Introduction to Programming

Fundamentals of Programming

Lecture 1A

Contents

- Introduction
- Software development
- Waterfall method
- Algorithm
- Pseudocode
- Flowcharts

Introduction

- **Program**: set of instructions that a computer follows to perform a task
  - Commonly referred to as *Software*
- **Programming:** is the process to create instructions to perform certain tasks on a computer system
- **The programming language** gives a programmer the guideline of the structure of the instructions

Programming

- Developing a program is a logical process
- The program code may run in sequence or repeated in any order
- Applications of software:
    - Office productivity
    - Mathematics
    - Graphics
    - Machine control
    - Robotics
    - Games
    - Networking

C programming language

- Adds a layer of abstraction above the machine code
- The abstraction layer allow programmers to focus on high-level software development
- A procedural programming language, that follows a sequential, step-by-step approach to solve problems
- Many modern programming languages (C++, C#, Objective-C) are based on C language

Software development

Developing software

- Software development requires a framework in order to work efficiently

- Following a framework, programmers work as a team to produce predictable results

- There are many models designed for the software development process

Software development process

- Knowledge of a programming language
- Knowledge of software patterns
- Compiler and debugging tools
- Tools to document software designs
- Methodology for working systematically in a team

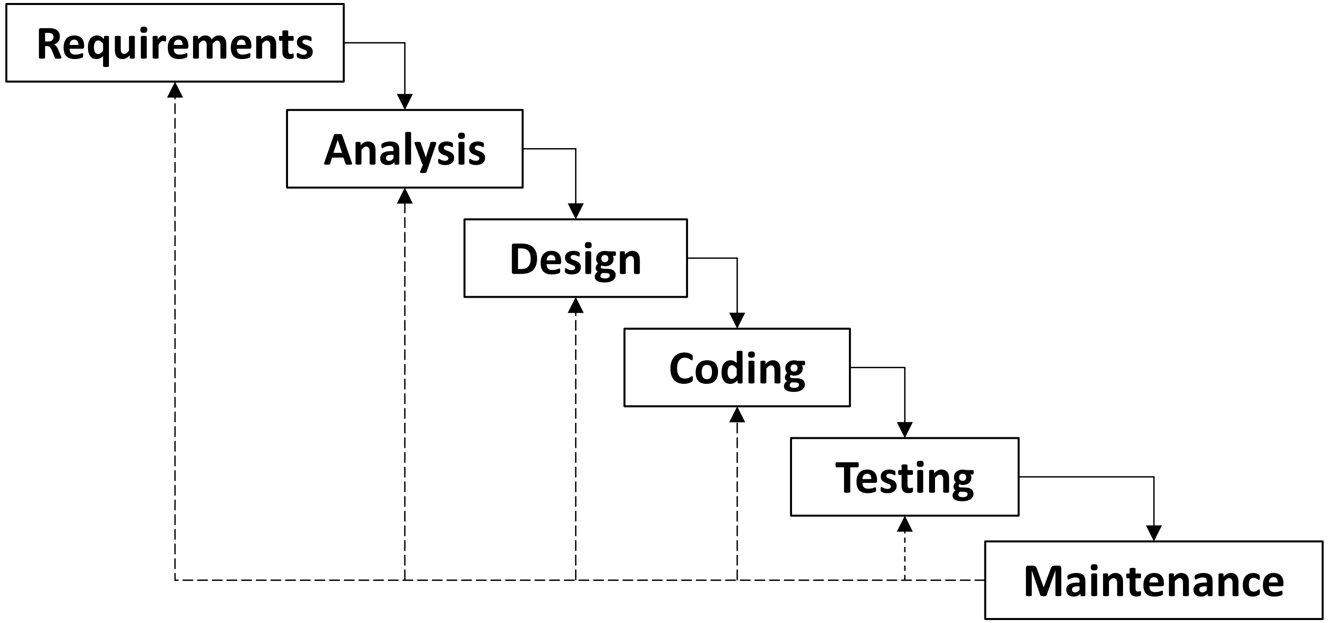
Software development methodology

- Waterfall
- Spiral
- V model
- Agile

Waterfall method

Waterfall method

- A software development methodology
- Organises work into sequential phases
- If any issue is encountered, corrections can be done by returning to any previous phase
- Documentation is performed at every phase



Waterfall method steps (1)

Requirements

- It is used to determine the requirements of the software project
- Project constraints may include software functionality, feasibility, tools, computer system, manpower, cost, scheduling, deployment method, etc.

