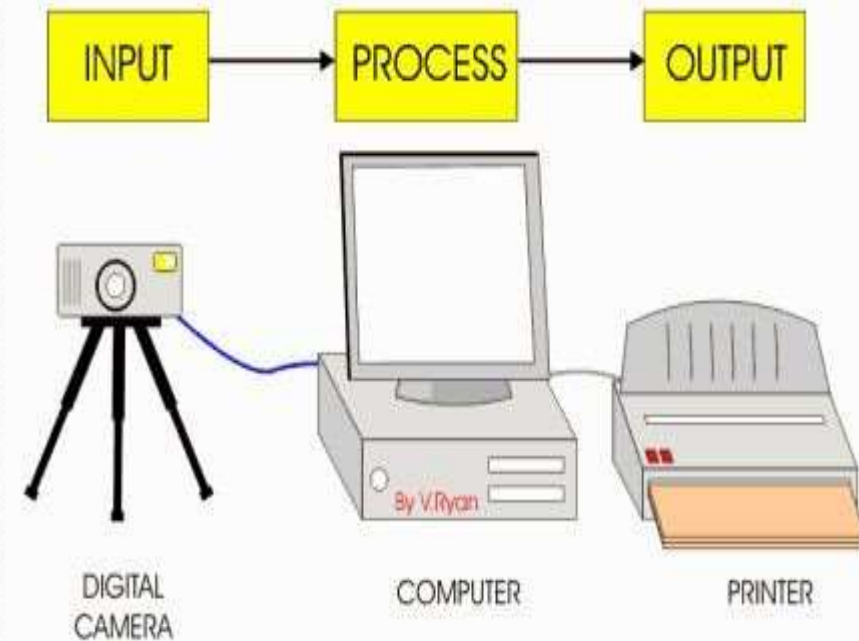


WHAT IS A COMPUTER?

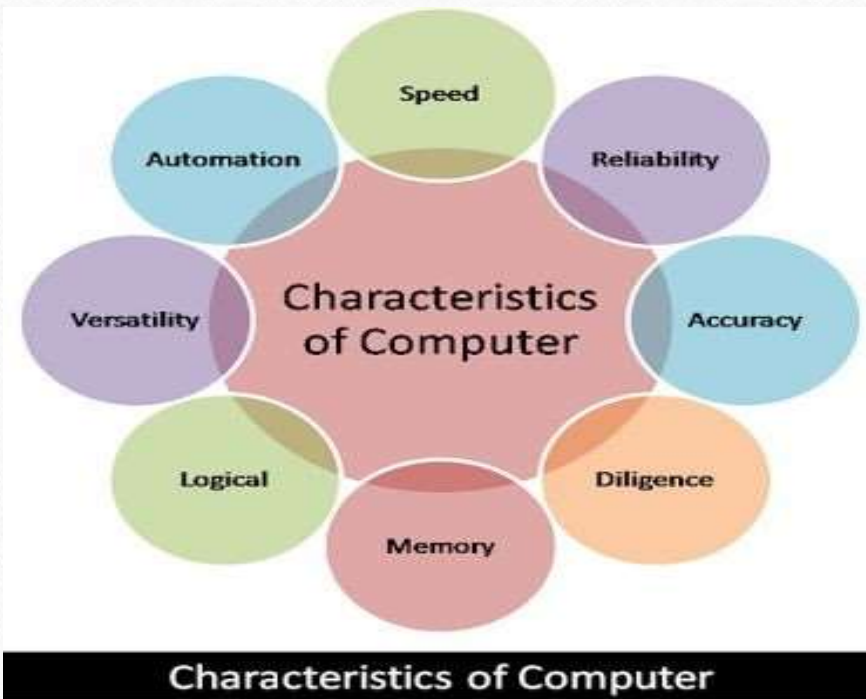
- A **computer** is a machine that accepts data as input, processes that data using programs, and outputs the processed data as information.



COMPUTER

- Computers store information in the form of "1" and "0"s in different types of storages such as memory, hard disk, and usb drives etc.
- The smallest unit of data in a computer is called Bit.
- Data storage units are: bit, byte, kilobyte (kb), megabyte (mb), gigabyte (gb), terabyte (tb), petabyte and exabyte, Zettabyte, Yottabyte

CHARACTERISTICS OF A COMPUTER

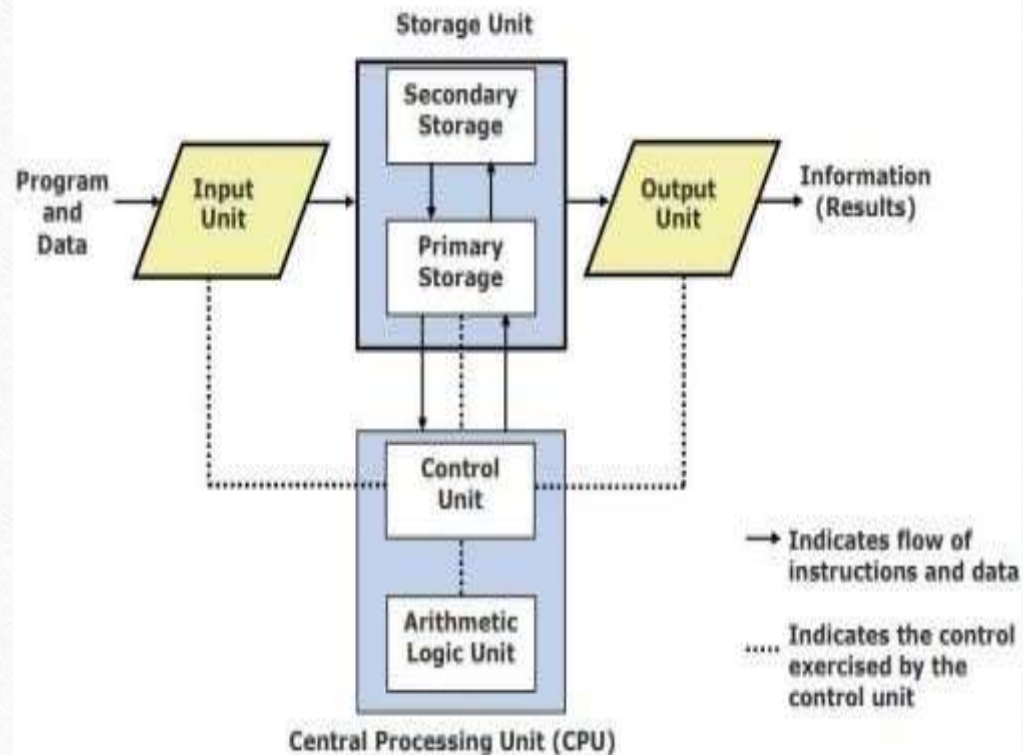


- **Speed:**
 - Computers are much faster to perform **mathematical calculations** than human. The computer is capable of performing millions of tasks per second.
- **Reliability:**
 - Today every major industry and companies in the world have full confidence in their computers, and their **entire business is running from computers**.

CHARACTERISTICS OF A COMPUTER

- **Accuracy:**
 - A computer is very accurate. It **does not make any kind of mistake in calculating**. Sometimes we get some error but these are because of the mistake performed by us.
- **Memory:**
 - Computer systems have a **very large capacity to store any type of data**. A computer can store and resell any information due to its storage capacity.
- **Versatility:**
 - Its means is that the computer is **capable of working in almost every field**. A computer system is **multitasking** so that you can do two tasks very easily at the same time.
- **Automatic:**
 - A computer is an automatic machine because **once started on a job they carry on until the job is finished without any human assistance**

BASIC ORGANISATION OF COMPUTER



- **Input Unit:**
- An input device is a **hardware or peripheral device** used to send data to a computer. An input device allows users to communicate and feed instructions and data to computers for processing, display, storage and/or transmission.
- Some of the most popularly used input devices are:
- Mouse
- Keyboard
- Scanner
- Bar Code Reader
- Joy Stick etc.

BASIC ORGANISATION OF COMPUTER

- **Output Unit:**

- The processed data is **displayed in the form of result** through the output device. Some of the most popularly used Output devices are:
 - Visual Display Unit (Monitor)
 - Printer: Dot Matrix, Line Printers, Ink-jet, Laser Printer.
-

- **Central Processing Unit:**

- The Central Processing Unit (CPU) is known as the **heart of the computer** which takes **control of the entire processing system** of a computer.
- It performs the basic **arithmetical, logical, and input/output operations** of a computer system.
- The part of a computer that interprets and carries out instructions.

BASIC ORGANISATION OF COMPUTER

- **Arithmetic-Logic Unit (ALU):**
 -
 - The ALU is an electronic circuit used to carry out the arithmetic operations like **addition, subtraction, multiplication and division**.
 - It performs the operation on the data provided by the input devices.
 - Logical operations can be used to determine whether particular statement is TRUE or FALSE.
 - The ALU operates on the data available in the main memory and sends them back after processing again to main memory.

BASIC ORGANISATION OF COMPUTER

- **Control Unit:**

- The control unit **coordinates the activities of all the other units** and in the system.
- Its main functions are to **control the transfer of data and information between various units and to initiate appropriate actions** by the arithmetic-logic unit.
- The control unit **fetches instructions from the memory**, decodes them, and directs them to various units to perform the on specified tasks.

BASIC ORGANISATION OF COMPUTER

- **Memory Unit:**
- Computer memory is divided into two types: **Primary Memory & Secondary Memory**
- **Primary Memory**
- The Primary memory is also called Main memory, is used to store data during processing. Once the CPU has carried out an instruction, it needs the result to be stored. This storage space is provided by the computer's memory.
- Primary memory: **Random Access Memory (RAM) and Read Only Memory (ROM).**
- **RAM :** RAM is a volatile memory, which means that the stored information is lost when the power is switched off used to read and write data in RAM
- **ROM:** We can only read the data from ROM and you cannot write anything into it and the data is permanent. ROM is a non - volatile memory

BASIC ORGANISATION OF COMPUTER

- **Secondary Memory**
- The data stored in it is permanent.
- Data can be deleted if necessary.
- It is cheaper than primary memory.
- It has high storage capacity.
- There are different kinds of secondary storage devices available. Few of them are :
- Floppy Disk
- Fixed or Hard Disk
- Optical Disk like: CD (Compact Disk) DVD (Digital Versatile Disk)
- Magnetic Tape Drive

BASIC ORGANISATION OF COMPUTER

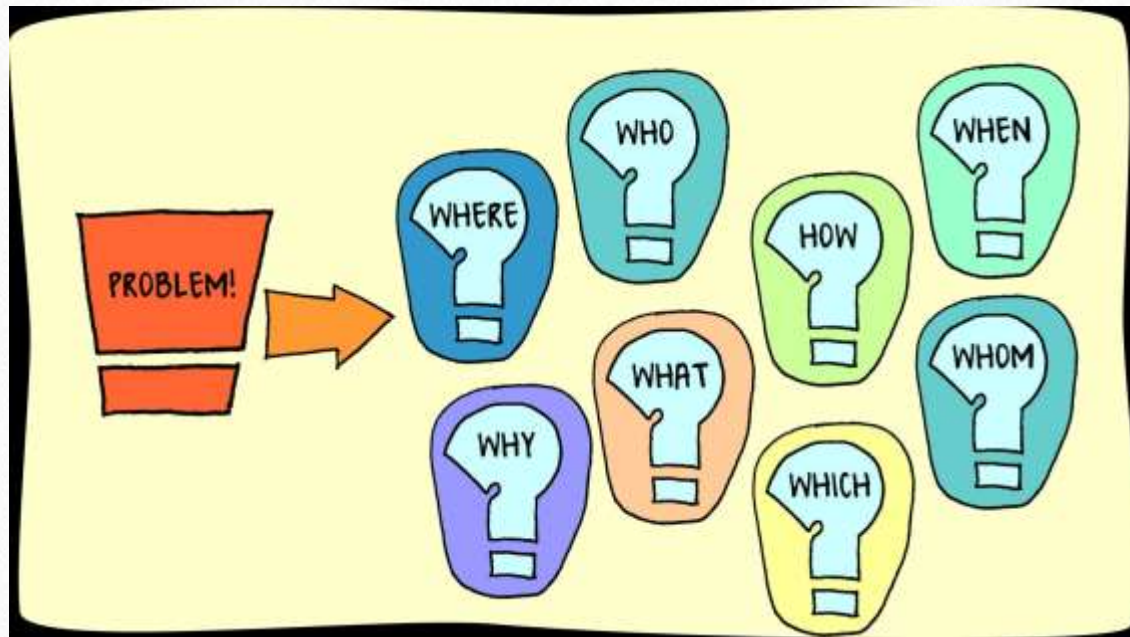
ROM (Read Only Memory)	RAM (Random Access Memory)
ROM is non-volatile	RAM is volatile
ROM is cheaper than RAM	RAM is very expensive
ROM cannot be updated or corrected	RAM can be updated and corrected
ROM serves as permanent data storage	RAM can serve as temporary data storage

