

Chapter 2: Flowcharts and Algorithm Design

CSC 111 – Computer Programming I

Dr. Sofien GANNOUNI

Week 1

Department of Computer Science

Lecture 3: Designing Algorithms Visually

Recall Step 2 of problem solving: **planning the solution**.

- We plan *before* we code, using tools that are independent of any programming language.
- Two such tools:
 - **Flowcharts** — a diagram showing the flow of steps.
 - **Pseudocode** — structured, plain-language steps.
- Both force you to think through the logic **before** worrying about Python syntax.

Designing Algorithms: A General Process

1. Read the problem statement carefully.
2. List the **inputs** and the desired **output**.
3. Write the steps in plain language (a first rough pseudocode).
4. Refine the pseudocode into precise, unambiguous steps.
5. Draw the corresponding flowchart.
6. **Trace** the flowchart by hand with sample values to verify correctness.

Flowcharts and Pseudocode

Flowchart

A **flowchart** is a diagram that represents an algorithm using standardized shapes connected by arrows that show the order of execution.

Pseudocode

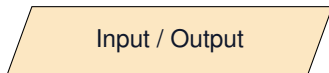
Pseudocode is an informal, structured description of an algorithm using everyday language and simple keywords (IF, WHILE, INPUT...).

- There are four basic types of shapes:
 - start/end
 - input/output
 - process
 - decision
- Arrows show **control flow** — which step happens next.
- Extremely useful for visualizing decisions and loops before coding them.

Flowchart Symbols and Conventions



Terminator



Parallelogram



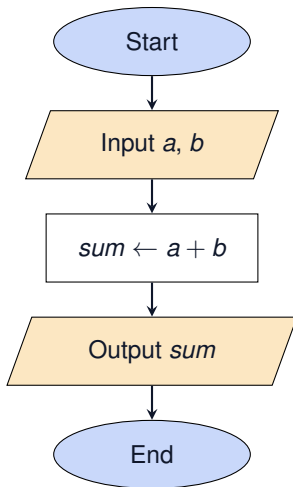
Rectangle



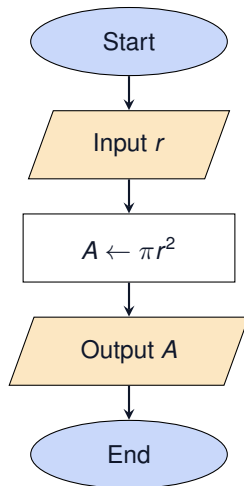
Diamond

- Arrows always show the direction of flow (never leave a shape unconnected).
- Every flowchart has exactly **one** Start and at least **one** End.
- A decision shape always has (at least) two exits: e.g., **Yes/No**.

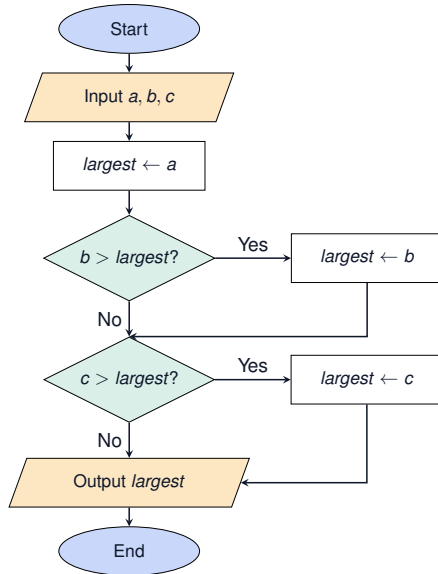
Example 1: Sum of Two Numbers a, b



Example 2: Calculate area of a circle $A = \pi r^2$



Example 3: Largest of Three Numbers a, b, c



What is Pseudocode?

Definition

Pseudocode is an informal, structured description of an algorithm using everyday language and simple keywords (IF, WHILE, INPUT...).

- No language-specific syntax — just clear, ordered logic.
- You may use = or $\leftarrow$ interchangeably.
- **Keywords:** ALGORITHM, BEGIN, END, INPUT, OUTPUT, IF, THEN, ELSE, ENDIF, WHILE, ENDWHILE, FOR, ENDFOR, DO, UNTIL, REPEAT, RETURN, PRINT, READ

Example 1: Sum of Two Numbers a, b

ALGORITHM SumTwo

BEGIN

INPUT a, b

$\text{sum} \leftarrow a + b$

OUTPUT sum

END

Example 2: Calculate area of a circle $A = \pi r^2$

```
ALGORITHM CircleArea
BEGIN
    INPUT r

    A ← pi * r * r

    OUTPUT A
END
```

Example 3: Largest of Three Numbers a, b, c

ALGORITHM FindLargest

BEGIN

 INPUT a, b, c

 largest $\leftarrow a$

 IF $b > \text{largest}$ THEN

 largest $\leftarrow b$

 ENDIF

 IF $c > \text{largest}$ THEN

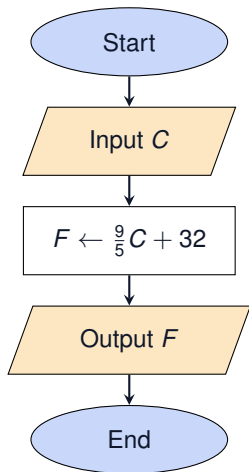
 largest $\leftarrow c$

 ENDIF

 OUTPUT largest

END

Example 4: Celsius to Fahrenheit



ALGORITHM CtoF

BEGIN

INPUT C

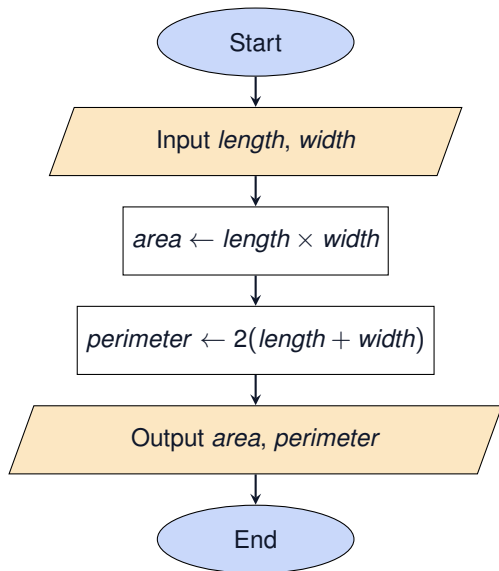
$F \leftarrow (9/5) * C + 32$

OUTPUT F

END

Trace: $C = 100 \Rightarrow F = 212$

Example 5: Area and Perimeter of a Rectangle



ALGORITHM RectStats

BEGIN

INPUT length, width

$\text{area} \leftarrow \text{length} * \text{width}$

$\text{perimeter} \leftarrow 2 * (\text{length} + \text{width})$

OUTPUT area, perimeter

END

Trace with $\text{length} = 5$, $\text{width} = 3$:

- $\text{area} = 5 \times 3 = 15$
- $\text{perimeter} = 2(5 + 3) = 16$

Try it yourself

Design pseudocode **and** a flowchart for an algorithm that:

- Takes as input the price of an item and a quantity purchased.
- Computes the total cost.
- Outputs the total cost.

Hint: identify inputs, the formula (process), and the output — then draw the four/five matching shapes.

- **Pseudocode** and **flowcharts** let us design algorithms independently of any programming language.
- Four core flowchart shapes: **terminator**, **input/output**, **process**, **decision**.
- **Sequential** algorithms execute steps strictly in order.
- Always **trace** your flowchart with sample values before coding it.

Coming up next: adding *decisions* to our algorithms.