Analysis

- It is used to analyse the problem to determine the models, rules, and the most suitable algorithm for the software

Waterfall method steps (2)

Design

- It is used to design the overall architecture, components and functions of the software
- Tools such as flowcharts, pseudocode, modelling diagrams, etc. are used to describe the designs

Coding

- It is used to translate the designs into actual code

Waterfall method steps (3)

Testing

- It is used to find any errors or defects in the software by applying test data

Maintenance

- It is used to install, support, and maintain the software using continuous improvements over a period of time

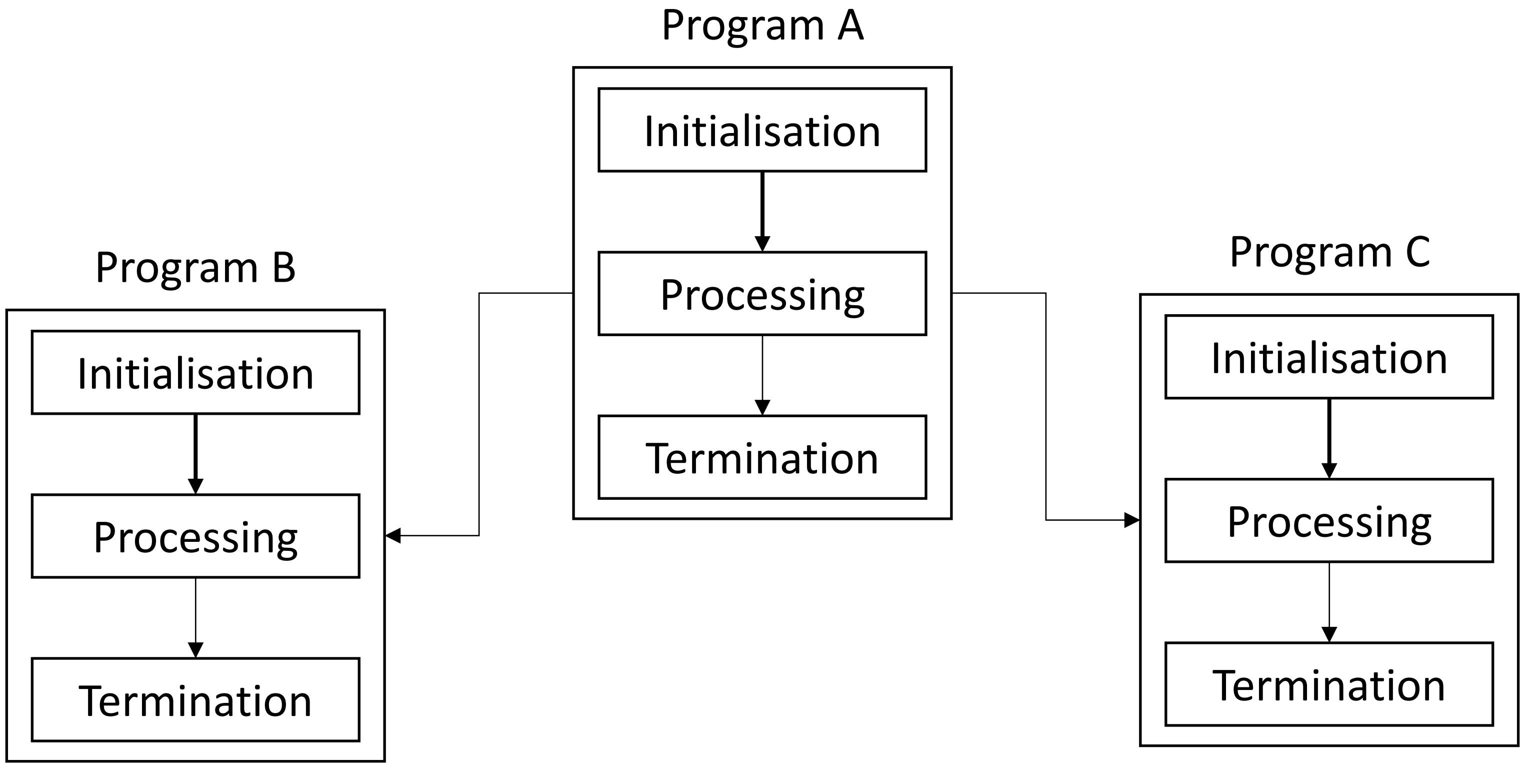
Design approaches

- There are two approaches for design: (1) Top-down; (2) Bottom-up
- Top-down approach
  - Start from an overview of the problem as a system
  - Break down the problem into sub-system parts
  - Solution is formed when individual sub-system parts are resolved
- Bottom-up approach
  - Start with the most basic element of a system for the problem
  - Build up the details of the element to increase functionality
  - Solution is formed when the top-level system is complete

Sections of a program

- Initialisation
    - Prepare variables and initial conditions for the application
- Processing
    - Execute the processes or actions
- Termination
    - Print results, close files, prepare to shut down application

Program structure



Algorithm

Algorithm

- An **algorithm** is a procedural list of instructions that is used to describe how to solve a problem
- How the instructions are written dependant on the specific discipline
  - Mathematics, computer science, engineering and other STEM-related instructions will be technical in nature

Algorithm example

- Convert value in inches to cm:
  - Find out the ratio of inches to cm (1 inch = 2.54 cm)
  - Create an algebraic formula, X inch = (2.54 * X) cm
- Therefore, the algorithm:
  - Read input X
  - Multiply X with 2.54

Control structures in algorithms

- Sequential structure
  - The instructions are ordered and run sequentially without any conditions
- Selection structure
  - Either one of two instructions are to be selected for execution
  - Each instruction represents a Boolean result (true or false)
  - An input condition is tested, and whichever instruction associated with the Boolean result is executed
- Repetition structure
  - The instructions are repeatedly executed until the loop test condition is false