APPLICATION OF COMPUTERS IN VARIOUS FIELDS OF ENGINEERING

- ✓ 3D Modelling and Animation
- ✓ Robotics
- ✓ Automated Car Modelling
- ✓ Mobile App Development
- ✓ Web Development
- ✓ Gaming
- ✓ User Interface design for Aerospace Applications
- ✓ Ship Building
- ✓ Architecture

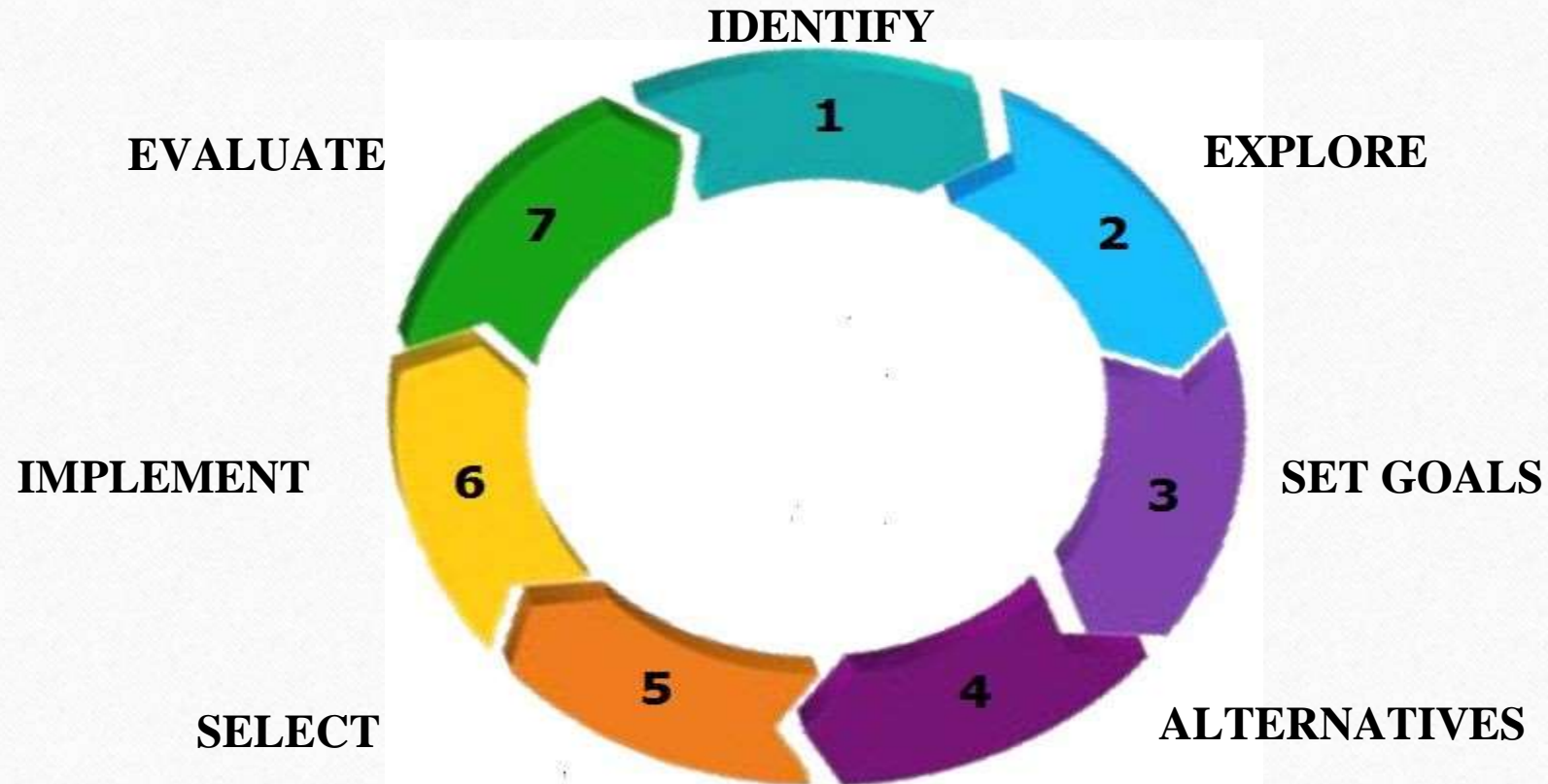


PROBLEM SOLVING



Problem solving is the process of identifying a problem, developing possible solution paths, and taking the appropriate course of action.

THE PROBLEM SOLVING CYCLE





Logic **Building**



IMPORTANT

LOGIC BUILDING

- **Logic**- “a particular way of thinking, especially one that is reasonable and based on good judgment”.
- Logic is a collection of **well-defined activities** to be performed in order to solve the problem.

HOW CAN LOGIC BE REPRESENTED?

- Logic must be represented in a form which can be **understood by humans**.
- In the programming world, the most basic and popular ways of representing logic are
 - Algorithms
 - Flow charts
 - Pseudocode

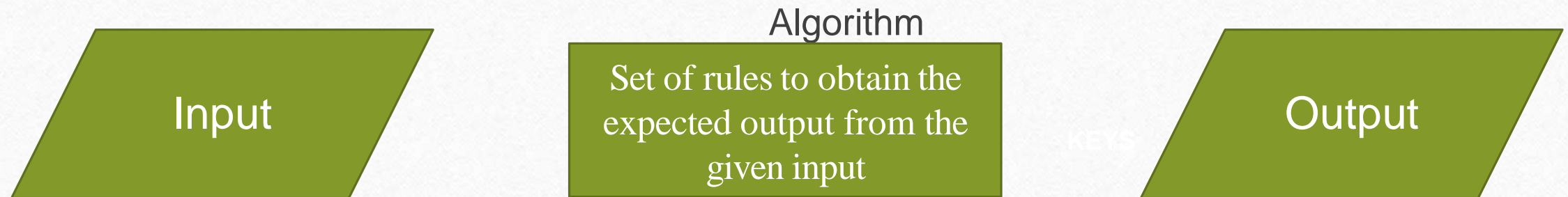
ALGORITHMS

Representation for Logic Building

Algorithms

set of rules for solving a problem

It is a **step-by-step process** to solve a problem



Criteria for Deciding Successful Algorithms



Accuracy

Expected Outcome

Consistency

Must produce the
same result each time
it is run

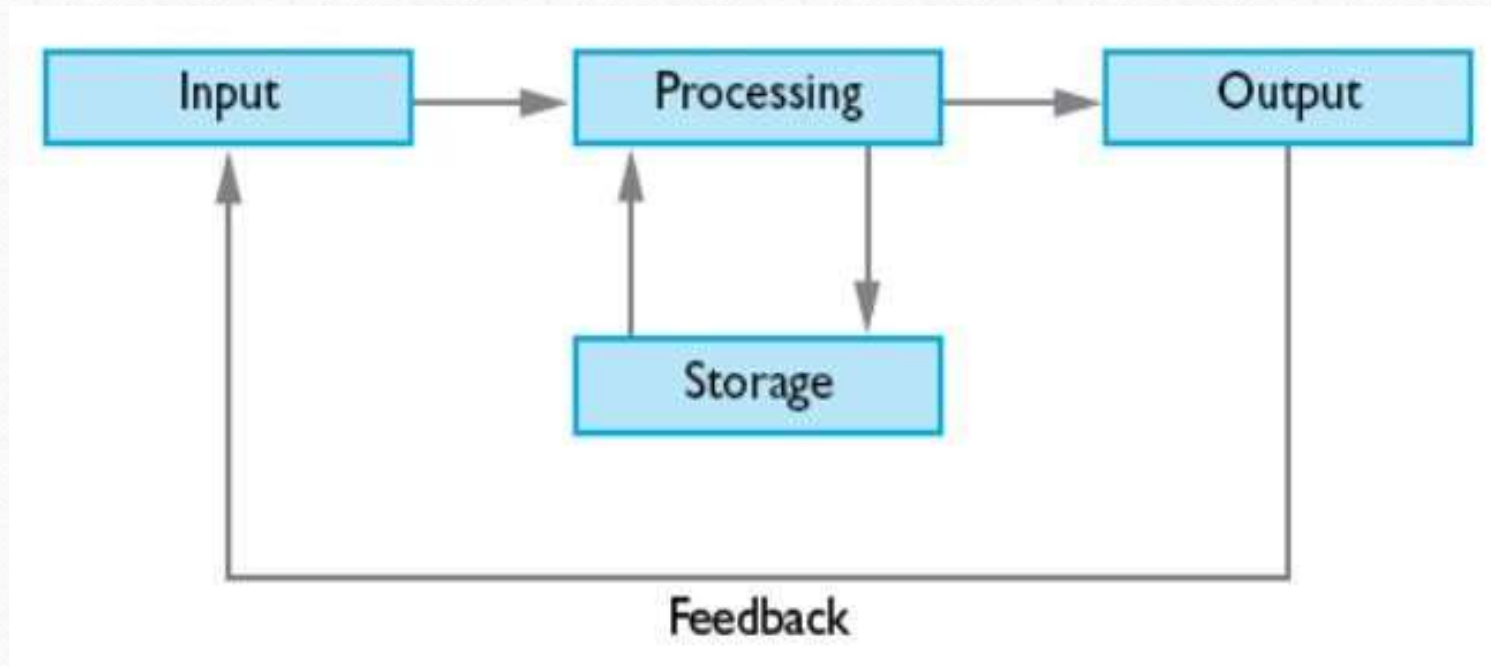
Efficiency

Must solve the problem in
the shortest possible time

Steps involved in algorithm development



Input - Process - Output Model (IPO)



Keywords used in developing algorithms

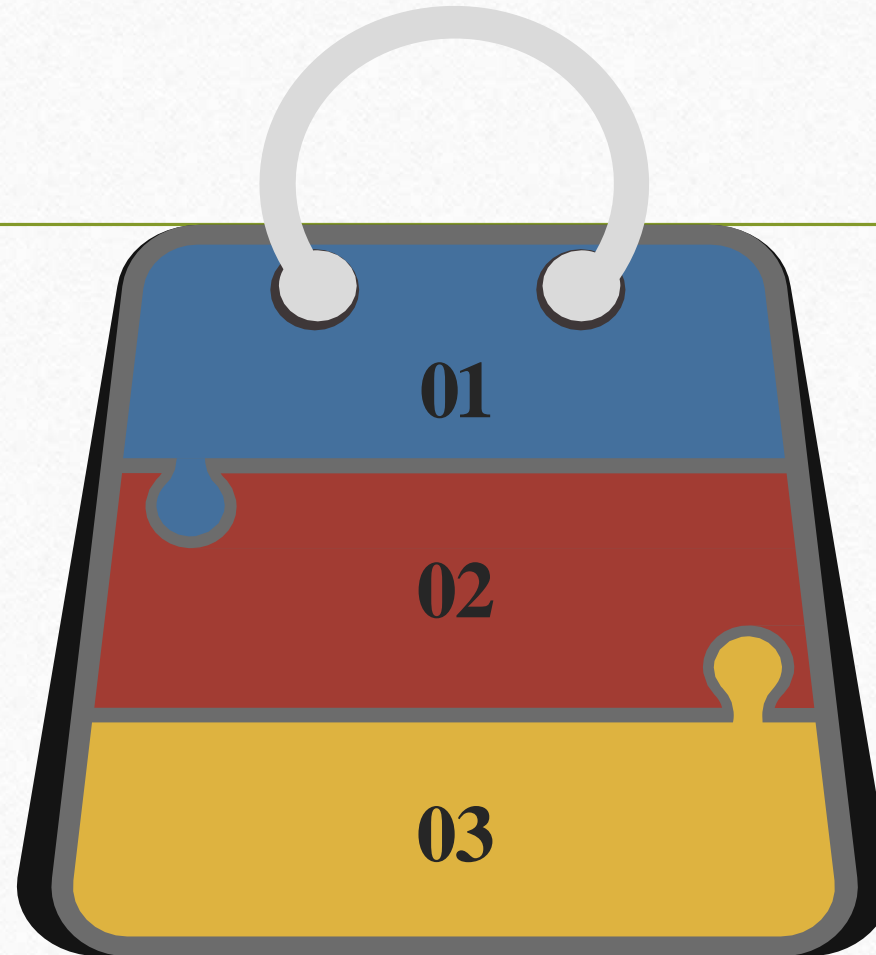
STAGE	KEYWORDS	EXAMPLE
Input	accept, read, input, get	Read name
Output	display, print, output	Display name
Process	calculate, compute	Calculate the area
Decision	check,if,compare	Check whether mark1>mark2
Arithmetic operation	add, subtract, multiply and divide	Add mark1 and mark2
Increment	Add,Increment	Increment counter by 1
Decrement	Subtract,Decrement	Decrement counter by 1
Initialize	Set,Initialize	Initialize counter to 0

Building Blocks of Algorithms

Sequence

Selection

Iteration



Sequence Method

Develop an Algorithm by accepting the name, Internal and External Marks. Compute and display the Total marks.

