. To check if a number is exactly divisible by some number we need to test if it leaves 0 as remainder or not.

→ In Programming, We a modulo operator %, that evaluates remainder on division of two operands. You can use this to check if a number is exactly divisible by some number or not.

→ For example - if(8 % 2), if the given expression evaluates 0, then 8 is exactly divisible by 2.

Step by step descriptive logic to check whether a number is divisible by 5 and 11 or not.

→ Input a number from user. Store it in some variable say num.
→ To check divisibility with 5,
check if(num % 5 == 0) then num is divisible by 5.

→ To check divisibility with 11,
check if(num % 11 == 0) then num is divisible by 11.

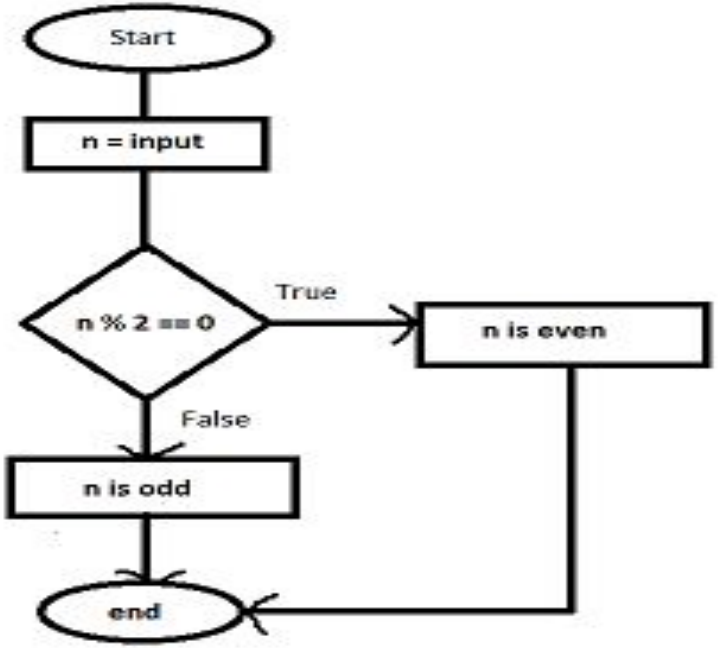
→ Now combine the above two conditions using logical AND operator.
→ To check divisibility with 5 and 11 both,
check if((num % 5 == 0) and (num % 11 == 0)),

then number is divisible by both 5 and 11.

Write an algorithm to check weather the number is even or odd

For a number to an even it should be divisible by 2 and if it is not, it's odd.

Let n ← input
If (n % 2 == 0)
    Then print n is even
Else
    Print n is odd



Leap year through if - else statement

Constraints-:

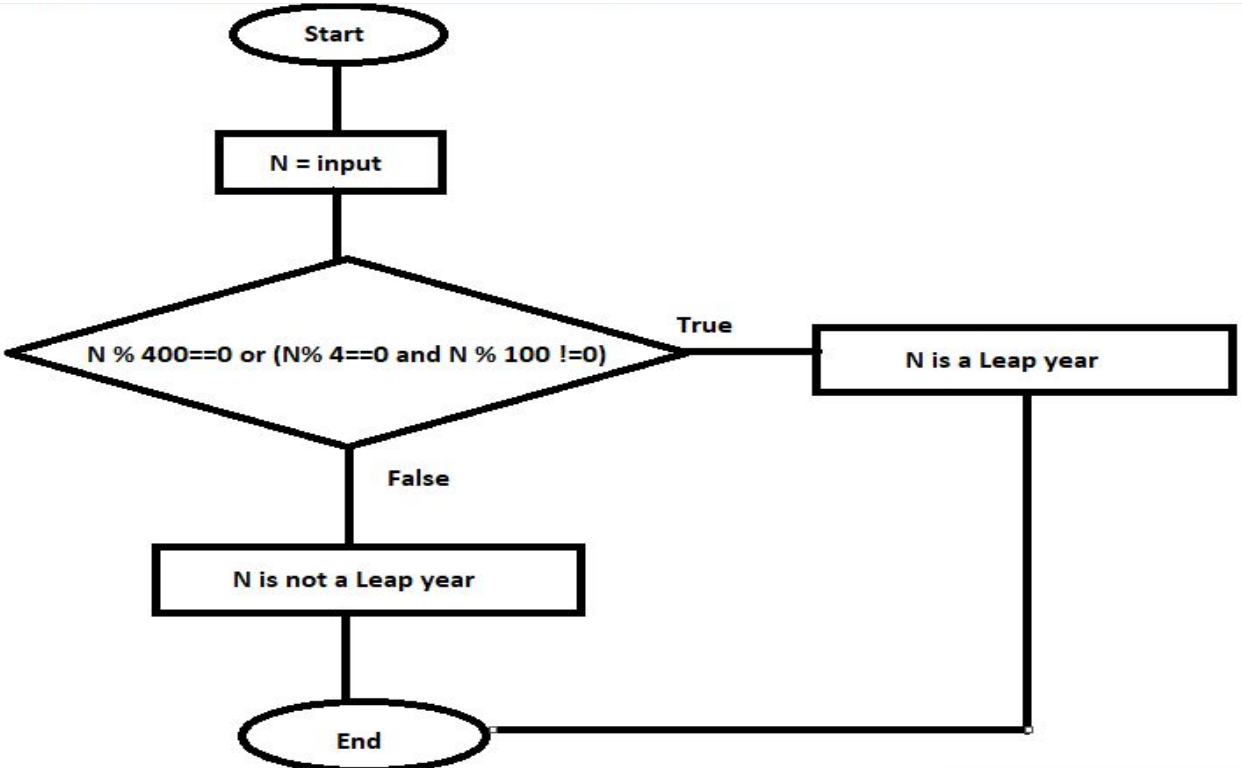
1. N div by 400 → leap year
2. N div by 4 , N not div by 100 → leap

