**Algorithm (C++)**

In the [C++](https://en.wikipedia.org/wiki/C%2B%2B) [Standard Library](https://en.wikipedia.org/wiki/C%2B%2B_Standard_Library), **algorithms** are components that perform algorithmic operations on [containers](https://en.wikipedia.org/wiki/Container_%28data_structure%29) and other sequences.[[1]](https://en.wikipedia.org/wiki/Algorithm_%28C%2B%2B%29#cite_note-C.2B.2B03_25.2F1-1)

The C++ standard provides some standard algorithms collected in the <algorithm> standard header.[[2]](https://en.wikipedia.org/wiki/Algorithm_%28C%2B%2B%29#cite_note-2) A handful of algorithms are also in the <numeric> header. All algorithms are in the std [namespace](https://en.wikipedia.org/wiki/Namespace).

### Overview of algorithms in STL

STL provide different types of algorithms that can be implemented upon any of the container with the help of iterators. Thus now we don’t have to define complex algorithm instead we just use the built in functions provided by the algorithm library in STL.

As already discussed earlier, algorithm functions provided by algorithm library works on the iterators, not on the containers. Thus one algorithm function can be used on any type of container.

Use of algorithms from STL saves time, effort, code and are very reliable.

For example, for implementing binary search in C++, we would have to write a function such as :

bool **binary\_search**( int l , int r , int key ,int a[])

{

if(l > r)

return -1;

else

{

int mid=(l+r)/2;

if(a[mid] == key) {

return true;

}

else if(a[mid] > key) {

return **binary\_search**(l, mid-1, key, a);

}

else if(a[mid] < key) {

return **binary\_search**(mid+1, r, key, a);

}

}

}

Note that the above function will work only if the array is of intergers and characters.

But in STL we can just use the binary\_search() provided by the algorithm library to perform binary search. It is already defined in the library as :

return binary\_search(a, a+a.size())

Plus the above function will work on any type of container.

#### Types of algorithms in algorithm library

1. Sorting Algorithms
2. Search algorithms
3. Non modifying algorithms
4. Modifying algorithms
5. Numeric algorithms
6. Minimum and Maximum operations.

# Flowchart In Programming

Flowchart is a diagrammatic representation of an algorithm. Flowchart are very helpful in writing program and explaining program to others.

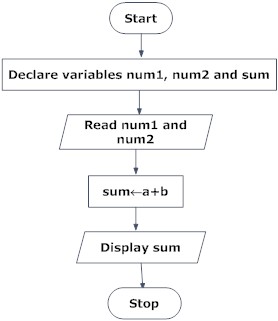
## Symbols Used In Flowchart

Different symbols are used for different states in flowchart, For example: Input/Output and decision making has different symbols. The table below describes all the symbols that are used in making flowchart

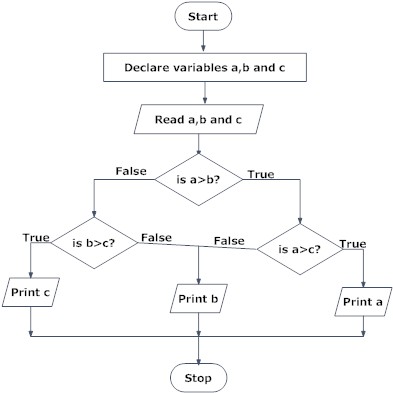
| **Symbol** | **Purpose** | **Description** |
| --- | --- | --- |
| Flowline symbol in flowchart of programming | Flow line | Used to indicate the flow of logic by connecting symbols. |
| Terminal symbol in flowchart of programming | Terminal(Stop/Start) | Used to represent start and end of flowchart. |
| Input/Output symbol in flowchart of programming | Input/Output | Used for input and output operation. |
| Processing symbol in flowchart of programming | Processing | Used for airthmetic operations and data-manipulations. |
| Decision making symbol in flowchart of programming | Decision | Used to represent the operation in which there are two alternatives, true and false. |
| On-page connector symbol in flowchart of programming | On-page Connector | Used to join different flowline |
| Off-page connector symbol in flowchart of programming | Off-page Connector | Used to connect flowchart portion on different page. |
| Predefined process symbol in flowchart of programming | Predefined Process/Function | Used to represent a group of statements performing one processing task. |

### Examples of flowcharts in programming

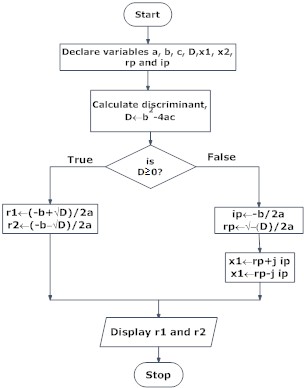
**Draw a flowchart to add two numbers entered by user.**



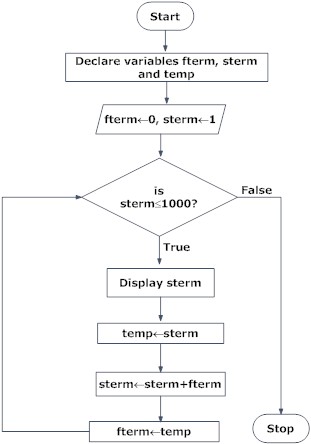
**Draw flowchart to find the largest among three different numbers entered by user.**



**Draw a flowchart to find all the roots of a quadratic equation ax2+bx+c=0**



**Draw a flowchart to find the Fibonacci series till term≤1000.**



Though, flowchart are useful in efficient coding, debugging and analysis of a program, drawing flowchart in very complicated in case of complex programs and often ignored.

# Flowchart Symbols in Programming: Definition, Functions & Examples

Much like symbols on a map can tell us what action to take, flowchart symbols represent the various actions of a computer program. In this lesson, we'll learn why these symbols are necessary and go over some of the most basic types used by programmers.

## What Are Flowchart Symbols?

As we go about our daily lives, we are constantly seeing symbols. For instance, even without words, we know the red octagon shaped sign means STOP. With so many symbols in the world, have you ever wondered about the symbols with lines and arrows connecting them that programmers seem to find fascinating? Well, these are flowchart symbols.

**Flowchart symbols** are specific shapes used to create a visual representation of a program.

Some of the basic flowchart symbols can be seen here:

|  |
| --- |
| Basic Flowchart Symbols |

## Functions of Flowchart Symbols

Each symbol has its own function within the program. Each symbol represents a piece of the code written for the program. The **start/end** symbol can be used to represent either the beginning or ending of a program. The symbol for **process** allows you to show how the program is functioning, like when you need the program to calculate two numbers or even analyze the information.

When you decide to enter data, show it on the screen, or print it to paper, you use the **input/output** symbol. The **display** symbol signifies that information is displayed to the user.

Next you'll find a brief example of a program that allows you to input your name, then the computer will display it on the screen.

|  |
| --- |
| Basic Flowchart to Display Your Name |
| Basic Flowchart to Display Your Name |

There are many other symbols frequently used in flowcharts. The **decision** symbol is used for things like 'if statements,' where you must choose an option based on a specified criteria. A decision question may be something as simple as: if the grade is at least 70 then send out the message 'Passing' to the screen. Otherwise, send 'Failing' to the screen.

Sometimes when you're drawing a flowchart, you may need to leave from a portion of the chart and restart somewhere else, causing a break in the flow of the chart. For instance, you may get to the bottom of a column and need to go back to the top of the page to continue with a new column. Adding the **connector** symbol with a number in it at the exiting and re-entry points with the same number in them helps the reader understand the flow and continue following the program logic correctly.