Step 1: INPUT the name # input stage

Step 2: INPUT the Internal marks # input stage

Step 3: INPUT the External marks # input stage

Step 4: ADD the Internal marks & External marks and assign it to marks # process stage

Step 5: OUTPUT marks # output stage

KEYS

Selection Method

Write an algorithm to accept name and mark. If the mark is greater than or equal to 50 display "PASS" otherwise "FAIL".

Step 1: INPUT the name # input stage

Step 2: INPUT the mark # input stage

Step 3: CHECK IF mark is greater than or equal to 50, if it is TRUE then goto

Step 4 Otherwise goto Step 5 # process stage

Step 4: OUTPUT: Result is PASS goto Step 6 # output stage 1

Step 5: OUTPUT: Result is FAIL # output stage 2

Step 6: End

KEYS

Iteration Method

Write an algorithm to accept name and mark of 5 students. If the mark is greater than or equal to 50 display "PASS" otherwise "FAIL".

-
- Step 1: INITIALIZE the value i equal to 1 # input stage
 - Step 2: IF i is less than or equal to 5, go to Step 3. Otherwise goto Step 9 # process stage 1
 - Step 3: INPUT the name # input stage
 - Step 4: INPUT the mark # input stage
 - Step 5: CHECK IF mark is greater than or equal to 50, if it is TRUE then goto Step 6
Otherwise goto Step 7 # process stage 2
 - Step 6: OUTPUT: Result is PASS goto Step 8 # output stage
 - Step 7: OUTPUT: Result is FAIL # output stage
 - Step 8: ADD 1 with i and store it as i, goto Step 2 # process stage 3
 - Step 9: End

Sequence Method

Write an algorithm to calculate the gross pay by accepting the name, number of days and rate per day for an employee.

Step 1: INPUT the name # input stage

Step 2: INPUT the Days # input stage

Step 3: INPUT the PayRate # input stage

Step 4: Multiply Days by PayRate. Store result in GrossPay. # process stage

Step 5: OUTPUT GrossPay # output stage

Selection

Write an algorithm to calculate the Total pay by accepting the name, grade (A,B and C), Basic Pay.

If grade is "A", then $HRA = 30\%$ of Basic Pay

If grade is "B", then $HRA = 20\%$ of Basic Pay

If grade is "C", then $HRA = 10\%$ of Basic Pay

Compute Total Pay = Basic Pay + HRA

Display Name and Total Pay

Selection

Step 1 : INPUT : name # input stage

Step 2 : INPUT : grade # input stage

Step 3 : INPUT : bpay # input stage

Step 4 : Check If grade is "A", If it is true multiply bpay with 30% and store it as HRA then goto Step 7 Otherwise goto Step 5 # Process stage 1

Step 5 : Check If grade is "B", If it is true multiply bpay with 20% and store it as HRA then goto Step 7 Otherwise goto Step 6 # Process stage 2

Step 6 : Check If grade is "C", If it is true multiply bpay with 10% and store it as HRA # Process stage 3

Step 7 : ADD bpay & HRA and store it as totalpay # Process stage 4

Step 8:OUTPUT1: name# output stage 1

Step 9:OUTPUT2: totalpay# output stage 2

Step 10 : End

Iteration

Write an algorithm to calculate the Total pay for 5 employees by accepting the name, grade (A,B and C), Basic Pay.

If grade is "A", then $HRA = 30\%$ of Basic Pay

If grade is "B", then $HRA = 20\%$ of Basic Pay

If grade is "C", then $HRA = 10\%$ of Basic Pay

Compute Total Pay = Basic Pay + HRA

Display Name and Total Pay

Iteration

Step 1 : INPUT : name # input stage

Step 2 : INPUT : grade # input stage

Step 3 : INPUT : bpay # input stage

Step 4: INITIALIZE the value i equal to 1 # input stage

Step 5: IF i is less than or equal to 5, goto Step 6 Otherwise goto Step 13. # Process stage 1

Step 6 : Check If grade is "A", If it is true multiply bpay with 30% and store it as HRA then goto Step 9 Otherwise goto Step 7
Process stage 2

Step 7 : Check If grade is "B", If it is true multiply bpay with 20% and store it as HRA then goto Step 9 Otherwise goto Step 8
Process stage 3

Step 8 : Check If grade is "C", If it is true multiply bpay with 10% and store it as HRA # Process stage 4

Step 9 : ADD bpay & HRA and store it as totalpay # Process stage 5

Step 10:OUTPUT1: name# output stage 1

Step 11:OUTPUT2: totalpay# output stage 2

Step 12: ADD 1 with i and store it as i goto Step 5 # Process stage 6

Step 13 : End

Sequence – Fill in the blanks

Write an algorithm to interchange (Swap) 2 numbers

Step 1: _____number1 and number2 # input stage

Step 2 : ADD number1 with number2 and store it as number1 # process stage 1

Step 3 : _____number2 from number1 and store it as number2,

process stage 2

Step 4: _____ number 2 from number1 and store it as number1 # process stage 3

Step 5: OUTPUT _____ and _____# output stage

KEYS

Sequence

Write an algorithm to interchange (Swap) 2 numbers

Step 1: INPUT number1 and number2 # input stage

Step 2: ADD number1 with number2 and store it as number1 # process stage 1

Step 3: SUBTRACT number2 from number1 and store it as number2,

process stage 2

Step 4: SUBTRACT number2 from number1 and store it as number1 # process stage 3

Step 5: OUTPUT number1 and number2 # output stage

KEYS

Selection – Fill in the blanks

Write an algorithm for eligibility to vote in India

Step 1: _____age # input stage

Step 2 : _____whether given age is greater than or equal to eighteen, if it is

TRUE goto Step 3, Otherwise goto Step 4 # process stage 1

Step 3: _____The person is eligible to vote goto Step 5 # output stage 1

Step 4: _____The person is not eligible to vote # output stage 2

Step 5: End

KEYS

Selection

Write an algorithm for eligibility to vote in India

Step 1: INPUT age # input stage

Step 2 :CHECK whether given age is greater than or equal to eighteen, if it is TRUE goto Step 3, Otherwise goto Step 4 # process stage 1

Step 3: OUTPUT The person is eligible to vote goto Step 5 # output stage 1

Step 4: OUTPUT The person is not eligible to vote # output stage 2

Step 5: End

KEYS

Iteration

Write an algorithm for sum of natural numbers

Step 1: _____number # input stage

Step 2: _____the value i equal to 1 and sum equal to zero # Input stage

Step 3: IF i is less than or equal to number, go to Step 4. Otherwise Step 6

process Stage 1

Step 4: _____sum with i and store it as sum. # process Stage 2

Step 5: _____1 with i and store it as i, Goto Step 3 # process Stage 3

Step 6: _____sum # output stage

Step 7: End

Iteration

Write an algorithm for sum of natural numbers

Step 1: INPUT number # input stage

Step 2: INITIALIZE the value i equal to 1 and sum equal to zero # Input stage

Step 3: IF i is less than or equal to number, go to Step 4. Otherwise Step 6

process Stage 1

Step 4: Add sum with i and store it as sum. # process Stage 2

Step 5: ADD 1 with i and store it as i, Goto Step 3 # process Stage 3

Step 6: DISPLAY sum # output stage

Step 7: End

Algorithm for String Searching

Step 1: INPUT text and pattern # input stage

Step 2: CHECK whether entire pattern is found or end of text is reached. If it is true goto Step 6
Otherwise goto Step 3 # Process stage

Step 3: CHECK whether the text letter is equal to the pattern letter. If it is true goto step 4 Otherwise
Step 5 # Process stage

Step 4: COMPARE next letter of pattern to next letter of text goto Step 2 # Process stage

Step 5: MOVE pattern down text by one letter goto Step 2 # Process stage

Step 6: End

```
tetththeheehthtehtheththehehtht
the
tetththeheehthtehtheththehehtht
the
tetththeheehthtehtheththehehtht
the
tetththeheehthtehtheththehehtht
the
tetttheheehthtehtheththehehtht
the
tetththeheehthtehtheththehehtht
the
```

Assignment Question – Supermarket Billing System I

Customer buying only one item - Sequence

Input

- Customer name
- Customer ID
- Price of an item (price)
- Quantity of the item (qty)
- Tax Percent (tax)
- Discount Percent (dis)

Calculate

- $\text{gross} = \text{price} + (\text{price} * (\text{tax} / 100))$
- $\text{netamt} = \text{qty} * (\text{gross} - (\text{gross} * (\text{dis} / 100)))$

Output

- Net Amount (netamt)

Assignment Question – Supermarket Billing System II

Discount is applied only if the customer buys a minimum of 5 quantities- Selection

Input

- Customer name
- Customer ID
- Price of an item (price)
- Quantity of the item (qty)
- Tax Percent (tax)
- Discount Percent (dis)

Calculate

- $\text{gross} = \text{price} + (\text{price} * (\text{tax} / 100))$
- $\text{netamt} = \text{qty} * (\text{gross} - (\text{gross} * (\text{dis} / 100)))$ ----> Applied when $\text{qty} \geq 5$
- $\text{netamt} = \text{qty} * \text{gross}$ ---> Applied when $\text{qty} < 5$

Output

- Net Amount (netamt)

Assignment Question – Supermarket Billing System III

5 customers do the purchase - Iteration

Input

- Customer name
- Customer ID
- Price of an item (price)
- Quantity of the item (qty)
- Tax Percent (tax)
- Discount Percent (dis)

Calculate

- $\text{gross} = \text{price} + (\text{price} * (\text{tax} / 100))$
- $\text{netamt} = \text{qty} * (\text{gross} - (\text{gross} * (\text{dis} / 100)))$ ----> Applied when $\text{qty} \geq 5$
- $\text{netamt} = \text{qty} * \text{gross}$ ---> Applied when $\text{qty} < 5$

Output

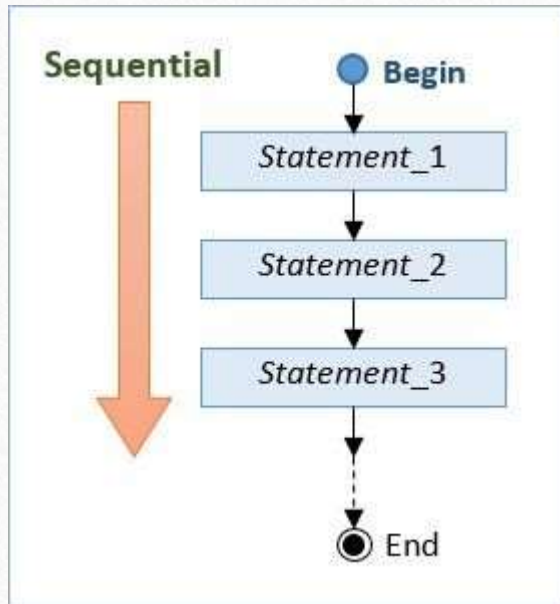
- Net Amount (netamt)

FLOWCHARTS

**DIAGRAMATIC REPRESENTATION FOR
LOGIC BUILDING**

FLOWCHART

IMPORTANCE



A flowchart is a diagram that depicts the “flow” of a program.

Diagrammatic representation of an algorithm.

Step-by-Step approach to solve a task.

FLOWCHART SYMBOLS



Oval



Rectangle



Parallelogram



Diamond



Arrow Lines



Circle

FLOWCHART NOTATIONS

Start / End

The start or end of a workflow.

Decision

Decision point in a process or workflow.

Project / Task

Process or action.

Flow Lines

Input / Output

Data: Inputs to, and outputs from, a process.

Connector

Used to connect one part of a flowchart to another.

