



Al-Mustaqbal University
College of Sciences
Department of Cybersecurity
المرحلة الاولى - اساسيات البرمجة

كلية العلوم قسم الأمن السيبراني

Subject: Programming Fundamentals

First Stage

Assistant Lecturer: Jaber Baqer Al-Hamdani

Lecture (2)

Algorithms and Flowcharts



1. Introduction to Algorithms

An **algorithm** is a finite set of precise steps or instructions designed to solve a problem or perform a specific task. It forms the foundation of programming, guiding the logical flow of a program.

Properties of a Good Algorithm

1. **Finiteness:** An algorithm must always terminate after a finite number of steps.
2. **Non-ambiguity:** Each step must be precisely defined, ensuring there is no ambiguity.
3. **Effectiveness:** All steps must be simple and directly executable.

2. Writing an Algorithm

An algorithm is often expressed in **plain language or pseudocode**. Let's consider an example:

Example 1: Algorithm to Calculate the Average of a Series of Numbers

1. Initialize the **sum** to 0 and the **count** to 0.
2. Read the first number.
3. While there are more numbers:
 - o Add the number to the **sum**.
 - o Increment the **count**.
 - o Read the next number.
4. Calculate the average by dividing the **sum** by the **count**.
5. Output the average.

Example 2: Algorithm to Find the Maximum and Minimum of a List of Numbers

1. Initialize **max** and **min** variables.
2. Read the first number as both the initial **max** and **min**.
3. For each subsequent number:
 - o If the number is **greater than** **max**, update **max**.
 - o If the number is **smaller than** **min**, update **min**.
4. Output the **max** and **min** values.








3. Introduction to Flowcharts

A **flowchart** is a graphical representation of an algorithm, using symbols to depict the steps and their order of execution. It is often used to visualize the logic of a program before writing the actual code.

Common Flowchart Symbols

1. **Oval**: Represents the start or end of a process.
2. **Arrow**: Shows the flow of control.
3. **Parallelogram**: Represents input or output operations.
4. **Rectangle**: Represents a process or instruction.
5. **Diamond**: Represents a decision-making step.

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.

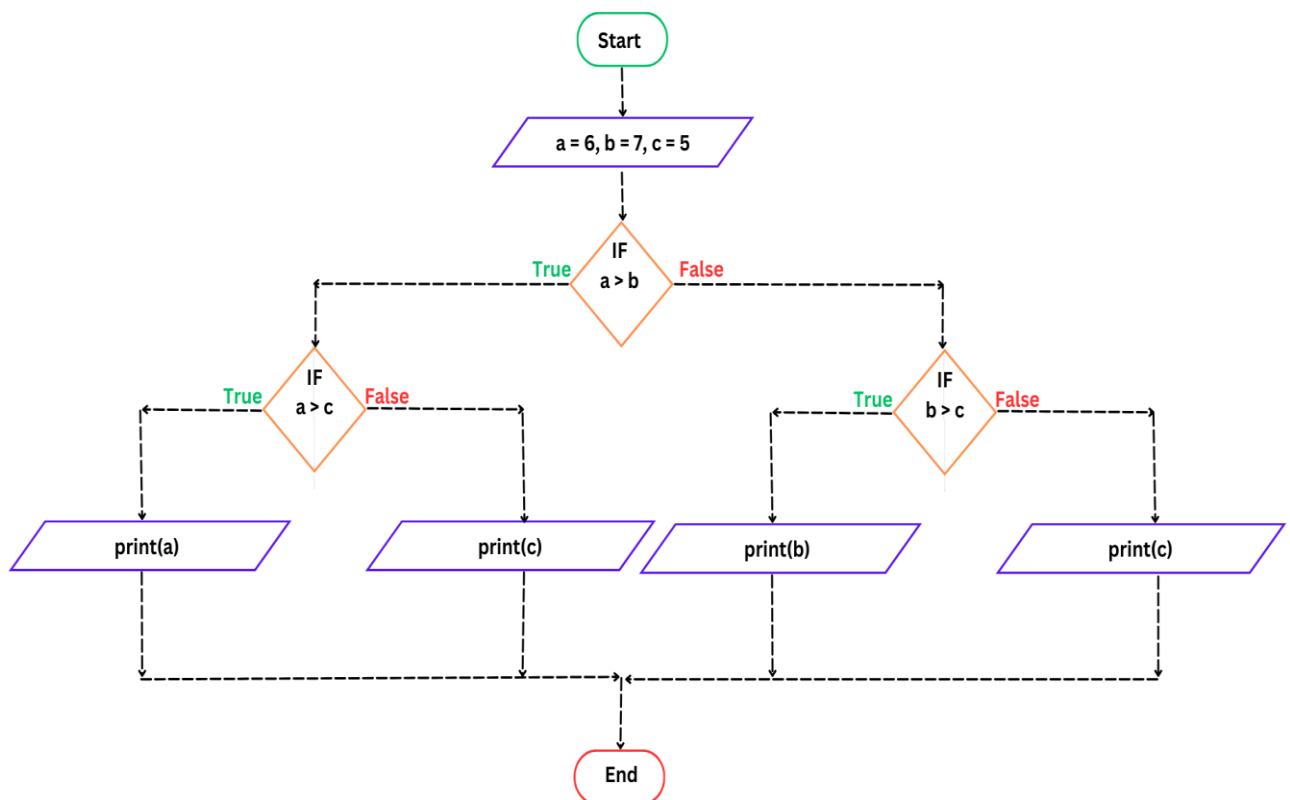
Example 1: Flowchart to Find the Largest of Three Numbers