Pseudocode

Pseudocode

- Pseudocode is a textual representation of an algorithm written in plain language
- It is a high-level description of software functions so that it is readable by humans.
- Programming keywords may be used to describe an action or operation
- For instance, IF-ELSEIF, REPEAT, GOTO

Pseudocode example 1

- **Convert Celsius to Fahrenheit**
- Prompt user to input value of temperature C in Celsius
- Apply the formula, $F = 1.8*C + 32$, where F is the temperature in Fahrenheit
- Print the result to the screen

Pseudocode example 2

- **Calculate the value of y = mx + c** for all values of x from 0 to 3
- Prompt user to input values for gradient m and y-intercept c
- Apply the formula, y = mx + c as part of a loop, where the value of x increments in each round
- Print each result of y to the screen

Pseudocode example 3

- **Calculate the average of 10 student marks in five modules**
- Add each module mark from the first student to total marks
- Save the first student’s total marks in the computer
- Repeat until tenth student’s total marks is stored
- Divide each student’s total marks by 5 to get the student’s average

Pseudocode example 4

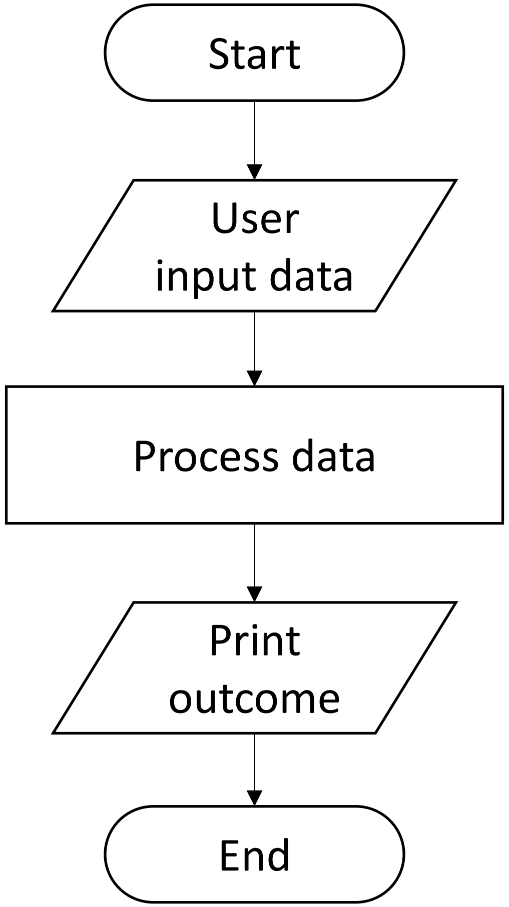
- **Programming a robot arm to make a cup of instant coffee**
- Move arm and grip a spoon
- Move arm to coffee powder container, and scoop out a spoonful
- Move arm to mug and tilt grip to pour coffee powder into mug
- Loosen grip and leave spoon in mug
- Move arm and grip kettle on stove, moving kettle to mug area
- Tilt grip to pour hot water into mug, return kettle back to stove
- Move arm to mug, grip spoon, position arm to stir for 10 seconds
- Loosen grip and move arm back to base

Lengthy and not easy to use sometimes

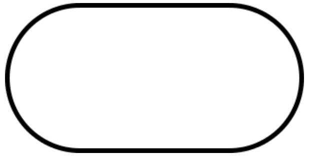
Flowcharts

Flowcharts

- A flowchart is a visual representation of software design
- A graphical object conveys more meaning than pure text
- Flowcharts allow loops and complex problems to be easily understood

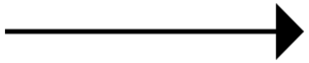


Flowchart symbols (1)



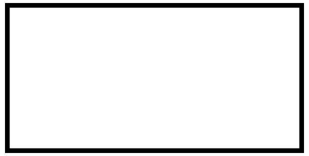
Terminator/Terminal

Indicates the Start or End of the program.



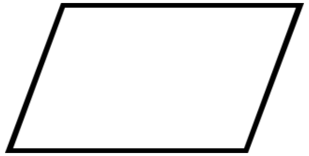
Flowlines/Arrows

Link symbols together to indicate the order of operation.



Process

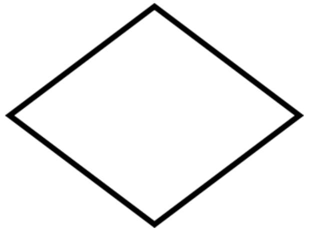
An action or process on data.



Input and Output (IO)

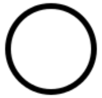
Input data, output data, or display results.

Flowchart symbols (2)



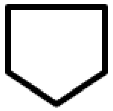
Decision

Indicates a conditional test which either one of two (or more) outcomes will be executed.



On-page connector

Connects a pair of labels (letter or number) from one section of the flowchart to another in the same page.