Symbols are defined by ANSI Standard

SCENARIO

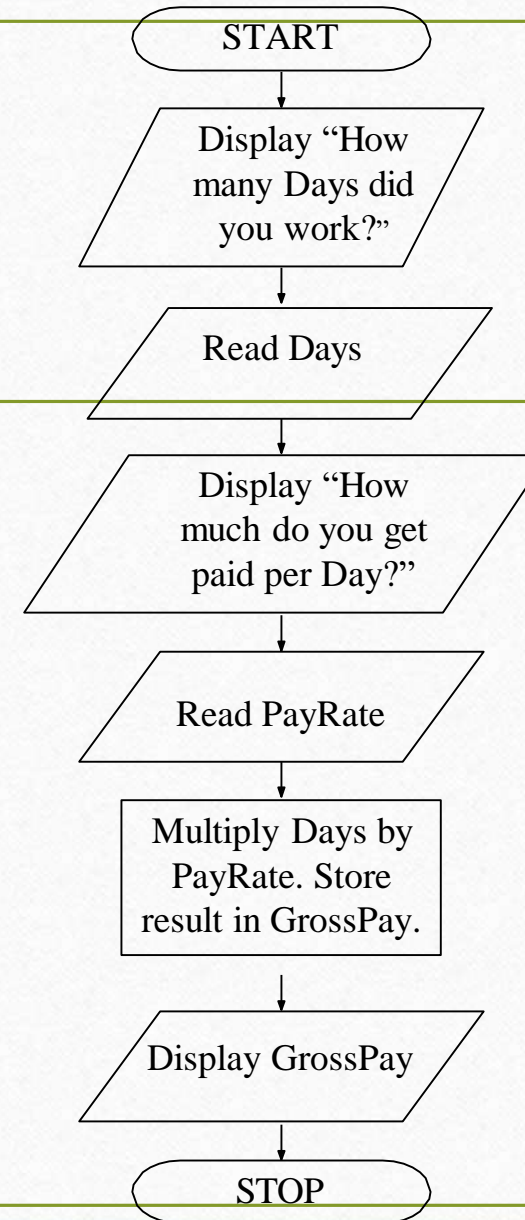
I am Ramu, working in the Company XYZ as a daily wage. I will tell you the number of Days I worked and how much I am paid for each day. I want to Calculate my Gross Pay. Kindly help me in calculation.

Lets see the Flowchart



I am Ramu, working in the Company XYZ as a daily wage. I will tell you the number of Days I worked and how much I am paid for each day. I want to Calculate my Gross Pay. Kindly help me in calculation.

- Start
- How many days did you work?
- 10
- How much do you get paid per days?
- 100
- $10 * 100$
- 1000
- Stop

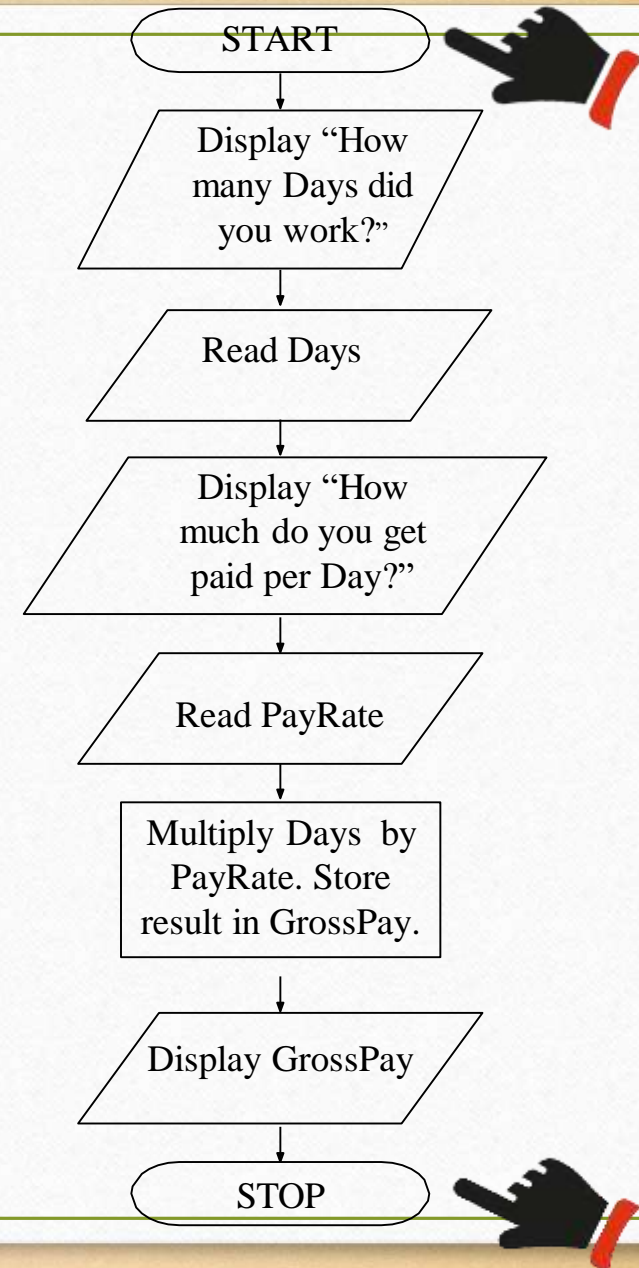


BASIC FLOWCHART SYMBOLS

TERMINALS

Represented by Rounded Rectangles

Indicate a Starting or Ending Point



BASIC FLOWCHART SYMBOLS

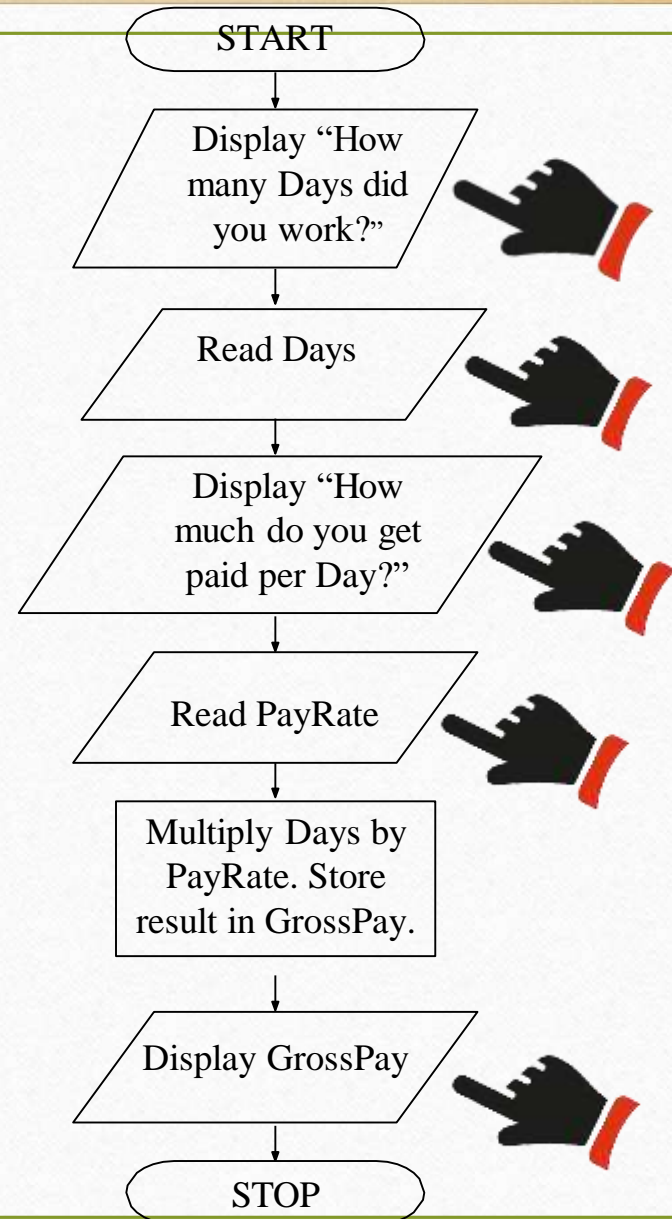
INPUT / OUTPUT OPERATIONS

Represented by Parallelograms

Indicate an Input or Output Operation

Display “How many Days did you work?”

Read Days



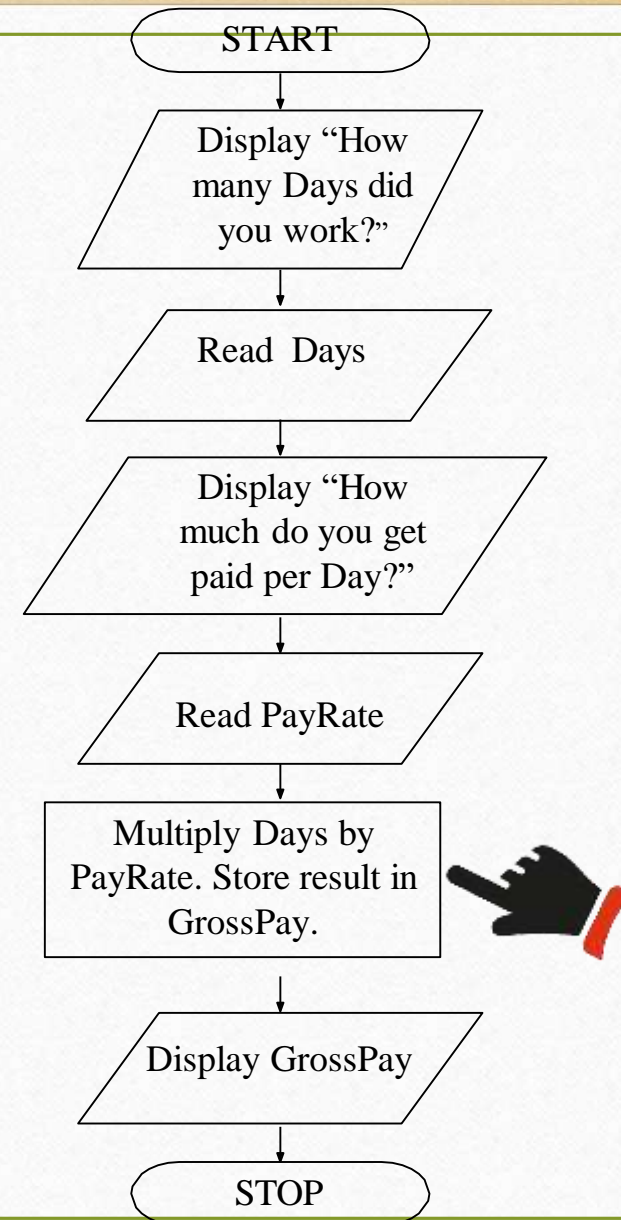
BASIC FLOWCHART SYMBOLS

PROCESSES

Represented by Rectangles

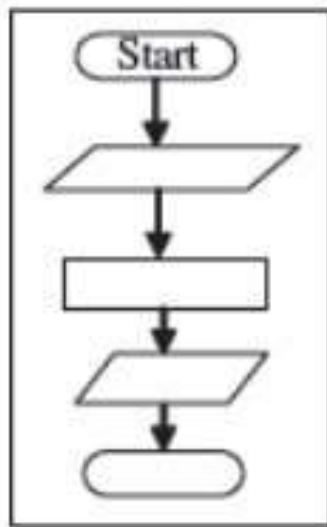
Indicates a process such as Mathematical Computation or Variable Assignment

Multiply Days by PayRate. Store result in GrossPay.

