

1. What is algorithm?

A finite set of steps that must be followed to solve any problem is called an **algorithm**. Algorithm is generally developed before the actual coding is done. It is written using English like language so that it is easily understandable even by non-programmers.

Sometimes algorithms are written using **pseudocodes**, i.e. a language similar to the programming language to be used. Writing algorithm for solving a problem offers these advantages –

- Promotes effective communication between team members
- Enables analysis of problem at hand
- Acts as blueprint for coding
- Assists in debugging
- Becomes part of software documentation for future reference during maintenance phase

These are the characteristics of a good and correct algorithm –

- Has a set of inputs
- Steps are uniquely defined
- Has finite number of steps
- Produces desired output

Example Algorithms

Let us first take an example of a real-life situation for creating algorithm. Here is the algorithm for going to the market to purchase a pen.

1. Get dressed to go to the market.
2. Check your wallet for money.
3. If there is no money in the wallet, replenish it.
4. Go to the shop.
5. Ask for your favorite brand of pen.
6. If pen is not available, go to step 7 else go to step 10
7. Give money to the shopkeeper.
8. Keep the purchased pen safely.
9. Go back home.
10. Ask for any other brand of pen.
11. Go to Step 7.

Step 4 in this algorithm is in itself a complete task and separate algorithm can be written for it. Let us now create an algorithm to check whether a number is positive or negative.

1. Print "Give any number"
2. Read num
3. if (num==0) print "You entered 0"
4. if (num>0) print "You entered a positive number"
5. if (num<0) print "You entered a negative number"

Algorithm for SI

Step 1:- Start

Step 2:- Enter p, r, t, SI

Step 3:- Evaluate $p * R * T / 100$ & store it in SI

Step 4:- Display SI




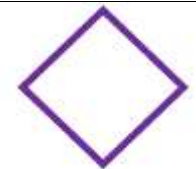

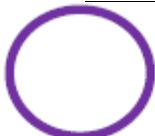

Step 5:- Stop

2. What is flowchart?

Flowchart is a diagrammatic representation of sequence of logical steps of a program. Flowcharts use simple geometric shapes to depict processes and arrows to show relationships and process/data flow.

Flowchart Symbols

Here is a chart for some of the common symbols used in drawing flowcharts.

Symbol	Symbol Name	Purpose
	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
	Process	Indicates processes like mathematical operations.
	Input/ Output	Used for denoting program inputs and outputs.
	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	Arrow	Shows relationships between different shapes.
	On-page Connector	Connects two or more parts of a flowchart, which are on the same page.
	Off-page Connector	Connects two parts of a flowchart which are spread over different pages.

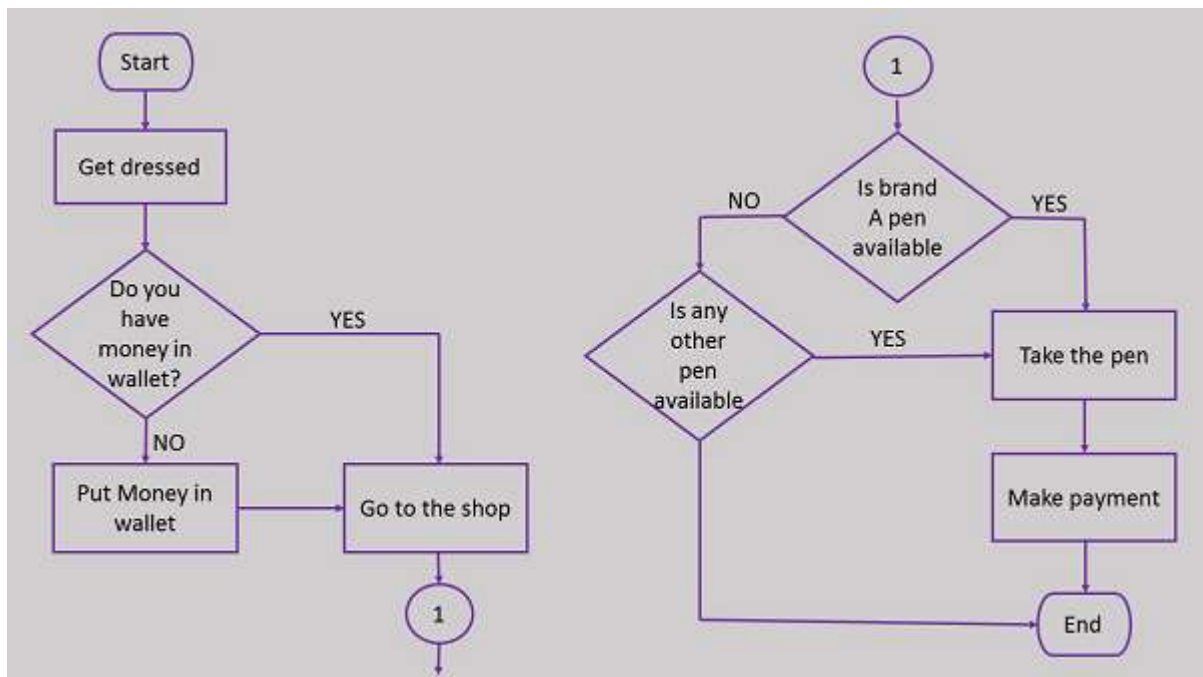
Guidelines for Developing Flowcharts

These are some points to keep in mind while developing a flowchart –

- Flowchart can have only one start and one stop symbol
- On-page connectors are referenced using numbers
- Off-page connectors are referenced using alphabets
- General flow of processes is top to bottom or left to right
- Arrows should not cross each other

Example Flowcharts

Here is the flowchart for going to the market to purchase a pen.



Here is a flowchart to calculate the average of two numbers.