1. **Start**: Begin the program execution.
2. **Input**: Initialize three variables a, b, and c with respective values or prompt the user to input them.
3. **Decision 1**: Check if $a > b$:
 - **True**: Proceed to Decision 2.
 - **False**: Move to Decision 3.
4. **Decision 2**: Check if $a > c$:



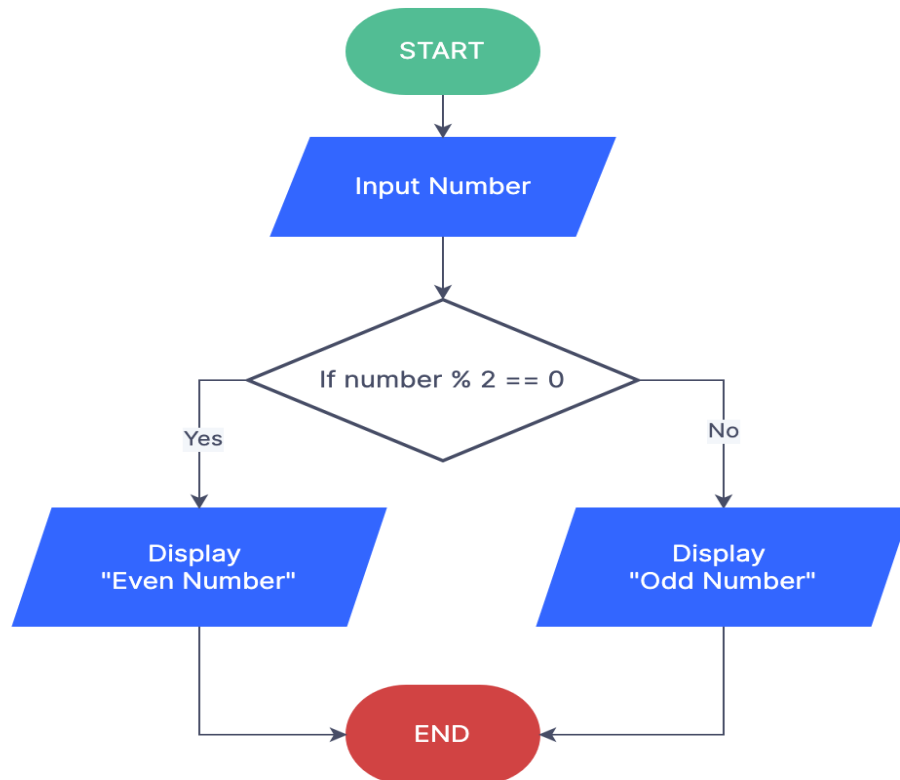
- **True:** Print a as the largest number.
- **False:** Print c as the largest number.
- 5. **Decision 3:** Check if $b > c$:
 - **True:** Print b as the largest number.
 - **False:** Print c as the largest number.
- 6. **End:** Terminate the program.

Flowchart Representation:





Example 2: Flowchart to determines whether a given number is even or odd.



5. Exercises

- Write an **algorithm** to calculate the area of a circle given its radius.
- Create a **flowchart** to check whether a given year is a leap year.