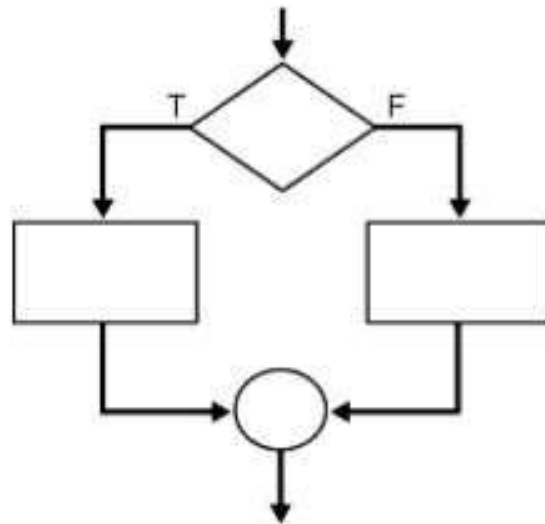


VARIOUS CONTROL STRUCTURES

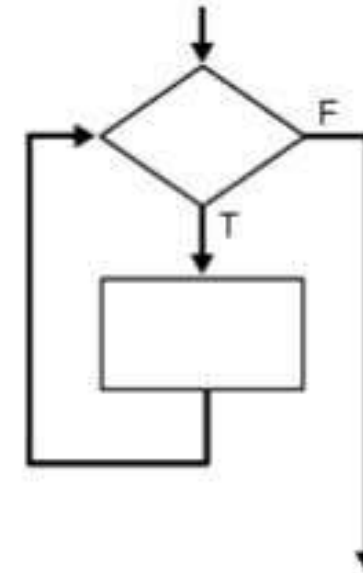
SEQUENCE



SELECTION



ITERATION



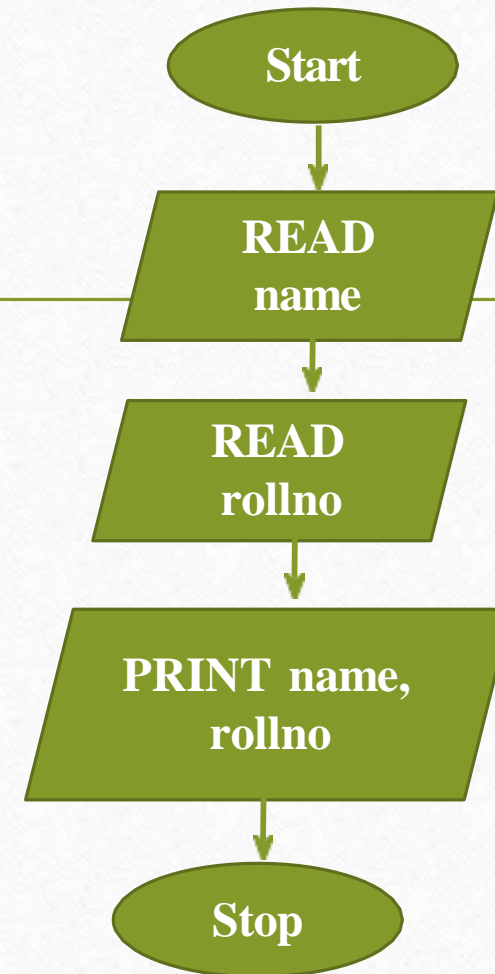
SEQUENCE

Steps are in Sequential Order

There are no Alternate paths

Simple Example

We need to develop a system which reads Name, Roll number of one student and displays it.



SELECTION

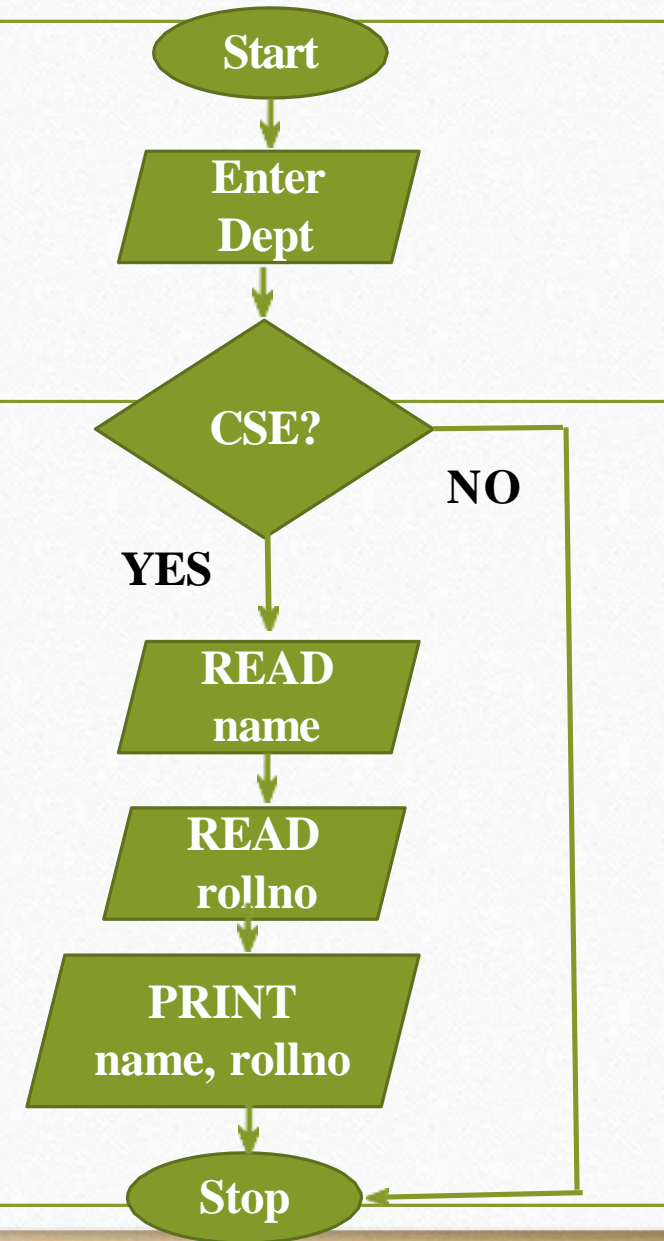
Steps are not executed in Sequence

We have Alternate paths. Path are chosen based on the decision

Decisions are made based on some Conditions

Simple Example

We need to develop a system which reads name, roll number of one student and displays it, if he/she belongs to CSE department.



ITERATION

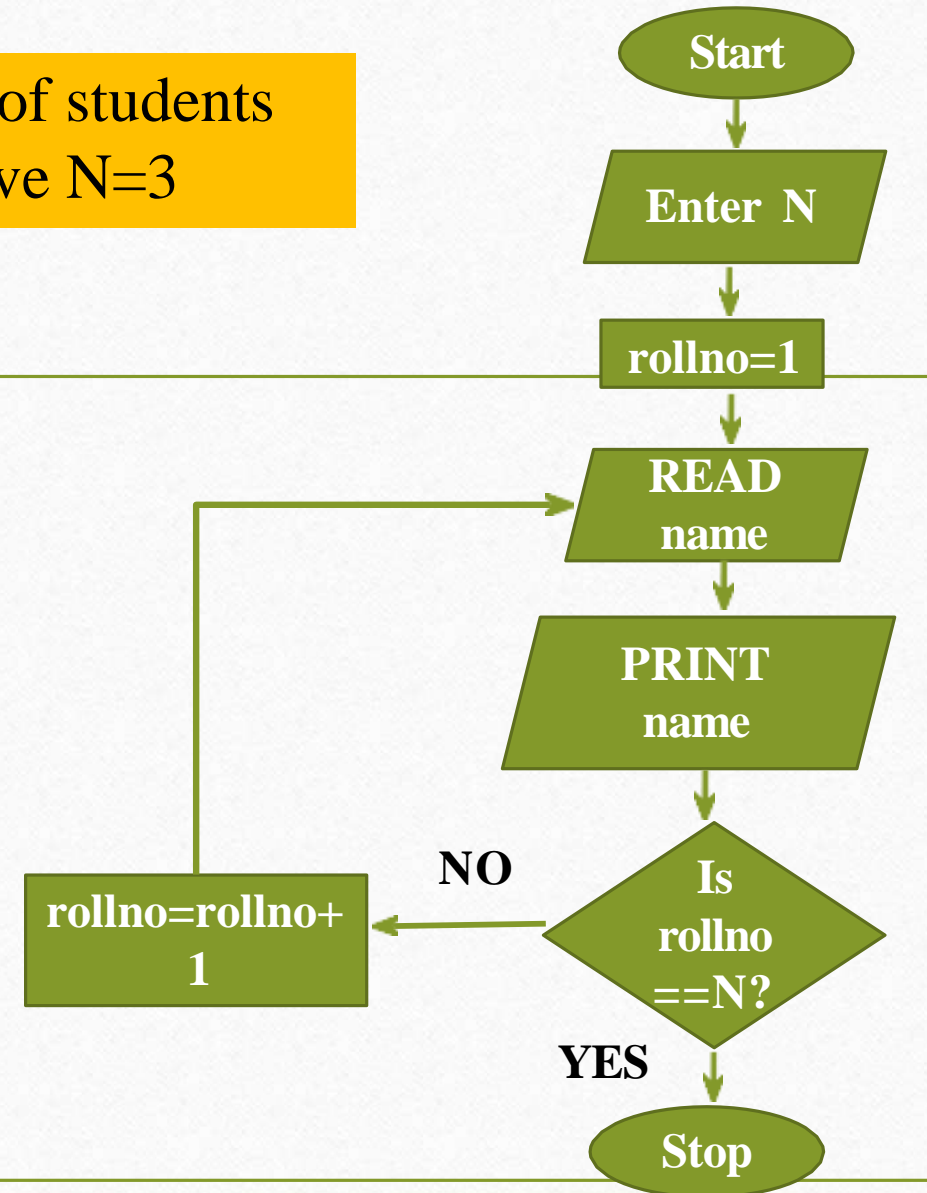
N - No. of students
Let's have $N=3$

Steps are in Sequence but executed many times and repetitively.

Repetitions occurs until the condition fulfills

Simple Example

We need to develop a system which reads name, roll number of 'n' students and displays it.



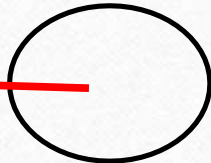
REVIEW

What do each of the following symbols represent?

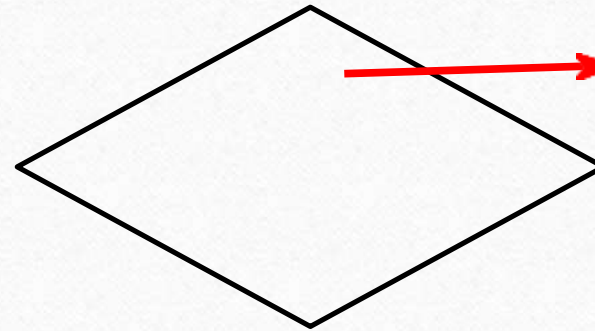
Terminal
(start / stop)



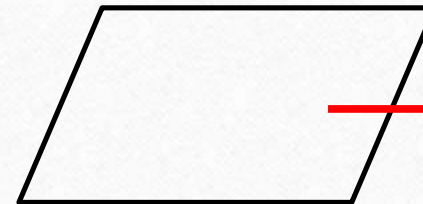
Connector



Process



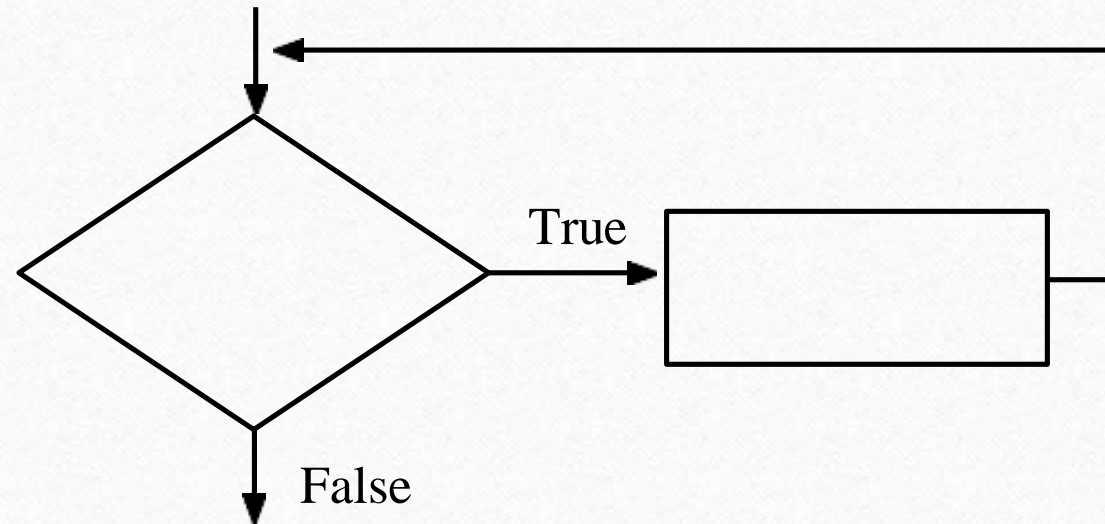
Decision



Input/Output

REVIEW

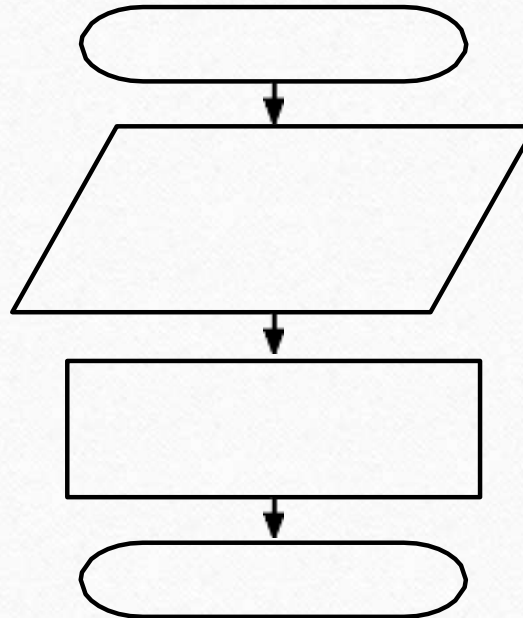
What type of Structure is this?



