

Algorithm and Flowchart

Pseudocode: is an artificial and informal language that helps programmers develop algorithms. Pseudocode is very similar to everyday English.

Algorithm:

An algorithm is a description of a procedure which terminates with a result. Algorithm is a step-by-step method of solving a problem.

Properties of an Algorithm:

- 1) Finiteness: - An algorithm terminates after a finite numbers of steps.
- 2) Definiteness: - Each step in algorithm is unambiguous. This means that the action specified by the step cannot be interpreted (explain the meaning of) in multiple ways & can be performed without any confusion.
- 3) Input: - An algorithm accepts zero or more inputs
- 4) Output:- An algorithm should produce at least one output.

5) Effectiveness: - It consists of basic instructions that are realizable. This means that the instructions can be performed by using the given inputs in a finite amount of time.

Writing an algorithm

An algorithm can be written in English, like sentences and using mathematical formulas. Sometimes algorithm written in English like language is Pseudo code.

Examples

- 1) Finding the average of three numbers
1. Let a,b,c are three integers
2. Let d is float
3. Display the message “Enter any three integers:”
4. Read three integers and stores in a,b,c
5. Compute the $d = (a+b+c)/3.0$
6. Display “The avg is:” , d
7. End.

- **Example 1:** Write an algorithm to determine a student’s final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.

Pseudocode::

- **Input a set of 4 marks**
- **Calculate their average by summing and dividing by 4**
- **if average is below 50**
 Print “FAIL”
 else
 Print “PASS”

■ **Detailed Algorithm :**

- **Step 1: Input M1,M2,M3,M4**
- Step 2: GRADE ← (M1+M2+M3+M4)/4**
- Step 3: if (GRADE < 50) then**
 Print “FAIL”
 else
 Print “PASS”
 endif





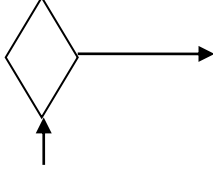

Flowcharts :

The pictorial representation of algorithm is called flowchart.

Uses of flow chart:

- 1 : flow chart helps to understand the program easily.
- 2 : as different symbols are used to specify the type of operation performed, it is easier to understand the complex programs with the help of flowcharts.

Flowchart Symbols

S.NO	Description	Symbols
1	Flowlines : These are the left to right or top to bottom lines connection symbols. These lines shows the flow of control through the program.	
2	Terminal Symbol : The oval shaped symbol always begins and ends the flowchart. Every flow chart starting and ending symbol is terminal symbol.	
3	Input / Output symbol : The parallelogram is used for both input (Read) and Output (Write) is called I/O symbol. This symbol is used to denote any function of an I/O device in the program.	
4	Process Symbol : The rectangle symbol is called process symbol. It is used for calculations and initialization of memory locations.	
5	Decision symbol : The diamond shaped symbol is called decision symbol. This box is used for decision making. There will be always two exists from a decision symbol one is labeled YES and other labeled NO.	
6	Connectors : The connector symbol is represented by a circle. Whenever a complex flowchart is morethan one page, in such a situation, the connector symbols are used to connect the flowchart.	

Algorithm to find whether a number even or odd:

Step1: Begin

Step2: Take a number

**Step3: if the number is divisible by2 then
print that number is even
otherwise print that number is odd**

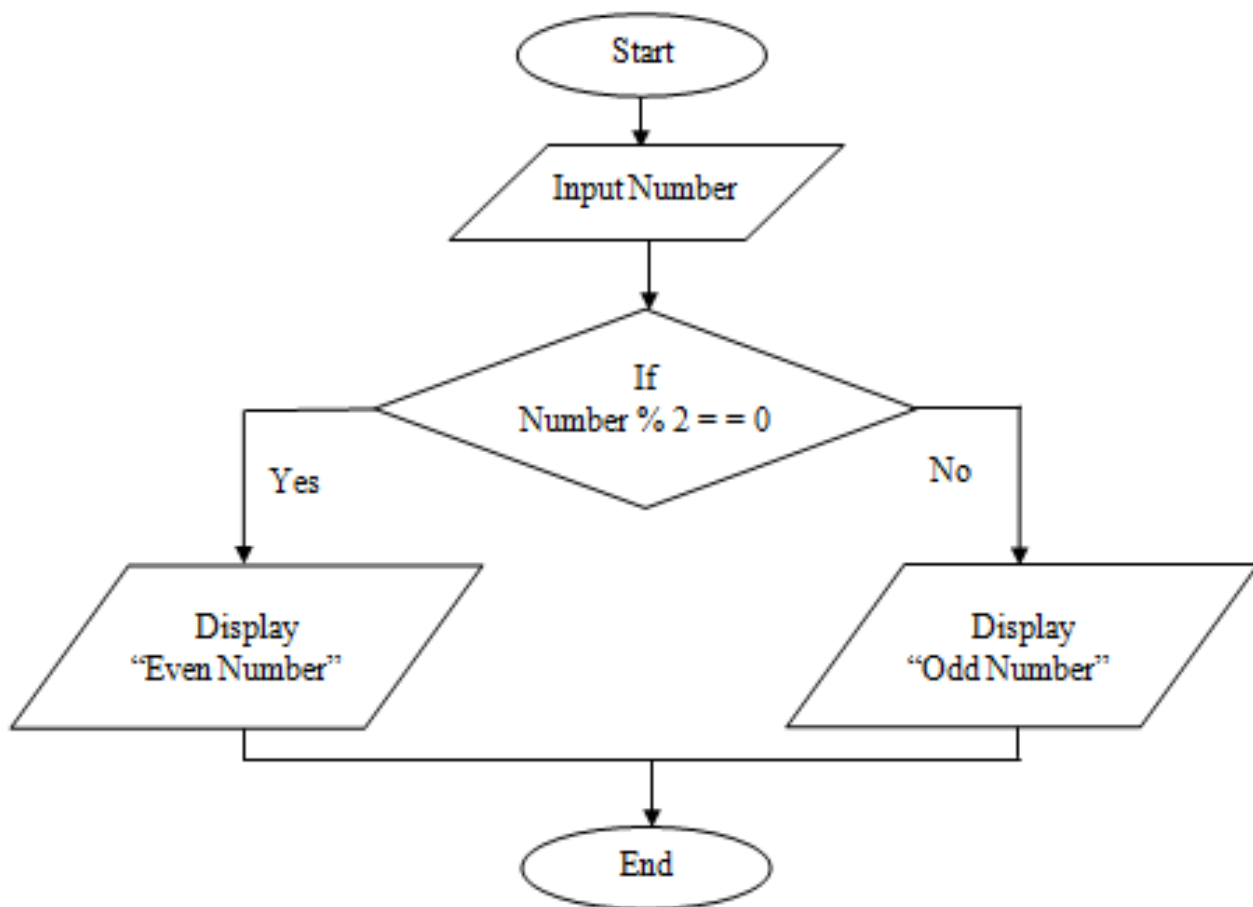
Step1: START

Step2: Read num

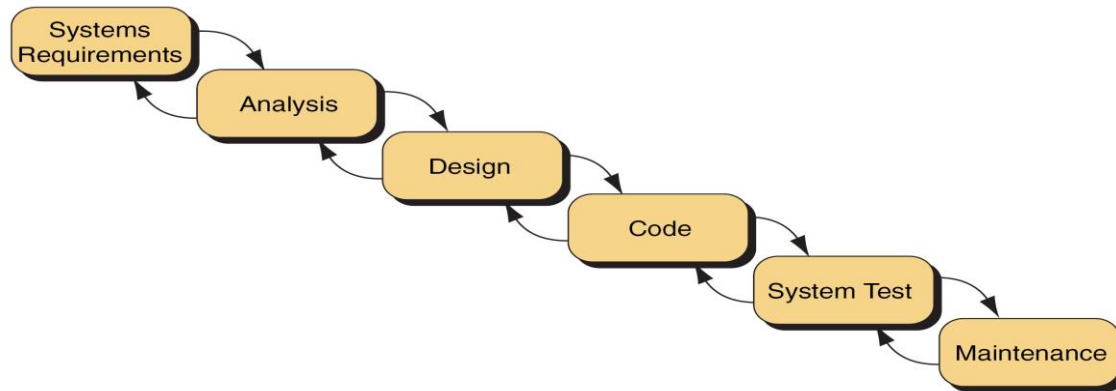
**Step3: if(num%2=0) then
print num is even
otherwise**