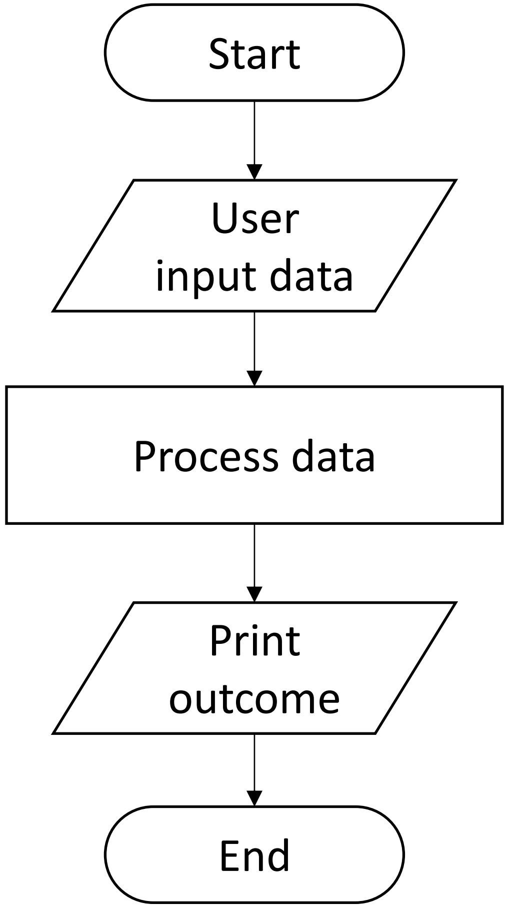


Off-page connector

Connects a pair of labels (letter or number) from one page to another.

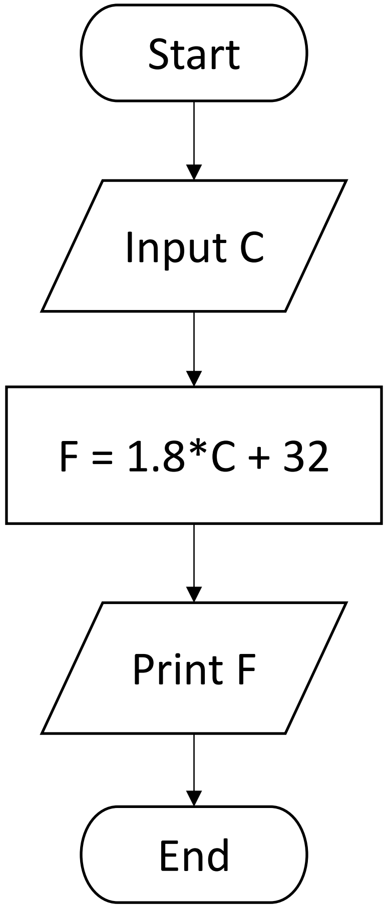
Basic flowchart

- The terminator Start indicates the beginning of the program
- The flowline leads to the process to get data from the user
- The next flowline connects to a process to print the data from the user to the terminal screen
- The terminator End indicates the program has completed