Iterative

REVIEW

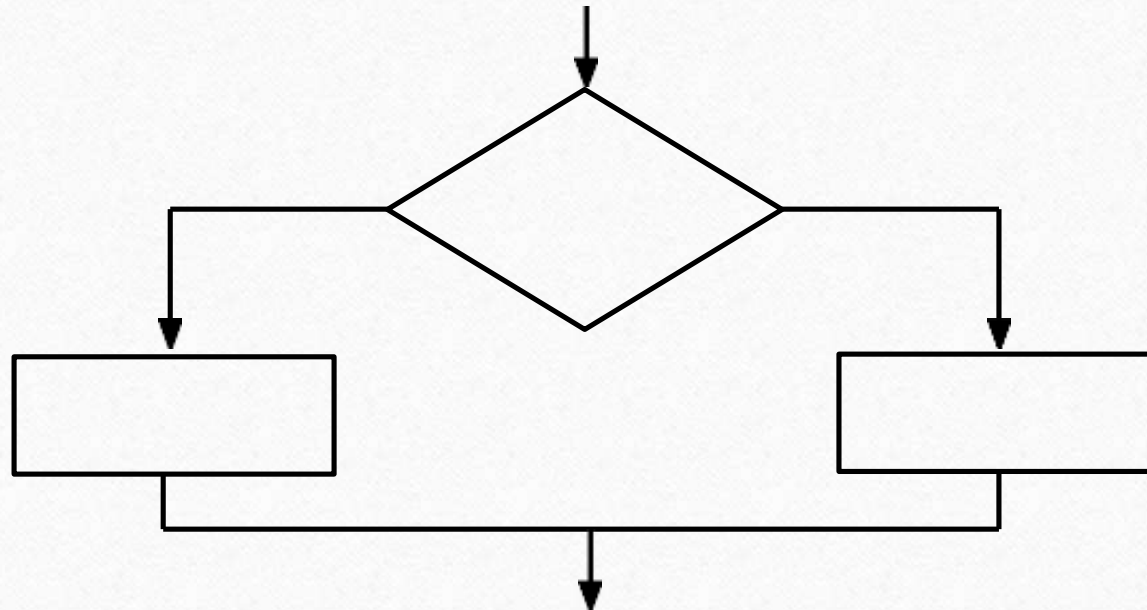
What type of Structure is this?



Sequential

REVIEW

What type of Structure is this?

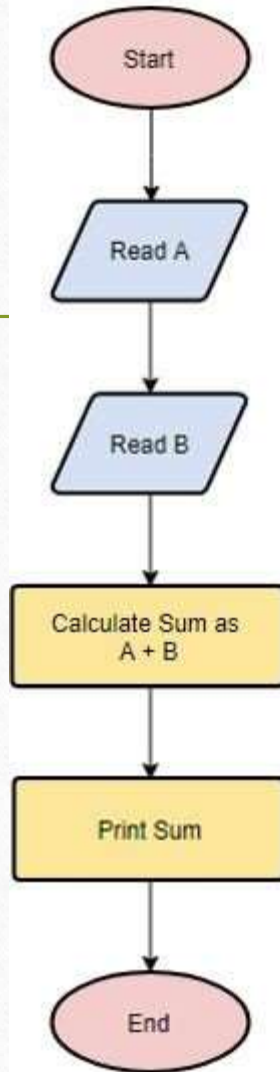


Selection

EXAMPLES

Lets Have a Look

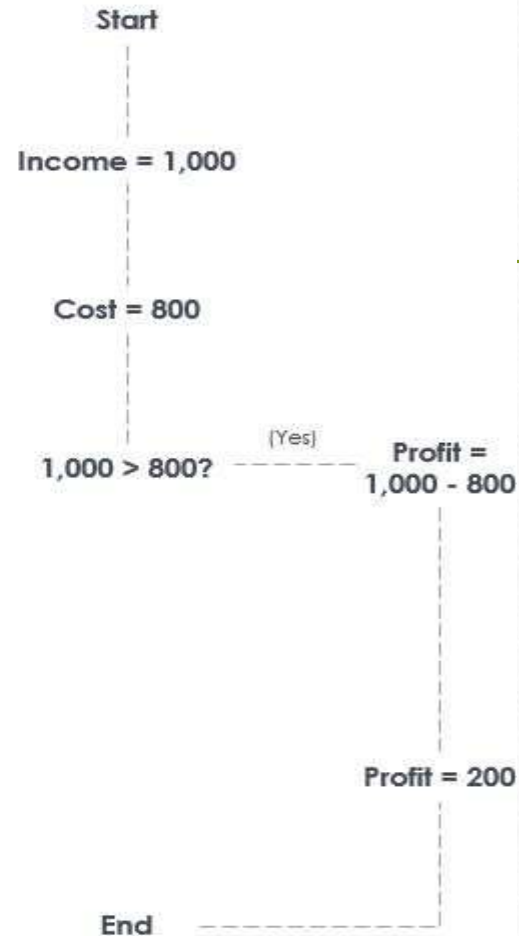
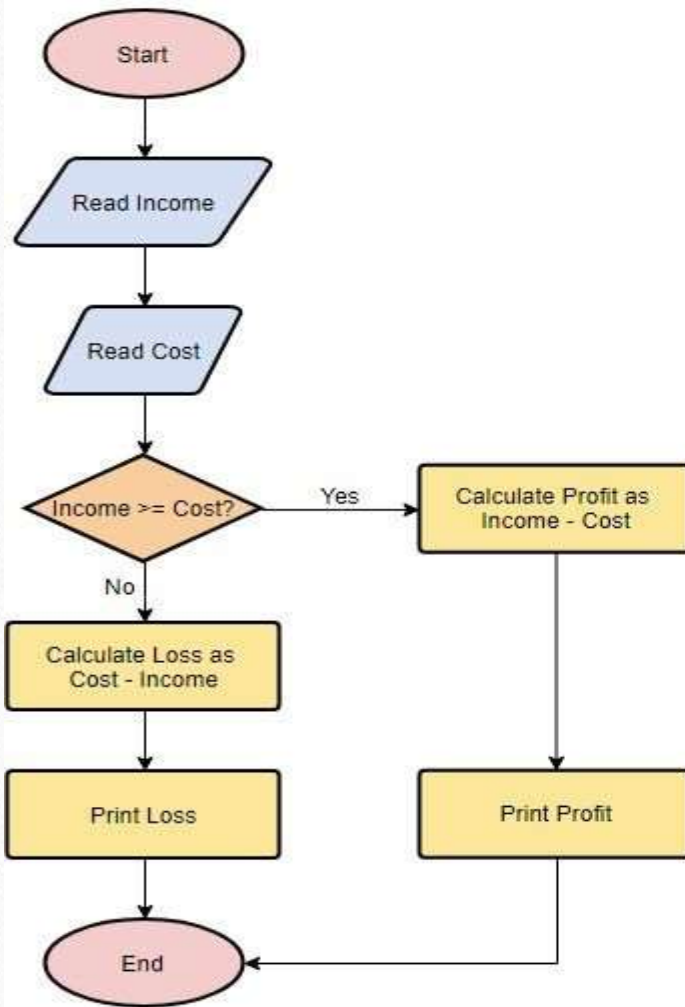
Find the sum of 529 and 256



Start
A = 529
B = 256
Sum = 529 + 256
Sum = 785
End

Sita went to a shop and purchased a saree and a handbag. Help the shopkeeper to find how much she is supposed to pay.

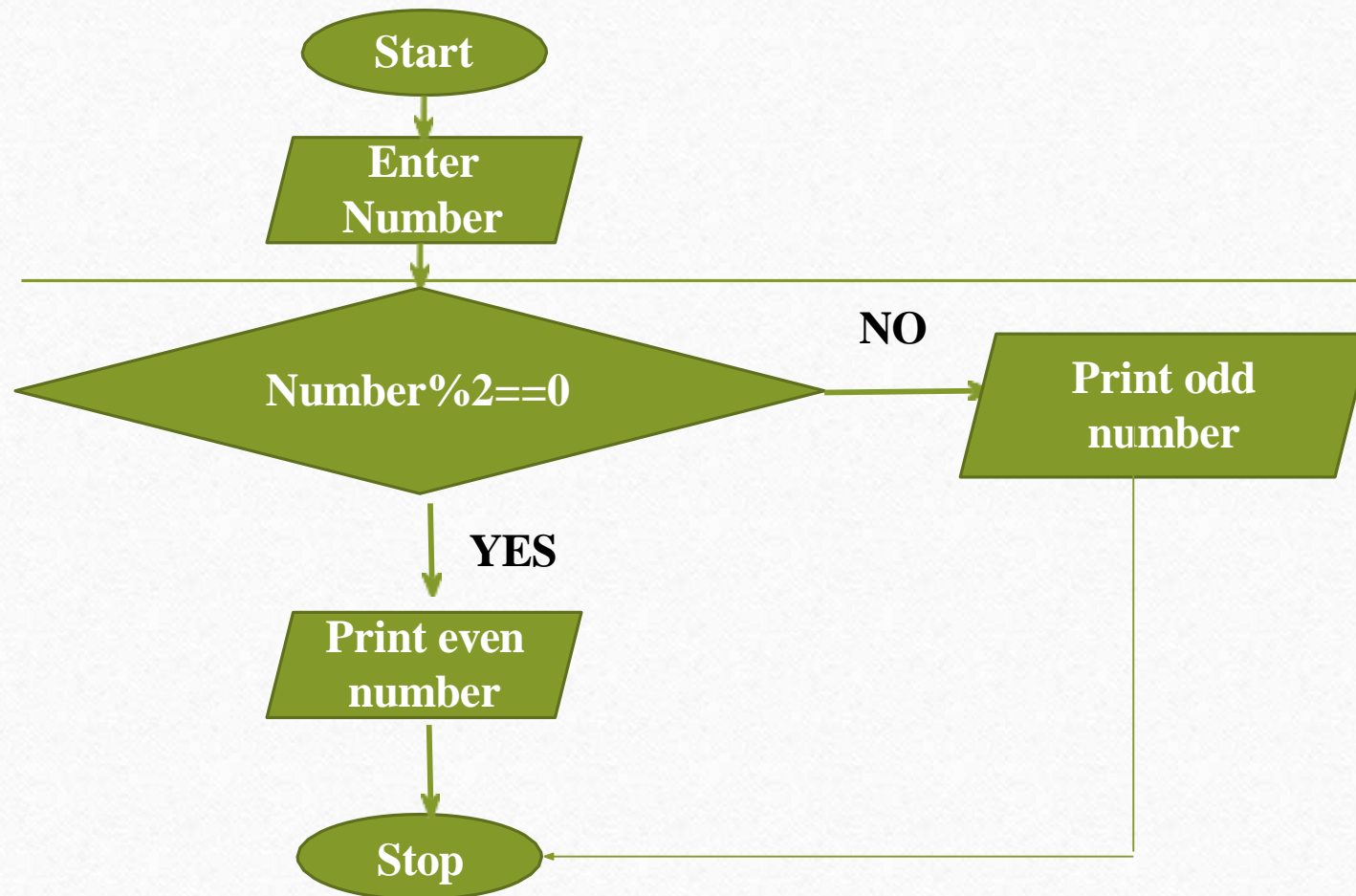
Find the profit/loss when
income = 1,000, cost = 800



Ramu is a shopkeeper. At the end of the day he calculating his expenses of that day. If his income is greater than his expenses(cost) he will have some profit. Otherwise he will end up with some loss. Display his profit / loss amount.

SOLVE IT!

