



UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI INFORMATICA

2. Introduction to Programming

Algorithm

An **algorithm** is a finite sequence of instructions performed in order to solve a problem or accomplish a task in a reasonable amount of time

It is important to distinguish an algorithm from a program: the implementation of an algorithm is known as a program

The steps must be logical and specific for machines to carry out (*correctness*) and it must finish in a reasonable amount of time (*efficiency*)



Properties of an Algorithm

- **Input:** it should take zero or more well-defined inputs
- **Output:** it should produce at least one well-defined output
- **Definiteness:** its steps should be clear and unambiguous
- **Finiteness:** no infinite loop should be allowed
- **Effectiveness:** its steps should be doable and effective
- **Independent:** the instructions should be independent of any programming language



Complexity of an Algorithm

The **space complexity** of an algorithm refers to the amount of memory used by the algorithm to store the variables and get the result

The **time complexity** of an algorithm refers to the amount of time that is required by the algorithm to execute and get the result



Representation of an Algorithm

A **pseudocode** is an informal high-level description of the operating principle of an algorithm; it is written in natural language and mathematical notation

A **flowchart** is a diagrammatic representation that illustrates a solution model to a given problem; it is written using different symbols



Pseudocode vs. Flowchart

Pseudocode

```
input M1, M2, M3
GRADE <- (M1 + M2 + M3)/3
if GRADE >= 18 then
    output 'Exam passed!'
else
    output 'Exam failed!'
endif
```

Flowchart