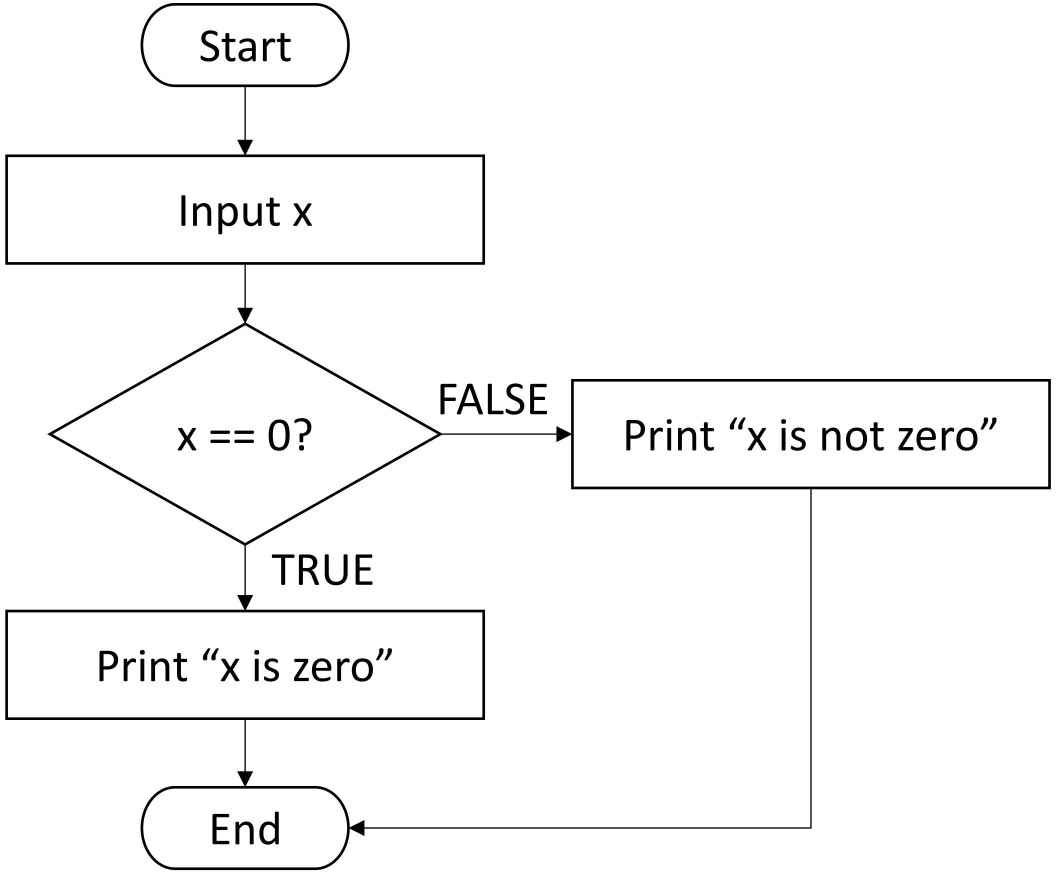
Flowchart example 1

- Convert Celsius to Fahrenheit
- Prompt user to input value of temperature C in Celsius
- Apply the formula, F = 1.8*C + 32, where F is the temperature in Fahrenheit
- Print the result to the terminal screen



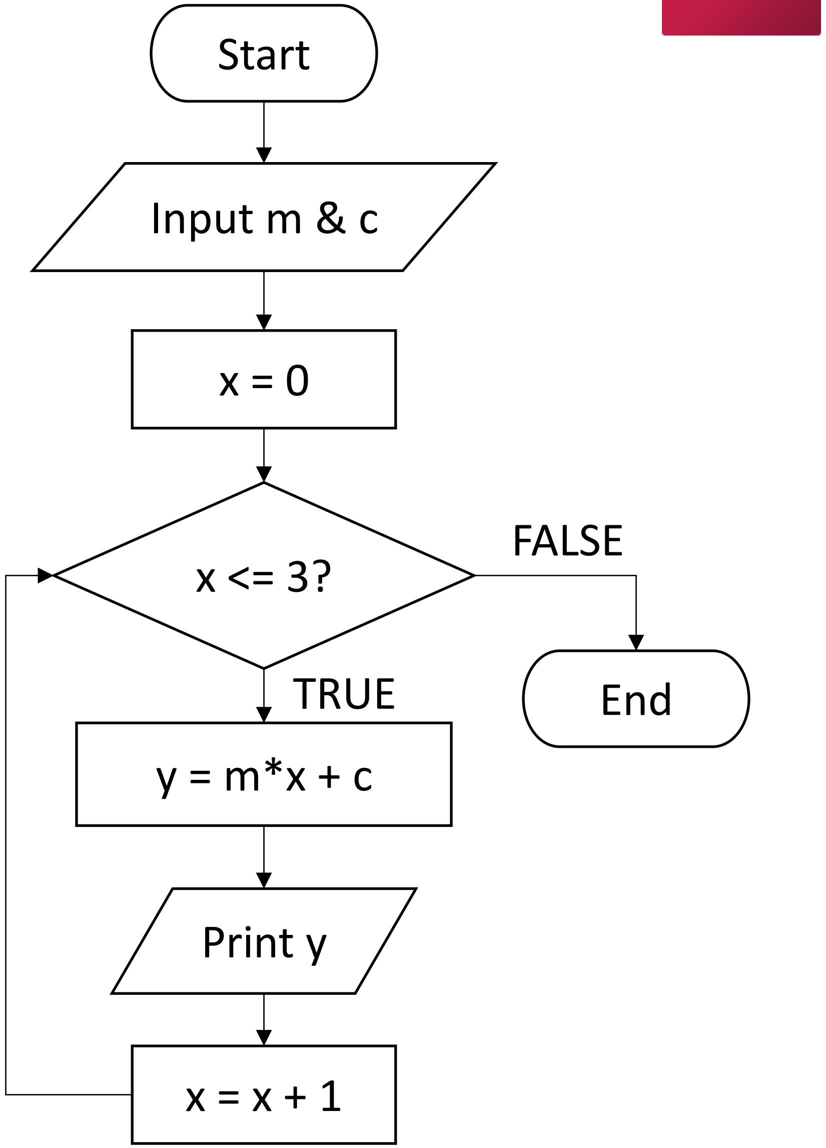
Flowchart example 2

- Determine if x is zero or not
- Prompt user to input x value
- Use conditional test to check if x is zero
- If x is zero, print “x is zero” to terminal screen
- Else print “x is not zero”