Lets Try

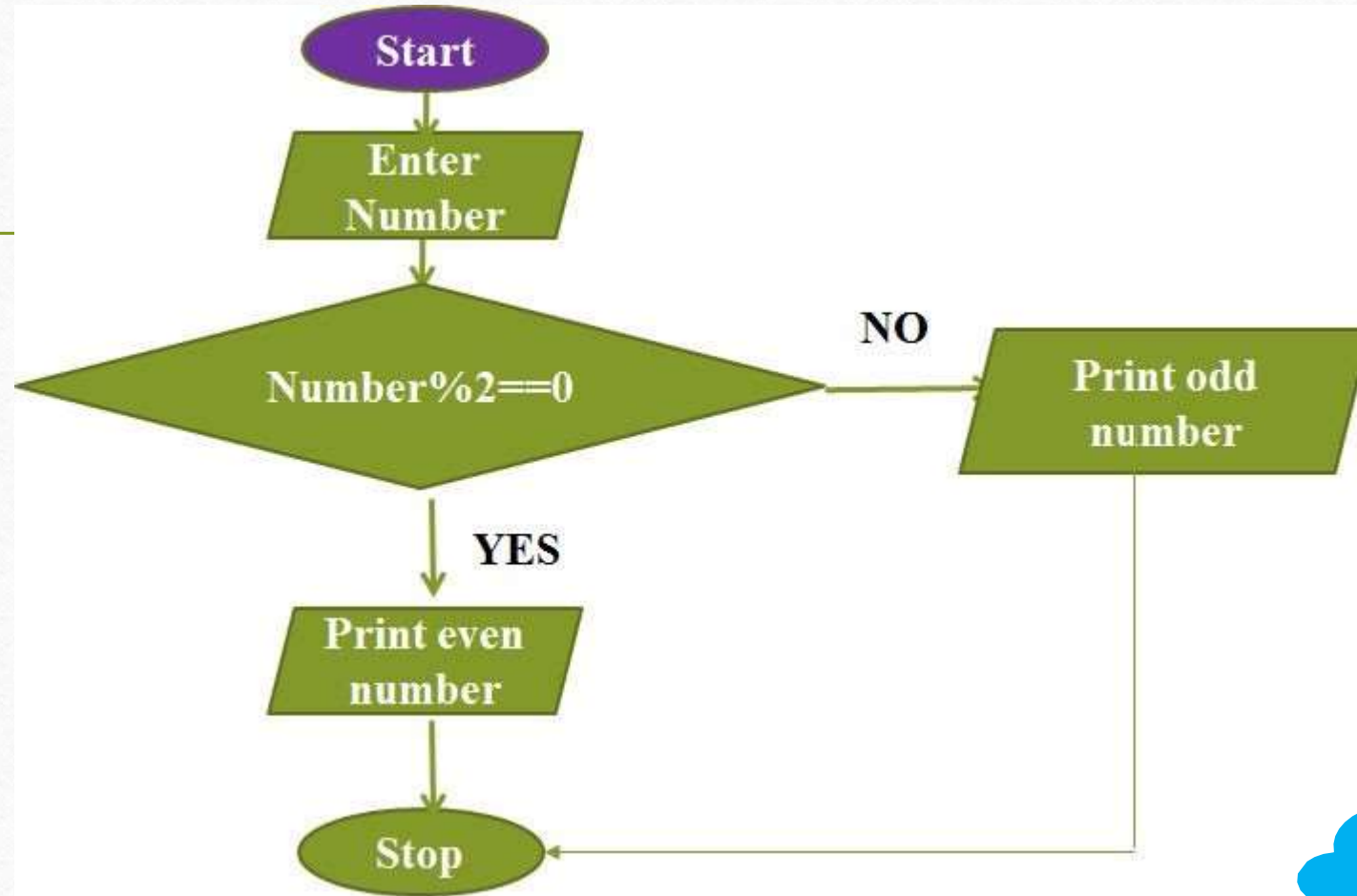


Number=8

Number=7



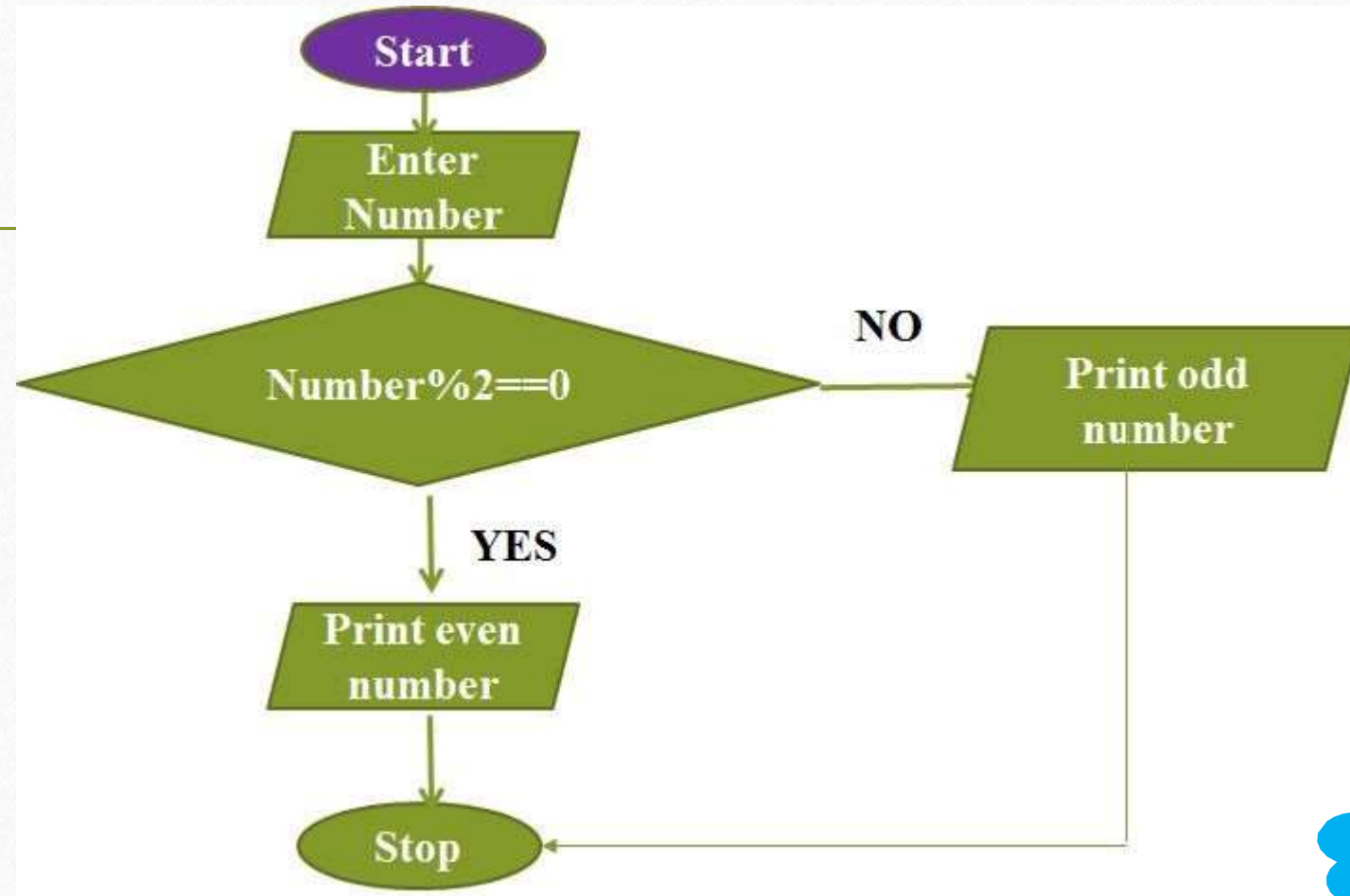
Number=8



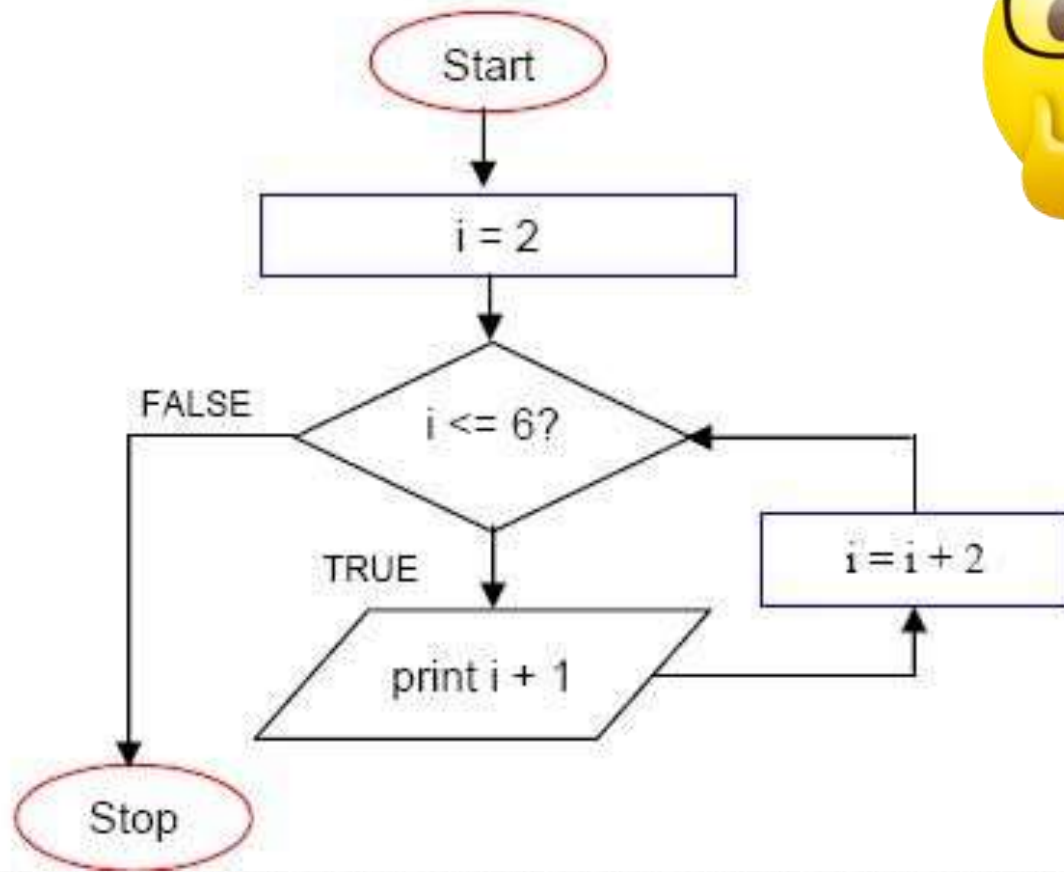
OUTPUT

even
number

Number=7



odd
number

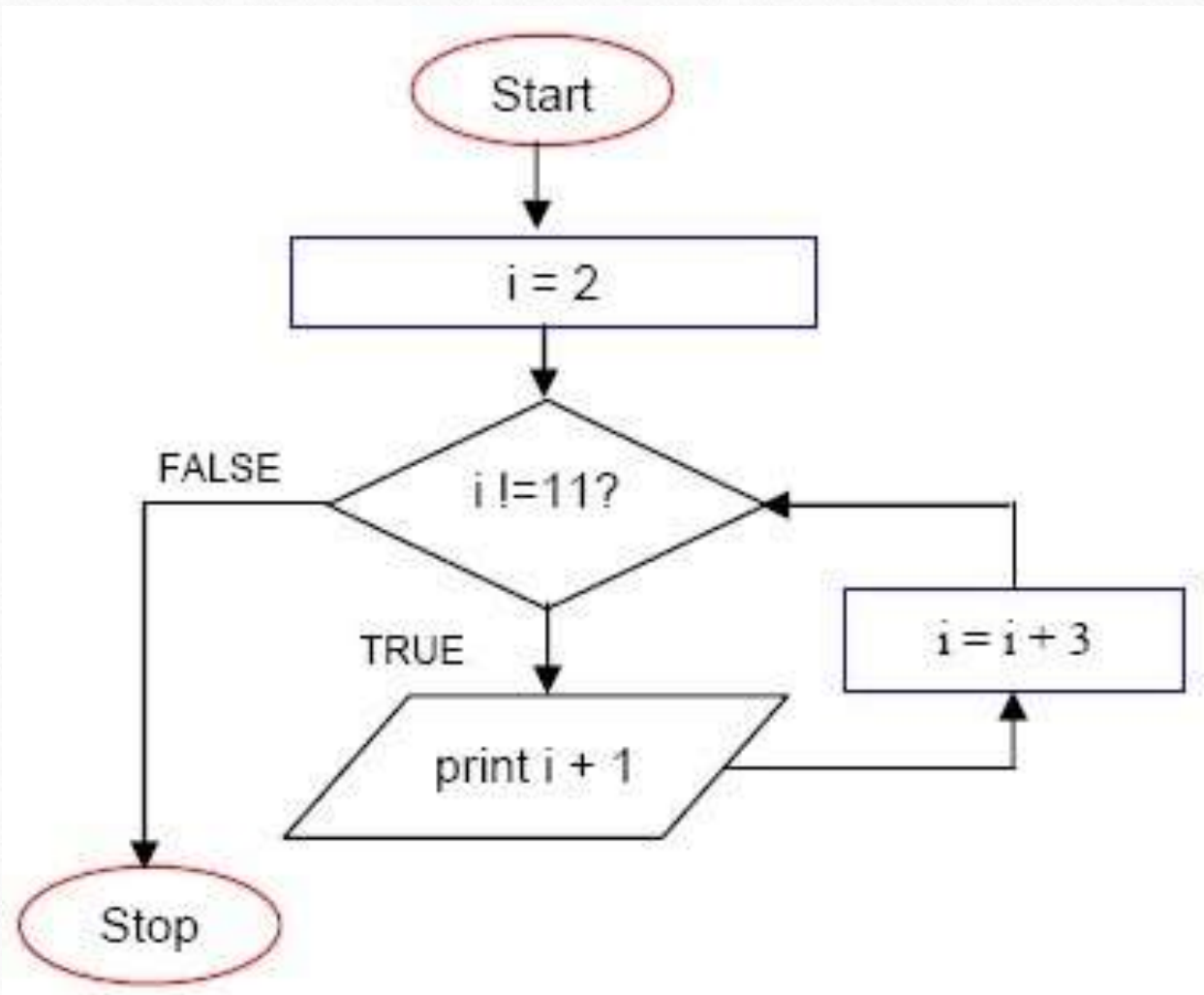


Guess the Output??