Step4: End
(Algorithm in natural language)

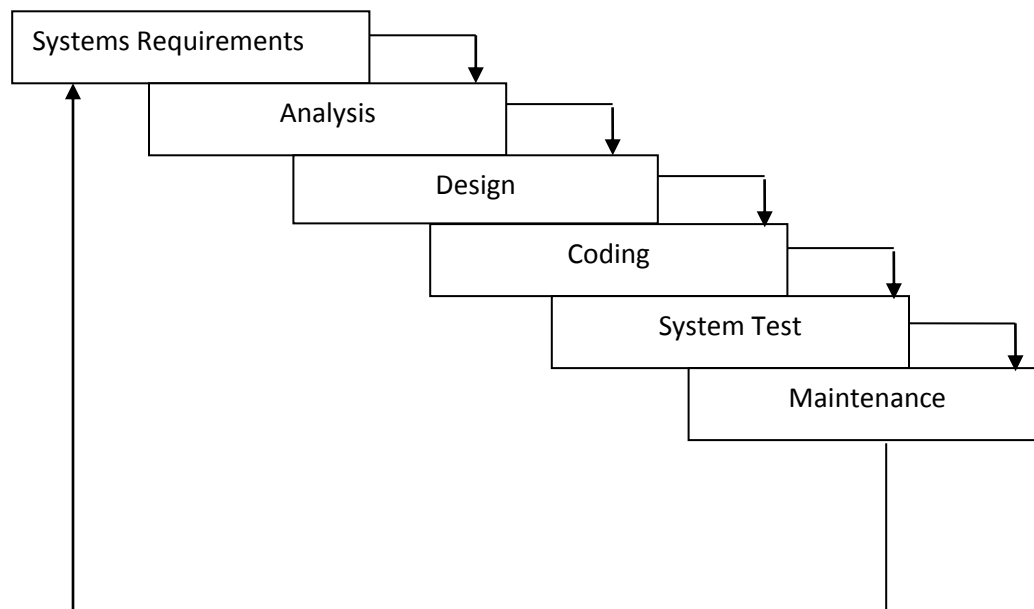
print num is odd
Step4: STOP
(Algorithm by using pseudo code)



System Development:



Or



1. Statement of Problem

- a) Working with existing system and using proper questionnaire, the problem should be explained clearly.
- b) What inputs are available, what outputs are required and what is needed for creating workable solution, should be understood clearly.

2. Analysis

- a) The method of solutions to solve the problem can be identified.
- b) We also judge that which method gives best results among different methods of solution.

3. Design

- a) Algorithms and flow charts will be prepared.
- b) Focus on data, architecture, user interfaces and program components.

4. System Test

The algorithms and flow charts developed in the previous steps are converted into actual programs in the high level languages like C.

a. Compilation

The process of translating the program into machine code is called as Compilation. Syntactic errors are found quickly at the time of compiling the program. These errors occur due to the usage of wrong syntaxes for the statements.

Eg: $x=a*y+b$

There is a syntax error in this statement, since, each and every statement in C language ends with a semicolon (;).

b. Execution

The next step is Program execution. In this phase, we may encounter two types of errors. Runtime Errors: these errors occur during the execution of the program and terminate the program abnormally.

Logical Errors: these errors occur due to incorrect usage of the instructions in the program. These errors are neither detected during compilation or execution nor cause any stoppage to the program execution but produces incorrect output.

5. Maintenance

We are maintenance the software by updating the information, providing the security and license for the software.