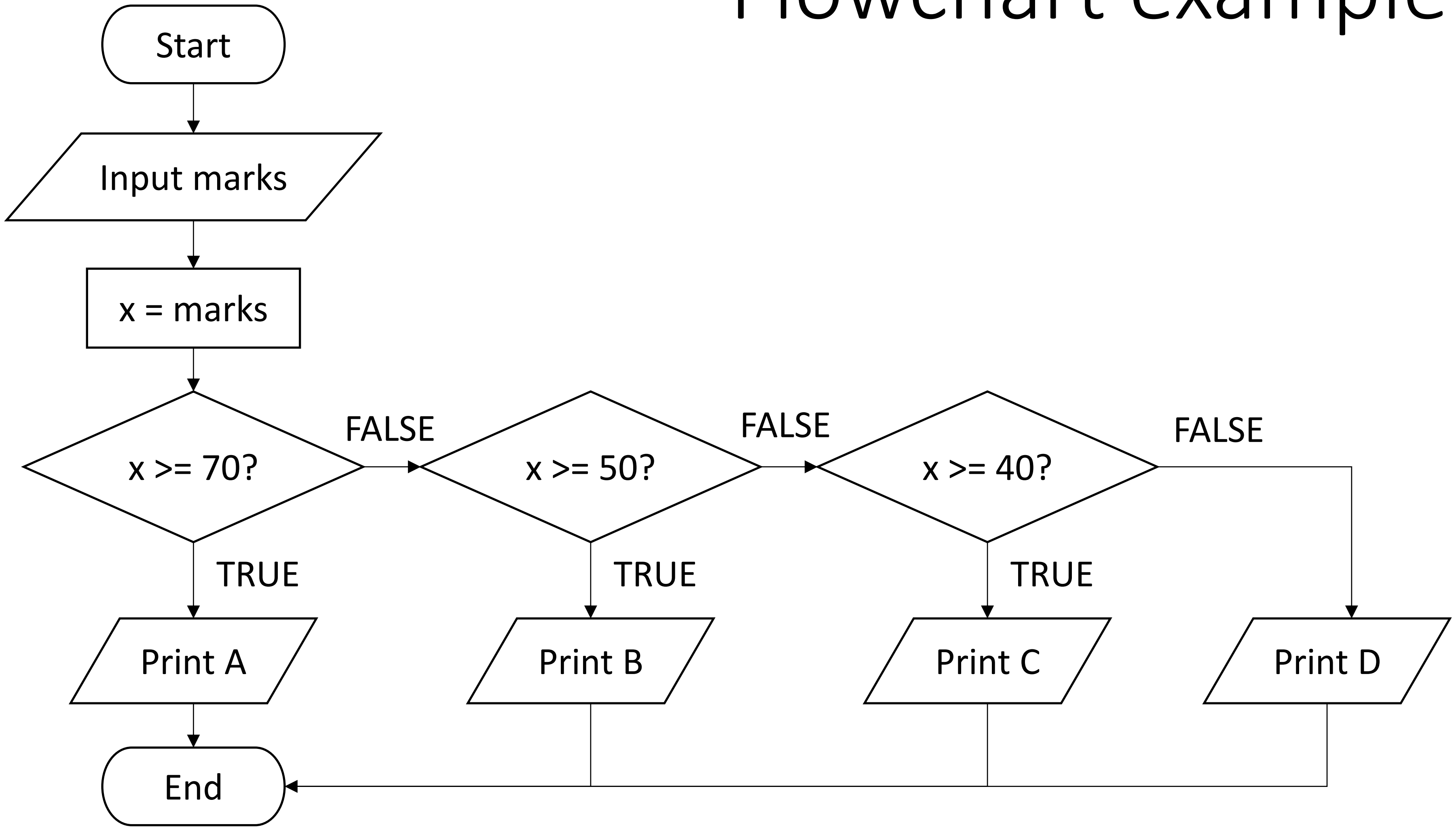


Flowchart example 3

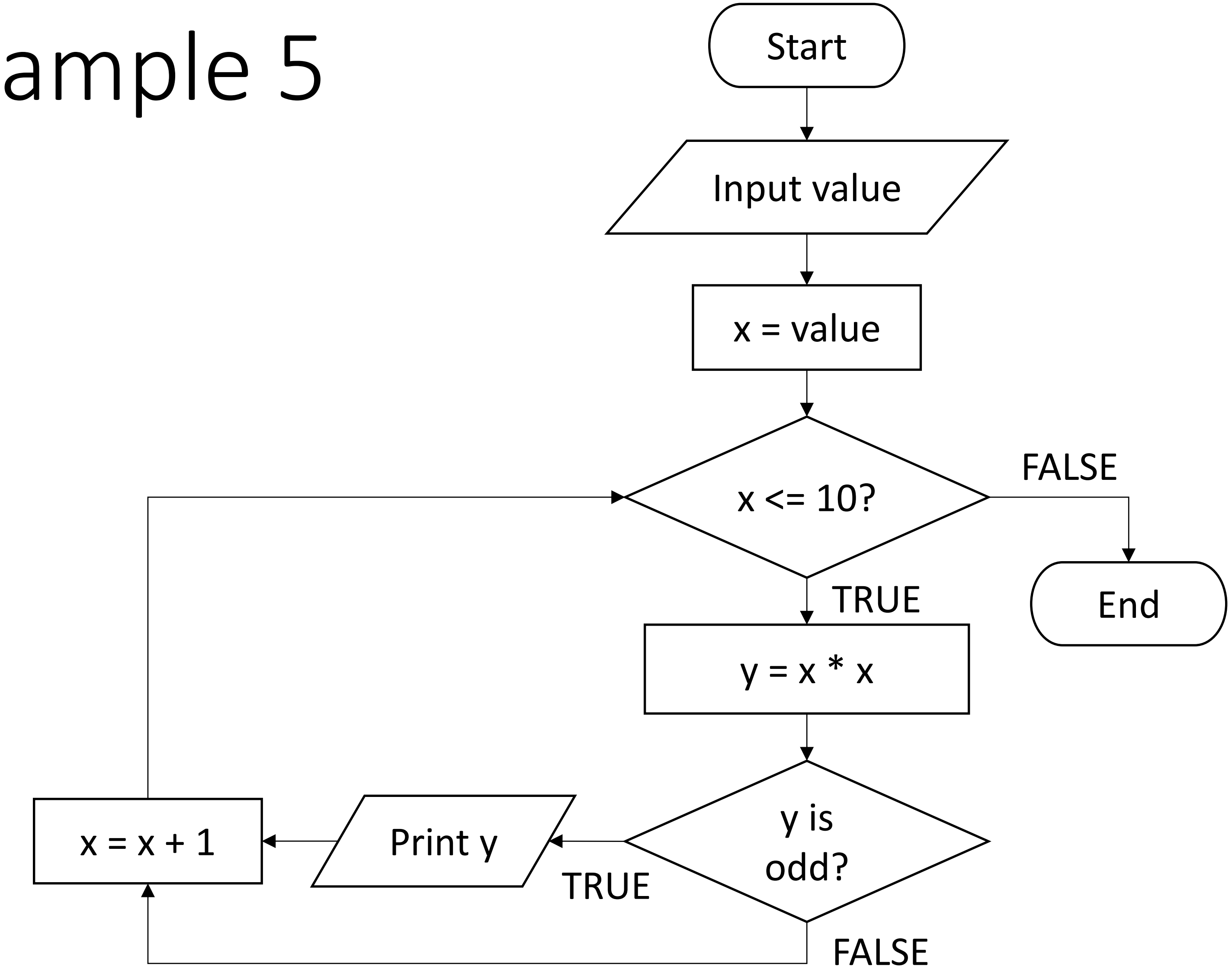
- Calculate the value of y = mx + c for all values of x from 0 to 3
- Prompt user to input values for gradient m and y-intercept c
- Apply the formula, y = mx + c as part of a loop, where the value of x increments in each round
- Print each result of y to the terminal screen



