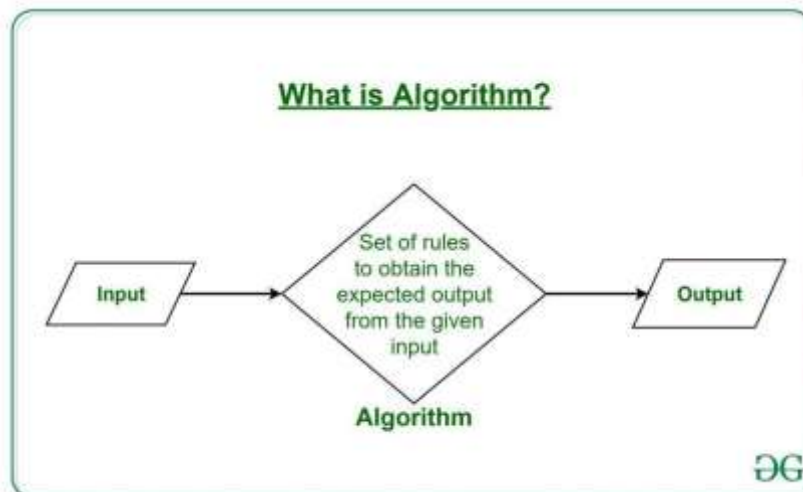




## Lecture 2: Algorithms

An algorithm is a clearly specified set of simple instructions to be followed to solve a problem. Once an algorithm is given for a problem and decided (somehow) to be correct, an important step is to determine how much in the way of resources, such as time or space, the algorithm will require. An algorithm that solves a problem but requires a year is hardly of any use. Likewise, an algorithm that requires thousands of gigabytes of main memory is not (currently) useful on most machines.

In general, an algorithm is a description of a procedure that terminates with a result. For example, the factorial of a number  $x$  is  $x$  multiplied by  $x-1$  multiplied by  $x-2$  and so on until it is multiplied by 1. The factorial of 6 is  $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$ . This is an algorithm that follows a set procedure and terminates in a result.



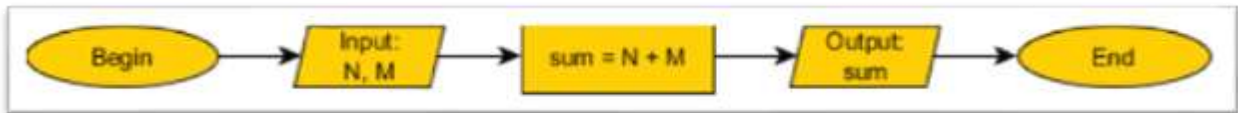
### The characteristics of an algorithm

- Be Well-defined and well-ordered- The instructions provided in an algorithm should be understandable and defined well.
- Each of the steps of an algorithm should be simple enough that it may not require any further simplification.

- Have an effectively computable operation.

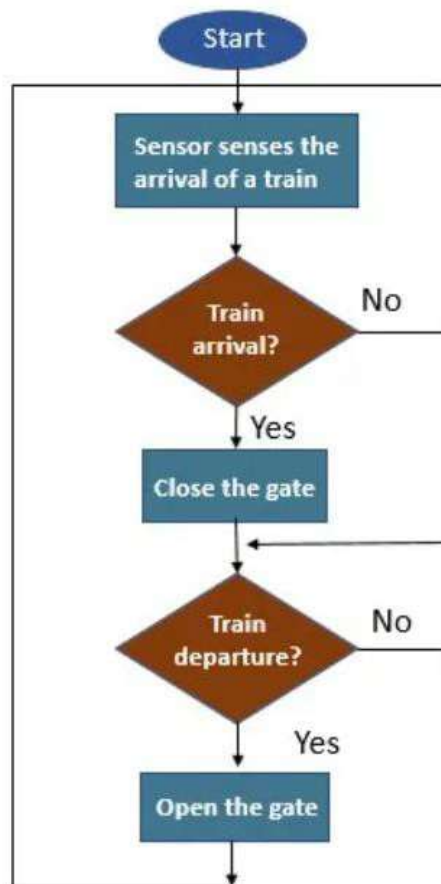
**Example1:**

1. Enter the two numbers in the variables N and M.
2. Sum them and save the result in the variable sum.
3. output the result.



**Example2:**

For instance, take an example of a railway gate point.



**Example 3:**

Devolve an algorithm that allows the user to enter the count of numbers in a list followed by these numbers. The algorithm should find and output the minimum and maximum numbers in the list.

An algorithm for this might be:

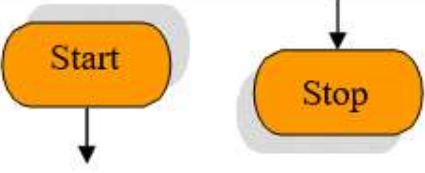

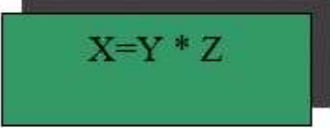
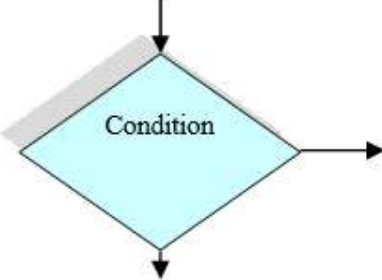
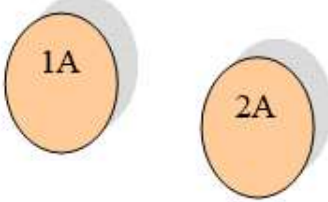
- Initialize.
- Get a count of numbers.
- Enter numbers and find the maximum and minimum.
- Output result.

