

Flowchart

A **flowchart** is a visual representation of the steps involved in solving a problem or executing a process. It uses symbols to represent different types of operations and arrows to show the flow of control from one step to the next. Flowcharts are widely used in programming to help plan and visualize the logic of a program before writing the actual code.

Common Flowchart Symbols:

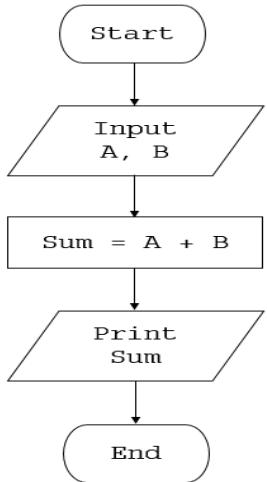
1. **Oval (Start/End):** Represents the start or end of the process.
2. **Rectangle (Process):** Represents an operation or action, such as a calculation or a command.
3. **Diamond (Decision):** Represents a decision point, where the flow can branch based on a condition.
4. **Parallelogram (Input/Output):** Represents input or output operations, such as reading from the user or displaying results.

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

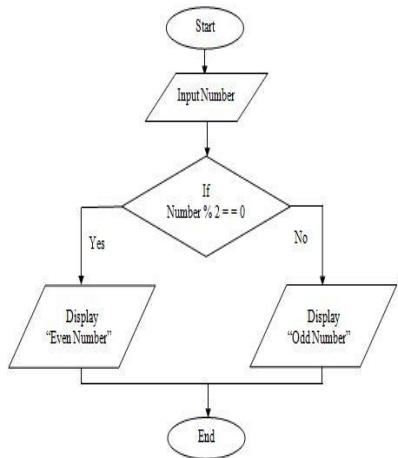
**Example:

Example:01

PRINT SUM OF 2 NUMBERS



Example:02



Algorithm in C Language

An **algorithm** is a step-by-step procedure or set of rules designed to perform a specific task or solve a particular problem. In the context of C programming, an algorithm outlines the logical steps needed to achieve the desired outcome.

Example: Algorithm to Find the Largest of Three Numbers

Step 1: Start

Step 2: Declare variables `a`, `b`, `c`, and `largest`.

Step 3: Input the values of `a`, `b`, and `c`.

****Step 4:**** If `a > b` and `a > c`, then:

- `largest = a`

****Step 5:**** Else, if `b > c`, then:

- `largest = b`

****Step 6:**** Else:

- `largest = c`

****Step 7:**** Print the value of `largest`.

****Step 8:**** End

C Program Based on the Above Algorithm

```
```c
#include <stdio.h>

int main() {
 int a, b, c, largest;

 // Input values
 printf("Enter three numbers: ");
 scanf("%d %d %d", &a, &b, &c);

 // Compare and find the largest number
 if (a > b && a > c) {
 largest = a;
 } else if (b > c) {
 largest = b;
 } else {
 largest = c;
 }

 // Output the largest number
 printf("The largest number is: %d\n", largest);

 return 0;
}
```

### **Assignment**

Assignment 1: Flowchart & Algorithm

Question:

Create a flowchart and write an algorithm to add two numbers entered by the user.

Assignment 2: Flowchart & Algorithm

Question:

Design a flowchart and write an algorithm to find the larger of two numbers entered by the user.

### Assignment 3: Flowchart & Algorithm

Question:

Create a flowchart and write an algorithm to check if a number entered by the user is positive, negative, or zero.