Let N ← input

If (N % 400==0 or (N% 4==0 and N % 100 !=0)
Then print N is a leap year

Else
Print N is not a leap year

FlowChart of Leap year



Leap year through if-elseif-else statement

Constraints-:

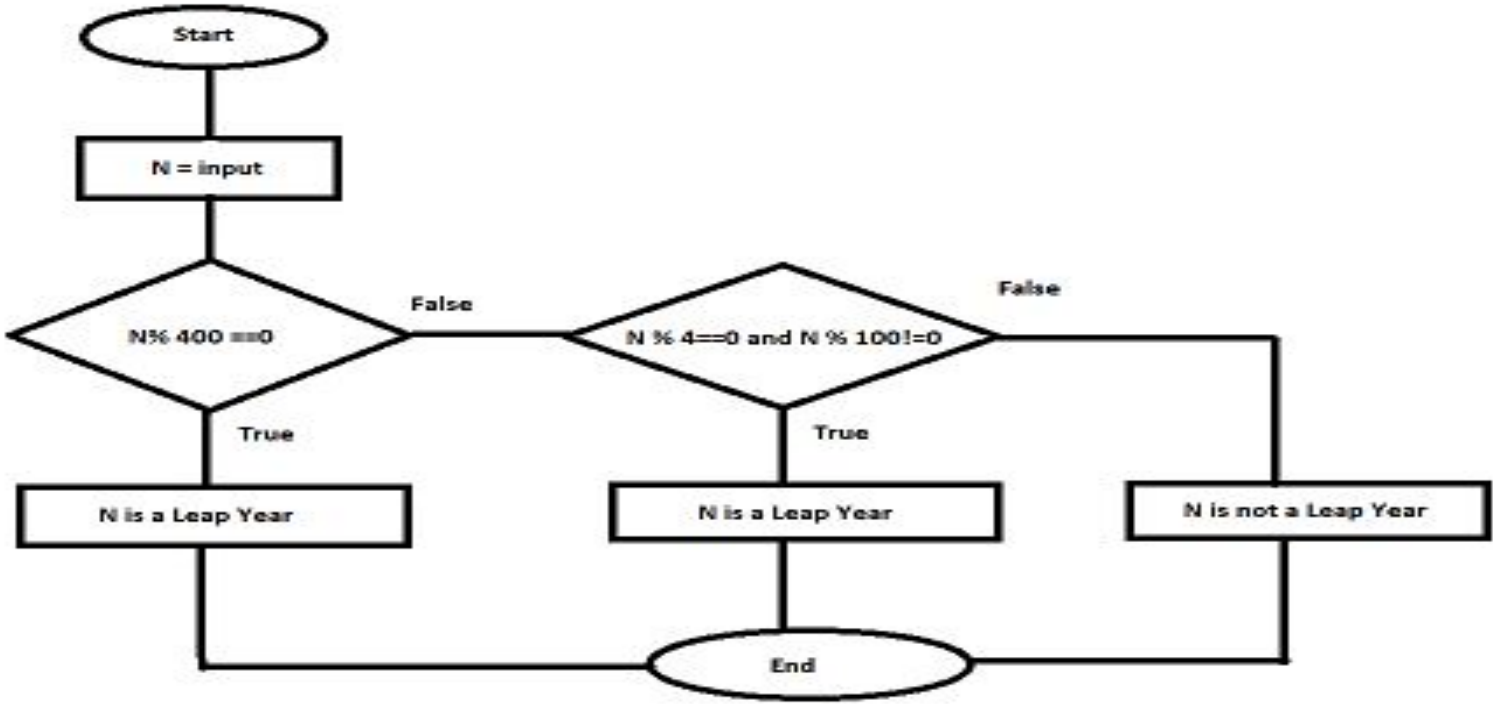
1. N div by 400 → leap year
2. N div by 4 , N not div by 100 → leap