3

5

7

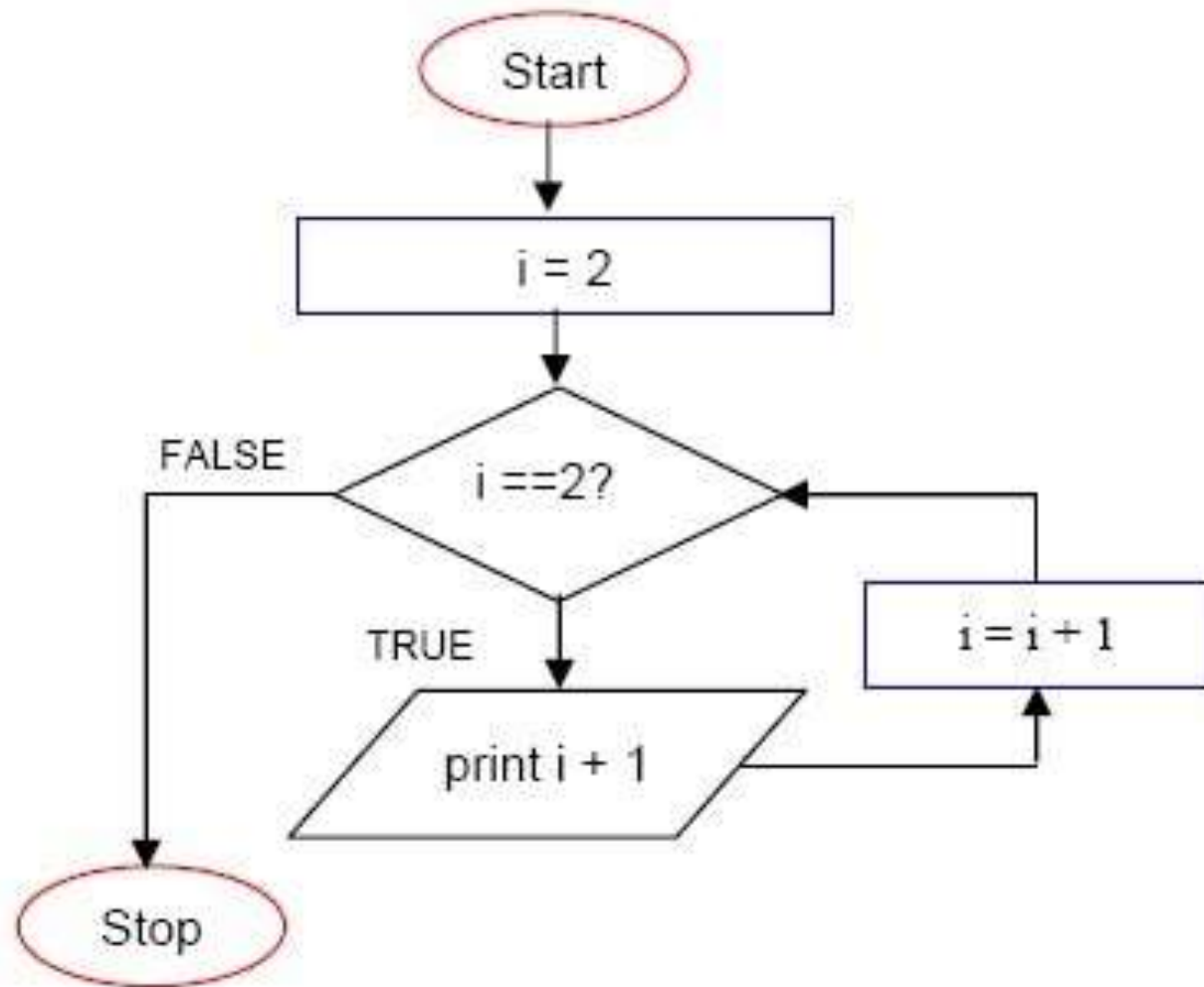


Guess the Output??

3

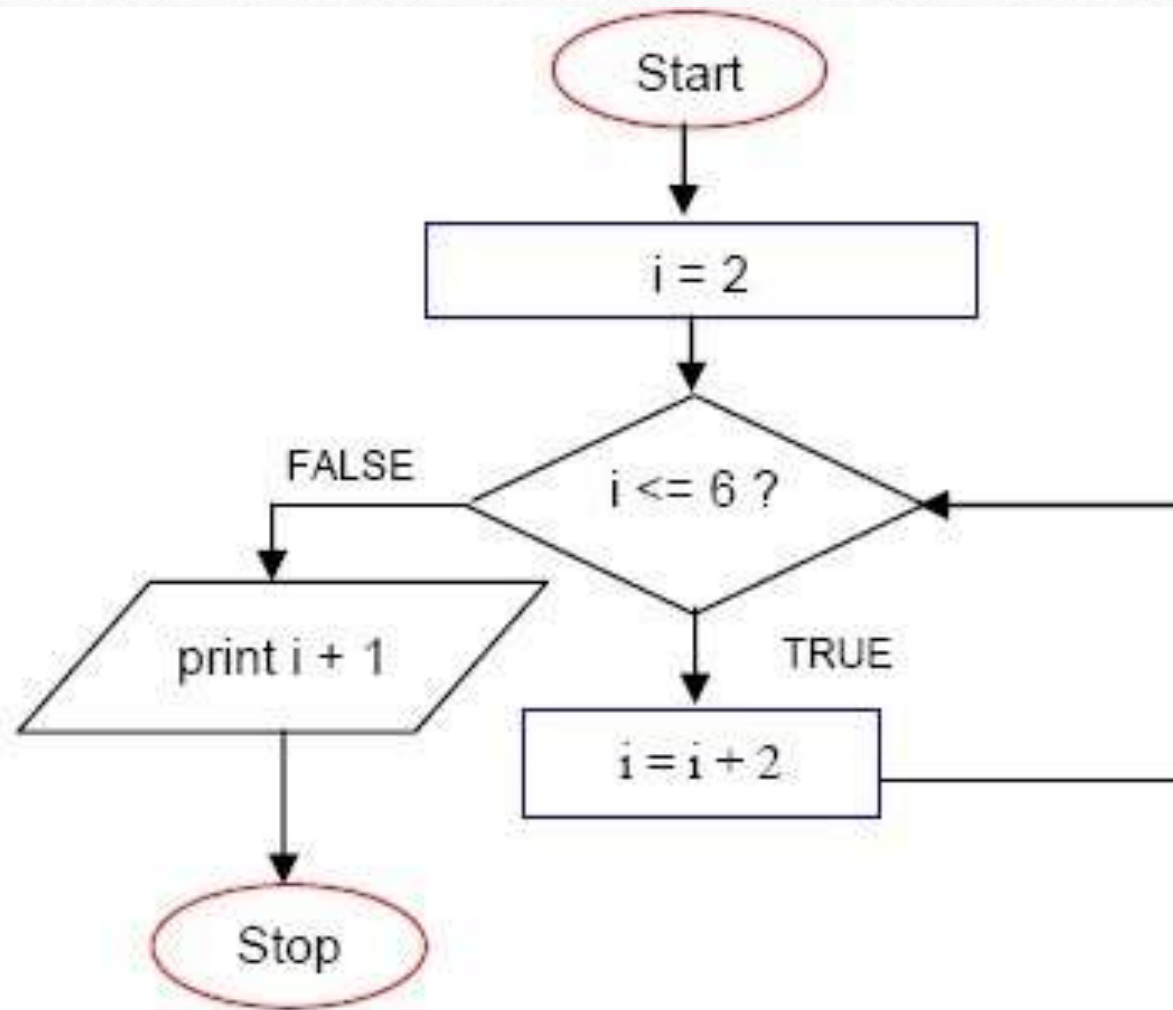
6

9



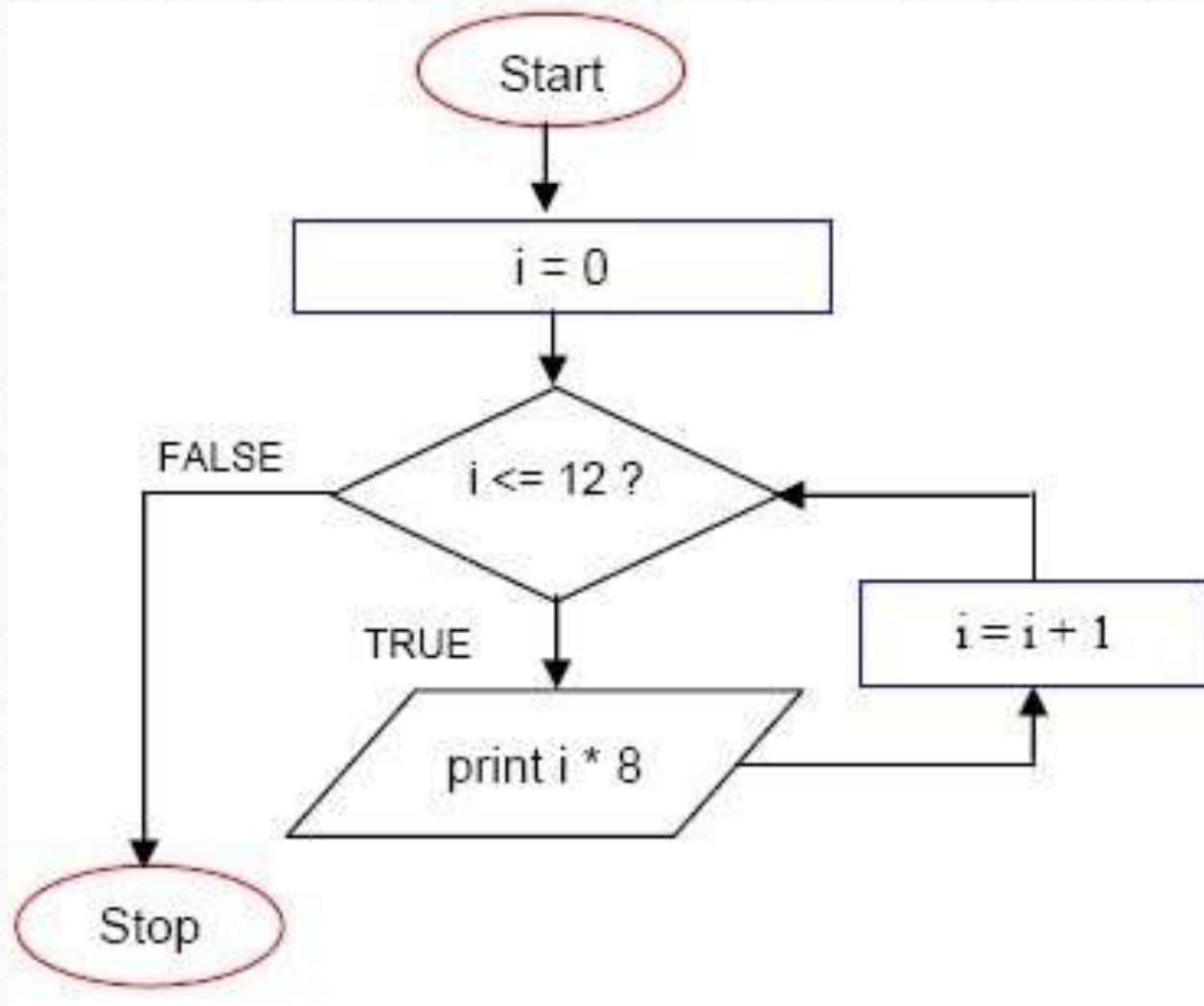
Guess the Output??

3



Guess the Output??

7



Guess the Output??

0 8 16
32 40 ...
... 96