The user might enter zero for the count. To deal with this case the above general case can be extended as follows to be an algorithm:

1. Initialize the required variables.
2. Get count of numbers.
3. If count is zero then exit.
4. Otherwise begin.
5. Enter numbers.
6. Find max and min.
7. Output result.
8. End.

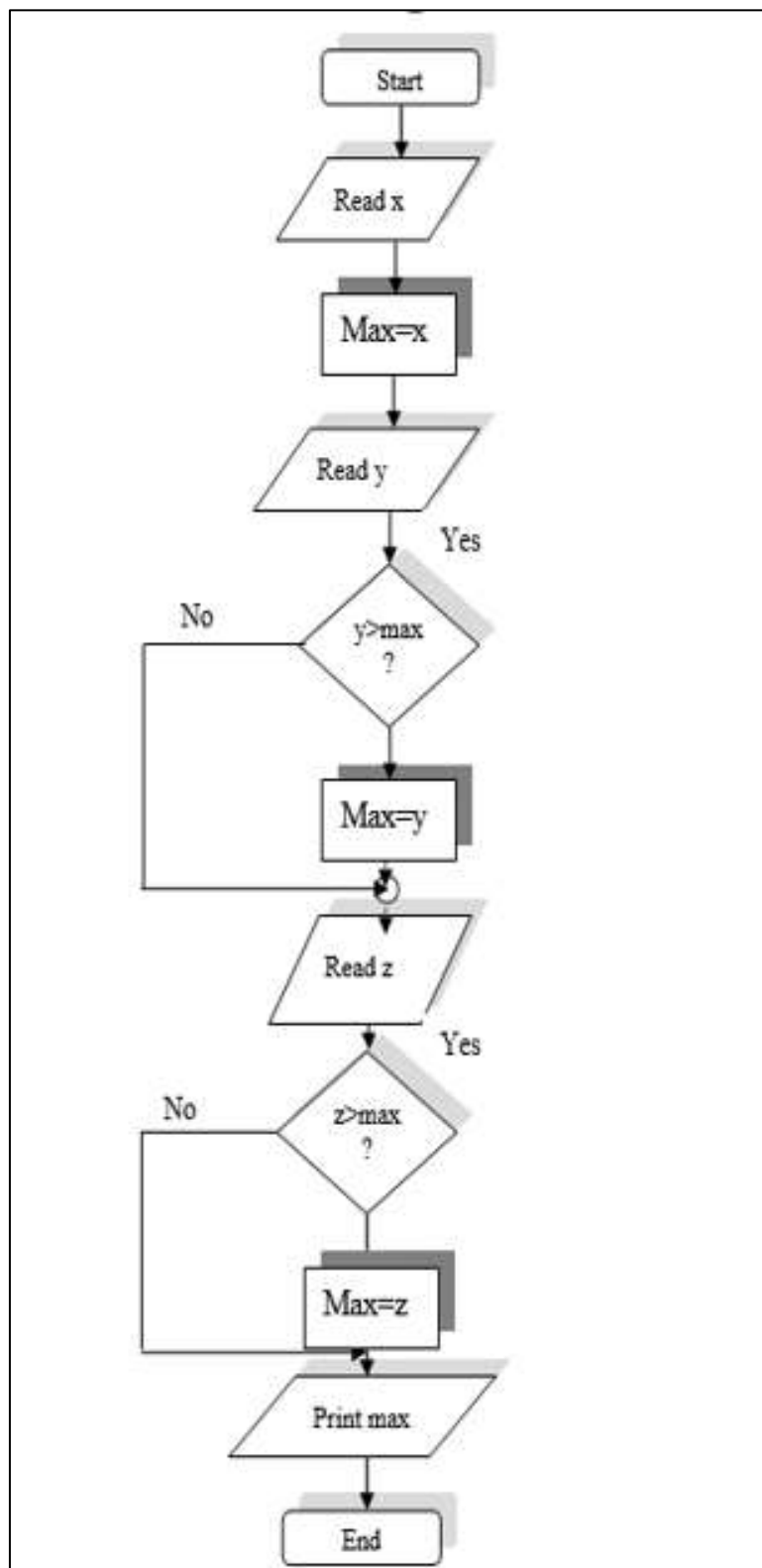
**Flowcharts**

A flowchart is a graphical representation of an algorithm or of a portion of an algorithm. Flowcharts are drawn using symbols. The main symbols used to draw a flowchart are shown in following figure.

	<p>Start and Stop Symbols</p>
	<p>Input and Output Symbols</p>
	<p>Mathematical and logical processing symbol</p>
	<p>Decision making symbol</p>
	<p>Connector symbols</p>

**Example 2:**

Draw a flowchart to read 3 numbers: x, y and z and print the largest number of them.



**Example 3:**

Draw the flowchart required to find the sum of negative numbers among 50 numbers entered by the user.