let N ← input

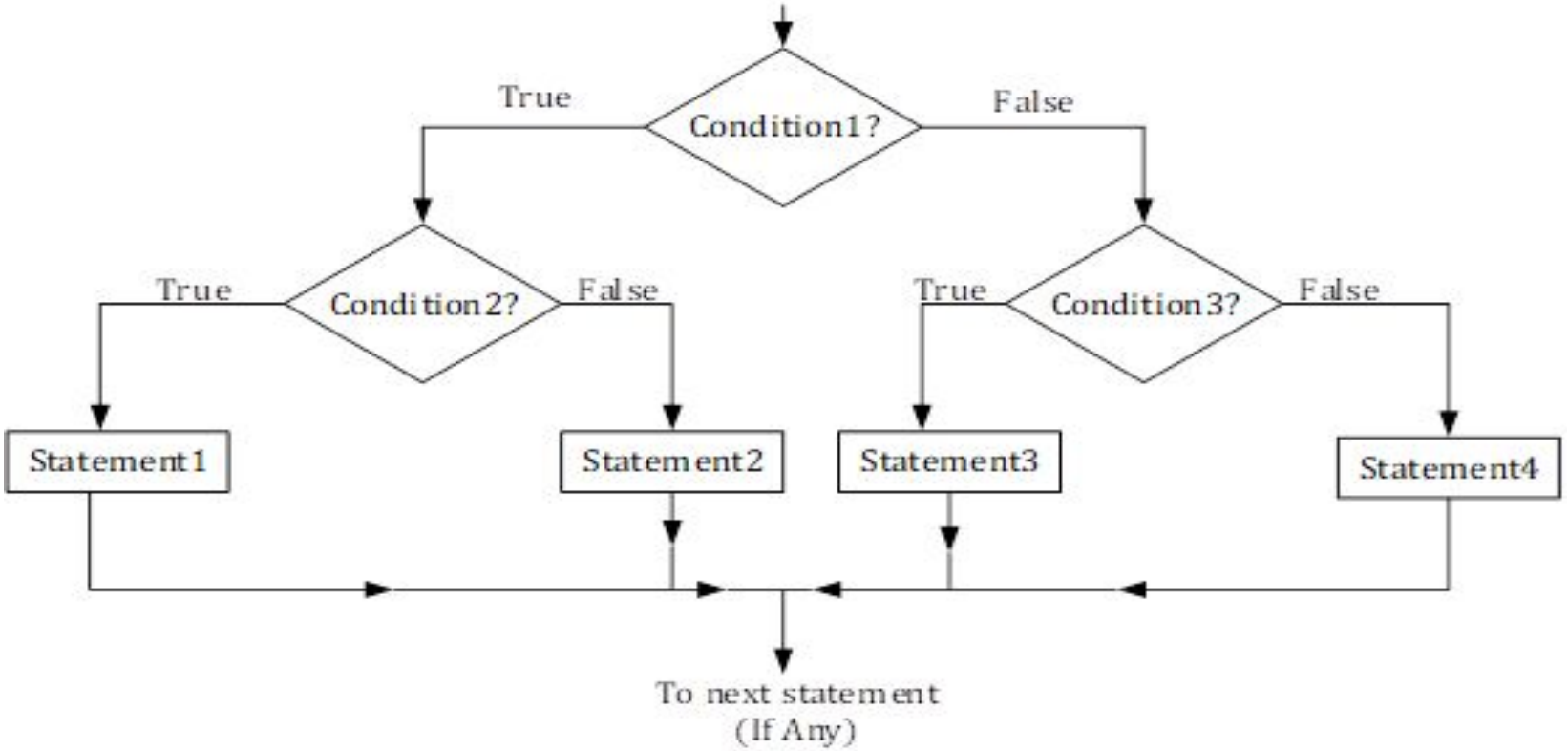
If (N% 400 ==0)
then print N is a Leap year