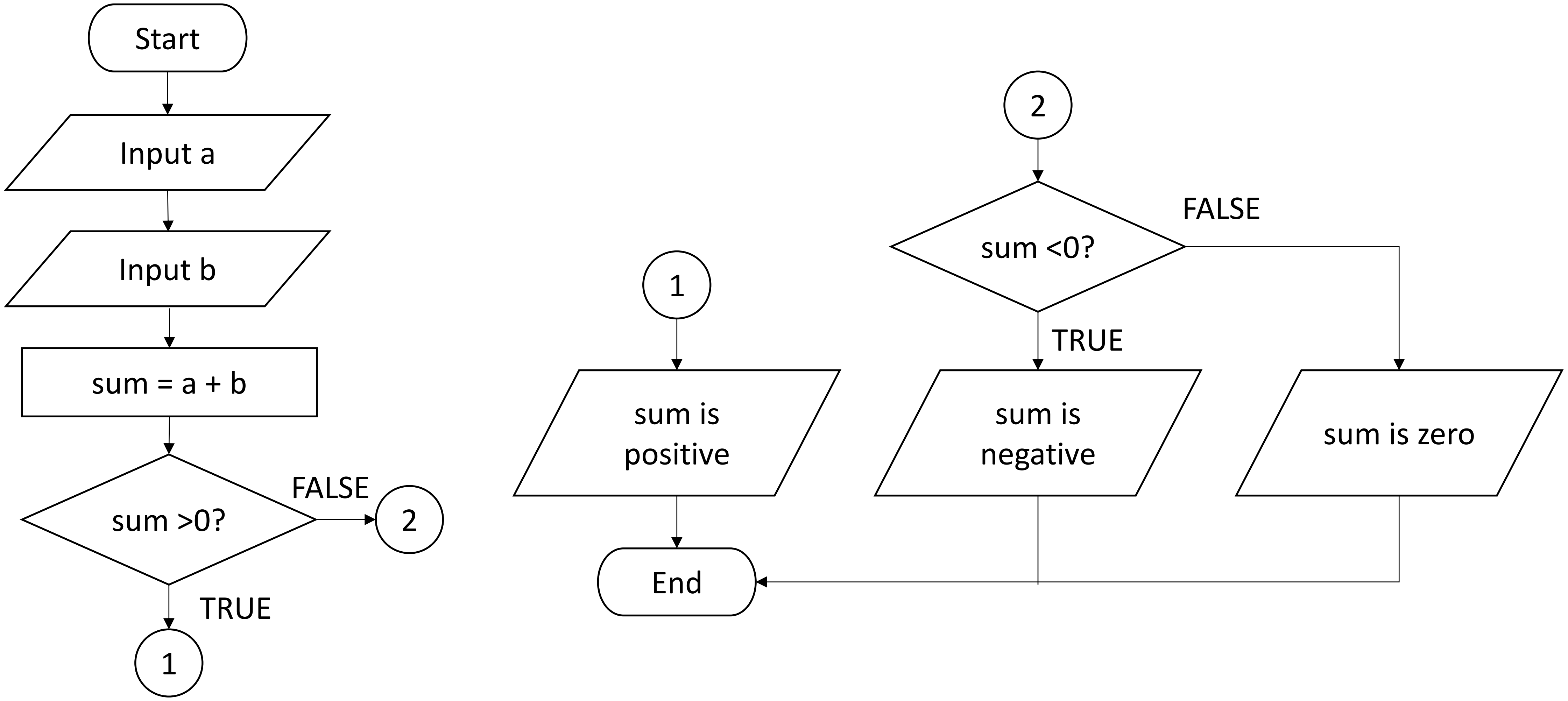
Flowchart example 4



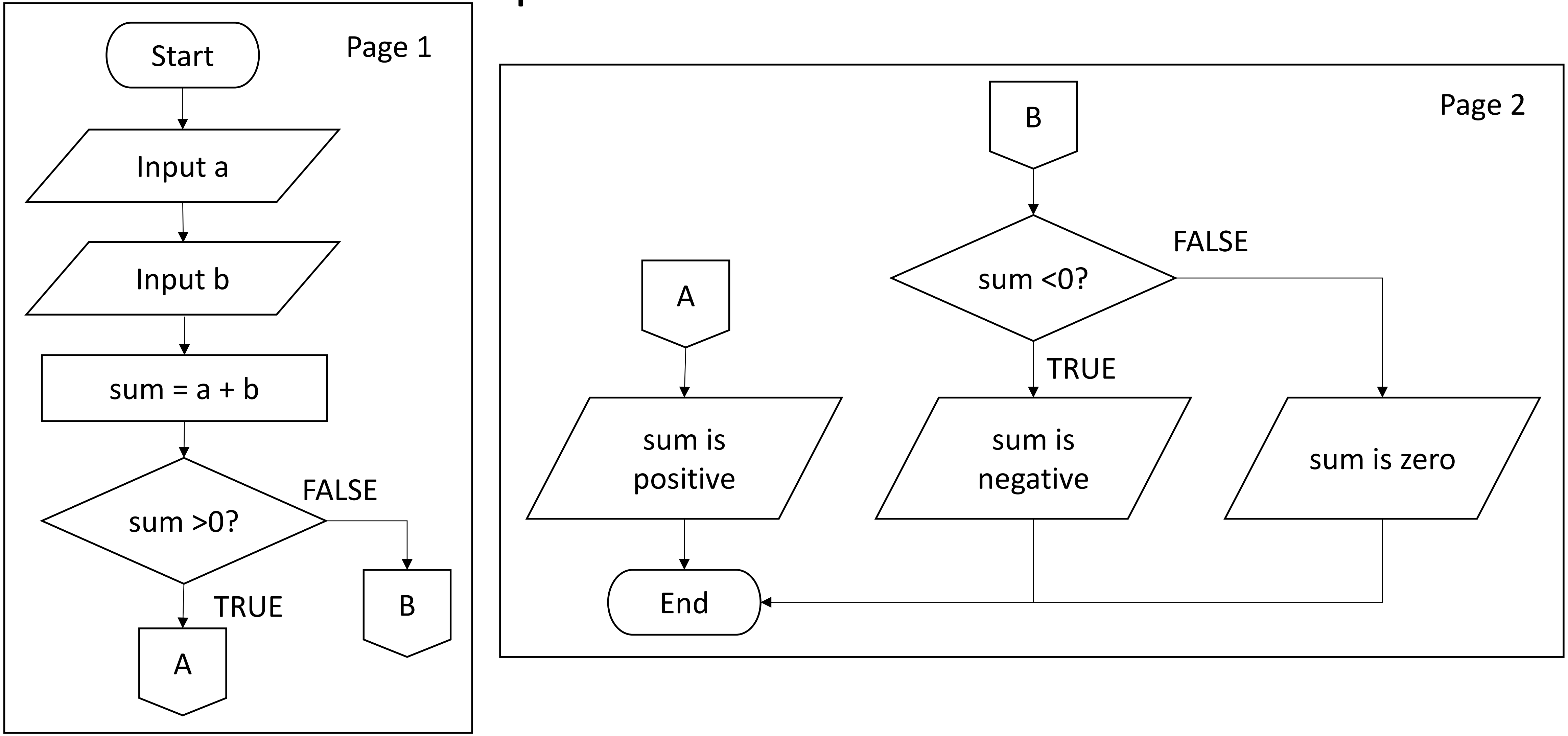
Flowchart example 5



Flowchart example 6



Flowchart example 7



Summary

- Programming is an essential skill for developing modern solutions
- Algorithm is a technical recipe to solve a problem
- Software development requires a systematic method for solving problems and working in a team
- Design tools such as pseudocode and flowcharts are used to describe how the software is to work

Further readings

- C programming language overview @ Wikipedia
  - https://en.wikipedia.org/wiki/C_(programming_language)
- C Language Reference
  - https://en.cppreference.com/w/c
- C Syntax @ Wikipedia
  - https://en.wikipedia.org/wiki/C_syntax