Elseif (N % 4==0 and N % 100!=0)
then print N is a leap year
Else
print n is not leap year

FlowChart of Leap year with if elseif else



4. Nested IF Condition



Check weather the year is leap or not

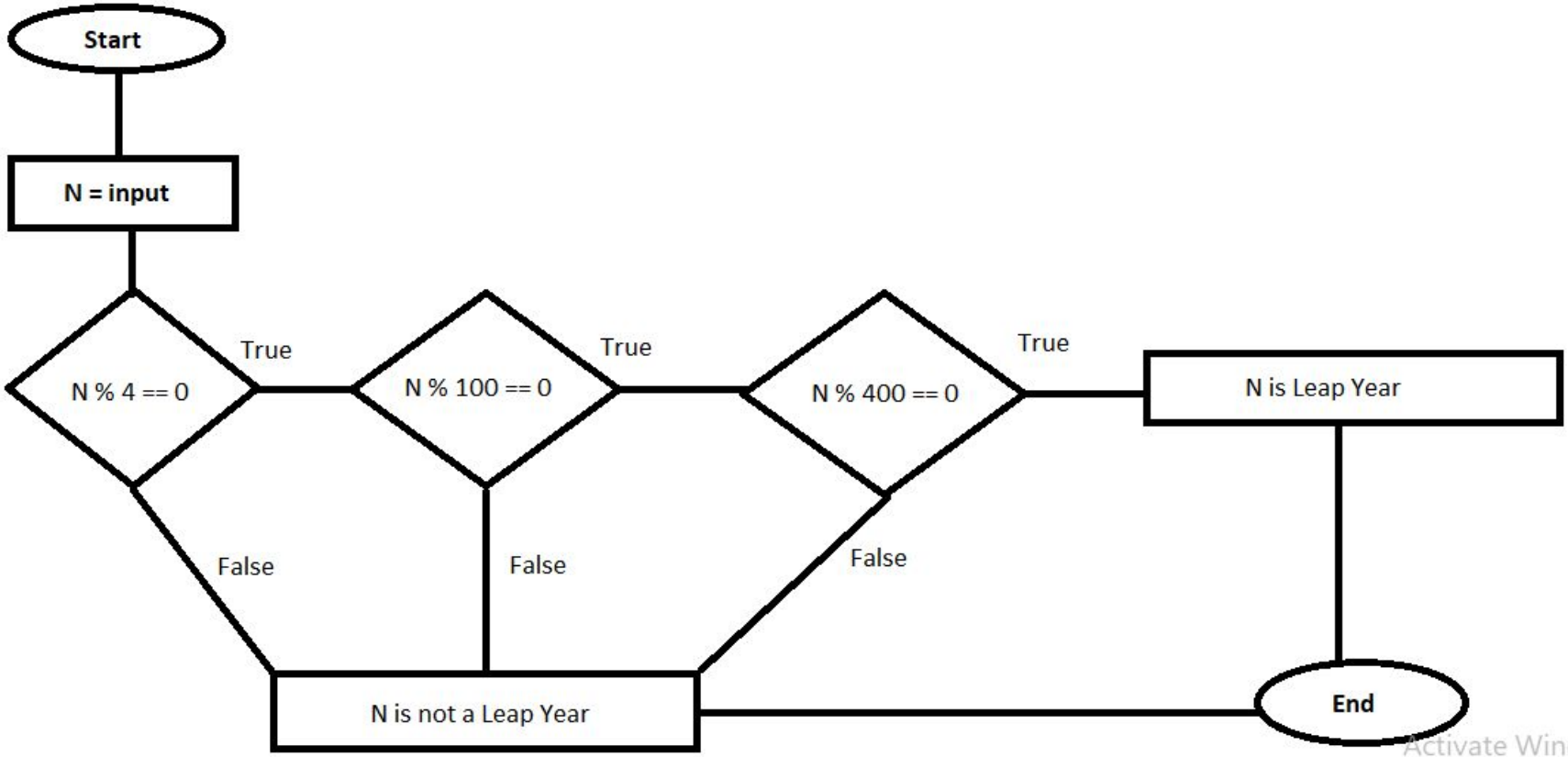
Constraints-:

1. N div by 400 → leap year
2. N div by 4 , N not div by 100 → leap

```
let N ← input

if (n % 4 == 0)
    if (n % 100 == 0)
        if (n % 400 == 0)
            print n is leap year
        else
            print n is not a leap year
    else
        print n i leap year
else
    print n is not leap year
```

FlowChart of Leap year



Exercise (Conditional statements)

1) Write a Flowchart to check user entered number is odd or even
2) Write a Flowchart to check whether the number is leap or not.
3) Write a Flowchart to check user entered gender code is for male or female
4) Write a Flowchart to find maximum out of 3 numbers
5) Write a Flowchart to find minimum out of 3 numbers
6) Write a Flowchart to find middle number out of 3 numbers
7) Write a Flowchart to give Grade on based of percentage
8) Write a Flowchart to check user entered number is negative ,positive or zero
9) Write a Flowchart to check user entered number is divisible by 3 and 7 or not
10) Write a Flowchart to check user entered character is vowel or consonants
11) Write a Flowchart to calculate BMI and validate ideal weight for person if person is overweight or underweight

Day 3-:

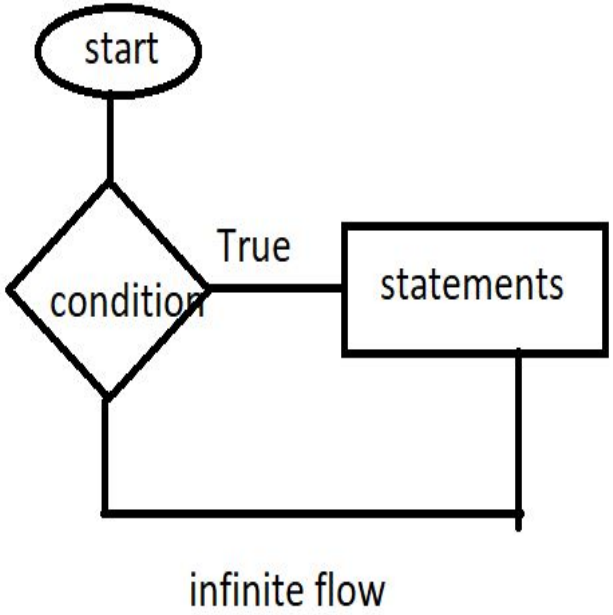
Iterative Statements

Iterative Statements

Iteration statements cause **statements** (or compound **statements**) to be executed zero or more times, subject to some loop-termination criteria. When these **statements** are compound **statements**, they are executed in order, except when either the break **statement** or the continue **statement** is encountered

1) Infinite
2) Finite statements

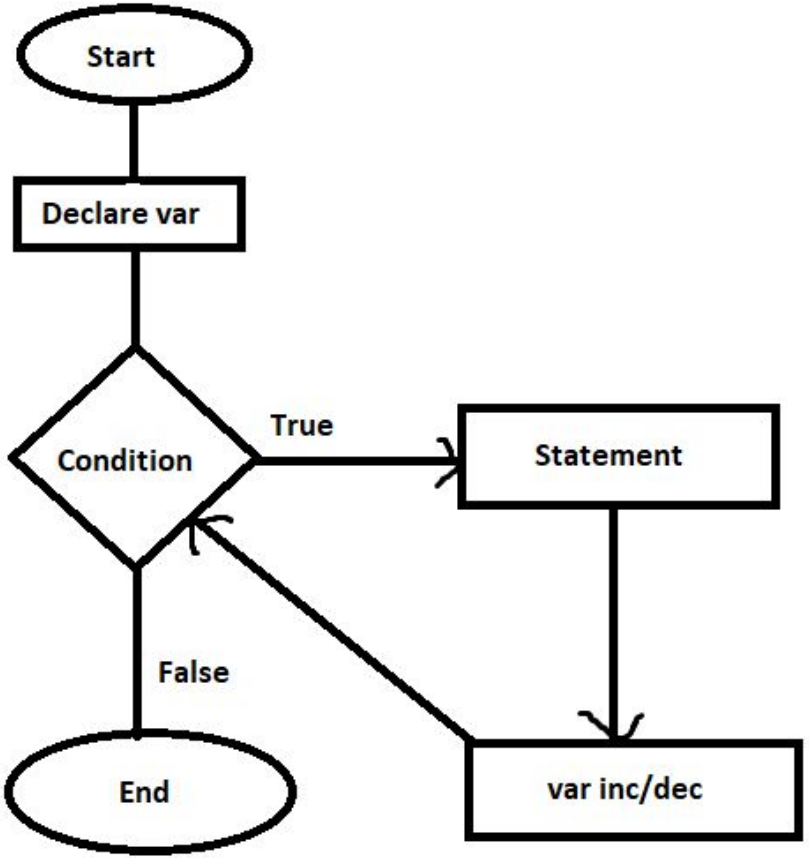
Infinite Statements



infinite flow

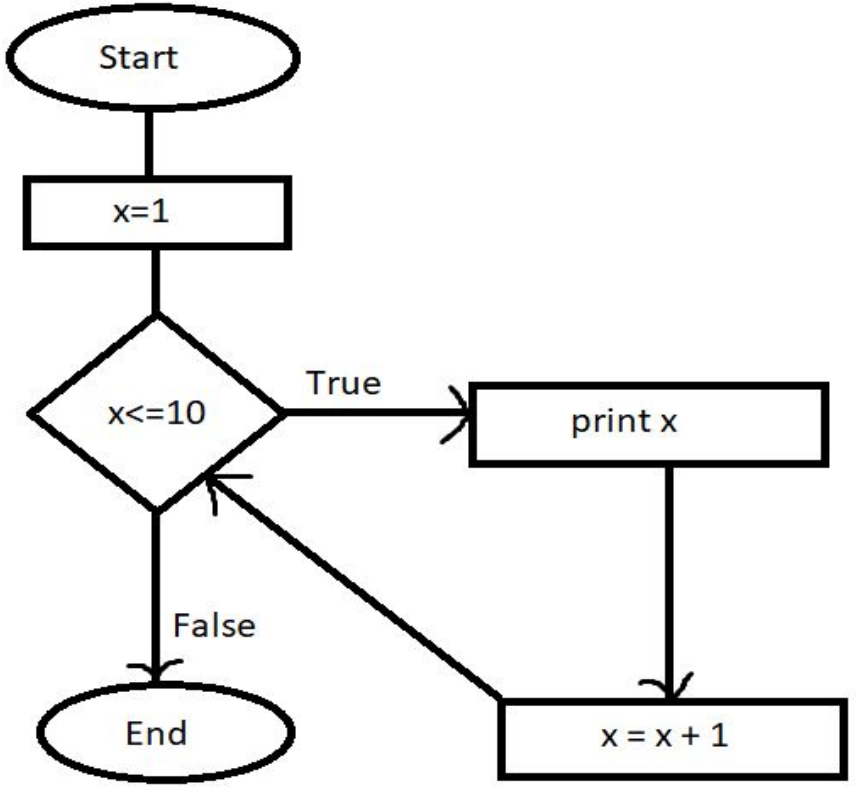
1) Start
2) msg=”Good Morning…!”
3) Loop (True)
   a) Then print msg

Finite Loop

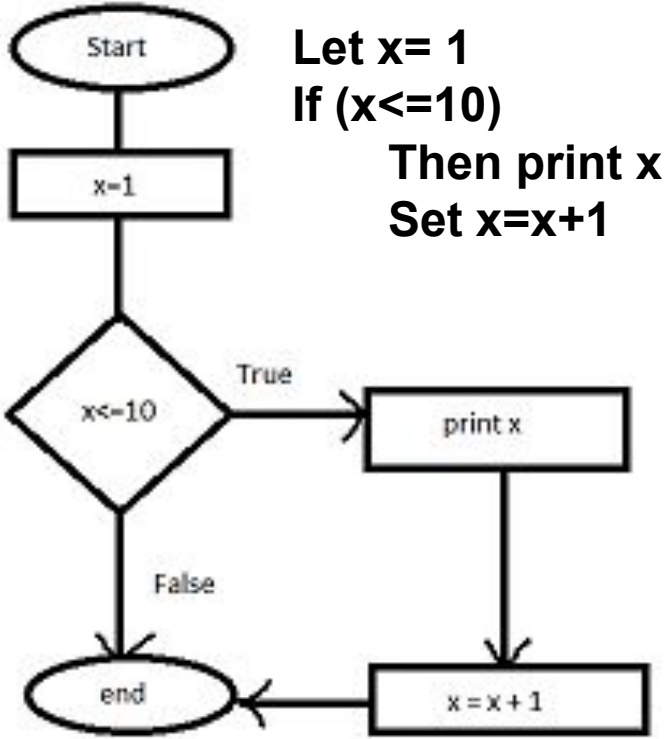

Start
Declare var
Condition
True
Statement
False
End
var inc/dec

Write a flowchart to print 1 to 10 numbers

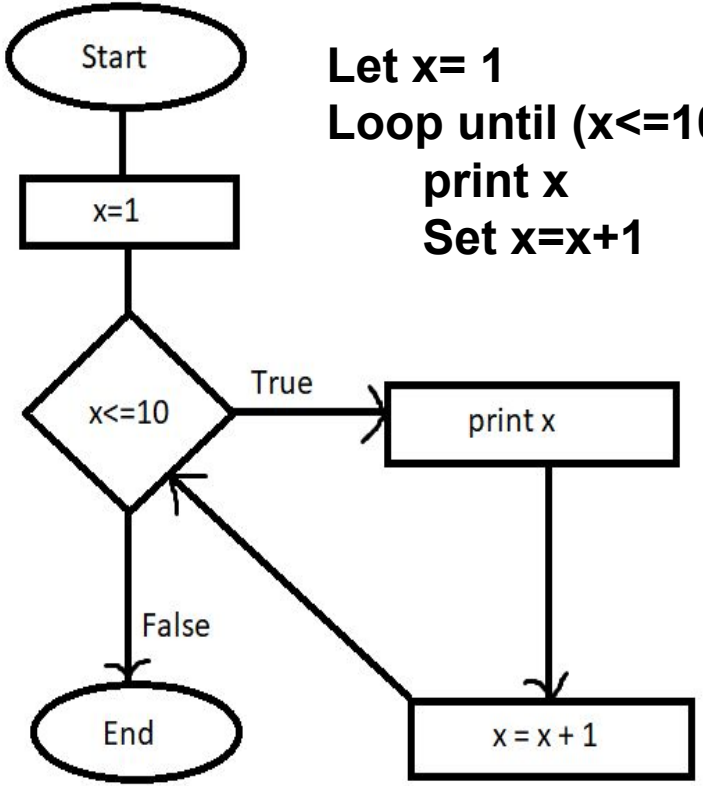
Let x = 1
Loop until x<=10
Print x
Set x = x + 1



If vs loop (to print 1-10)



Let x= 1
If (x<=10)
Then print x
Set x=x+1



Let x= 1
Loop until (x<=10)
print x
Set x=x+1

calculating the factorial of given n

n ← input 5

let f = 1

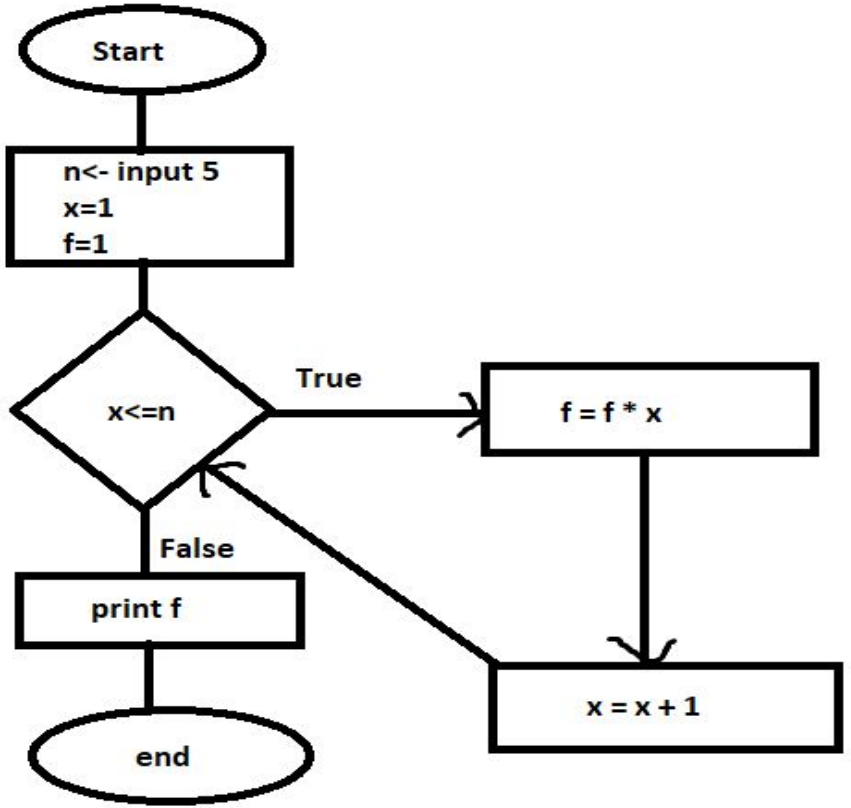
let x = 1

loop until (x <= n)

    f = f * x

    x = x + 1

print f



Working of loop for factorial

Initially f = 1 and x = 1,

loop1: f = 1, x = 1 --> f = f * x = 1 * 1 = 1 , x = x+1=1+1 = 2

Initially f = 1 and x = 1,

loop1: f = 1, x = 1 --> f = f * x = 1 * 1 = 1 , x = x+1=1+1 = 2

loop2: f = 1, x = 2 --> f = f * x = 1 * 2 = 2 , x = x+1=2+1 = 3

Initially f = 1 and x = 1,

loop1: f = 1, x = 1 --> f = f * x = 1 * 1 = 1 , x = x+1=1+1 = 2

loop2: f = 1, x = 2 --> f = f * x = 1 * 2 = 2 , x = x+1=2+1 = 3

loop3: f = 2, x = 3 --> f = f * x = 2 * 3 = 6 , x = x+1=3+1 = 4

Initially f = 1 and x = 1,

loop1: f = 1, x = 1 --> f = f * x = 1 * 1 = 1 , x = x+1=1+1 = 2

loop2: f = 1, x = 2 --> f = f * x = 1 * 2 = 2 , x = x+1=2+1 = 3

loop3: f = 2, x = 3 --> f = f * x = 2 * 3 = 6 , x = x+1=3+1 = 4

loop4: f = 6, x = 4 --> f = f * x = 6 * 4 = 24 , x = x+1=4+1 = 5

Initially f = 1 and x = 1,

loop1: f = 1, x = 1 --> f = f * x = 1 * 1 = 1 , x = x+1=1+1 = 2

loop2: f = 1, x = 2 --> f = f * x = 1 * 2 = 2 , x = x+1=2+1 = 3

loop3: f = 2, x = 3 --> f = f * x = 2 * 3 = 6 , x = x+1=3+1 = 4

loop4: f = 6, x = 4 --> f = f * x = 6 * 4 = 24 , x = x+1=4+1 = 5

loop5: f = 24, x = 5 --> f= f * x = 24 * 5 = 120 , x = x+1=5+1 = 6

Initially f = 1 and x = 1,

loop1: f = 1, x = 1 --> f = f * x = 1 * 1 = 1 , x = x+1=1+1 = 2

loop2: f = 1, x = 2 --> f = f * x = 1 * 2 = 2 , x = x+1=2+1 = 3

loop3: f = 2, x = 3 --> f = f * x = 2 * 3 = 6 , x = x+1=3+1 = 4

loop4: f = 6, x = 4 --> f = f * x = 6 * 4 = 24 , x = x+1=4+1 = 5

loop5: f = 24, x = 5 --> f= f * x = 24 * 5 = 120 , x = x+1=5+1 = 6

X becomes 6, that make loop condition i.e x<=5 is False, that breaks the loop. Coming out of the loop, It print the value of F i.e 120

Iterative Exercise (Iterative Statements)

1) Write a flowchart to print 1 to 10 number's sum
2) Write a flowchart to find sum of user entered 10 numbers
3) Write a flowchart to print multiplication table of user entered number
4) Write a flowchart to print odd numbers
5) Write a program to print series of cube between 1 to 10 numbers
6) Write a program to find factorial of given numbers
7) Write a program to find reverse of user entered number
8) Write a program to check whether number is palindrome or not
9) Write a program to check whether number is armstrong number or not
10) Write a program to check whether number is prime or not
11) Write a program to check number is perfect number or not
12) Write a program to print prime number between 1 to 100
13) Write a program to print fibonacci series
14) Write a program to print number between 1 to 50 which is divisible by 3 but not by 7