ĐẠI HỌC

BÁCH KHOA

25 YEARS ANNIVERSARY

SOICT

# ĐẠI HỌC BÁCH KHOA HÀ NỘI
# VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

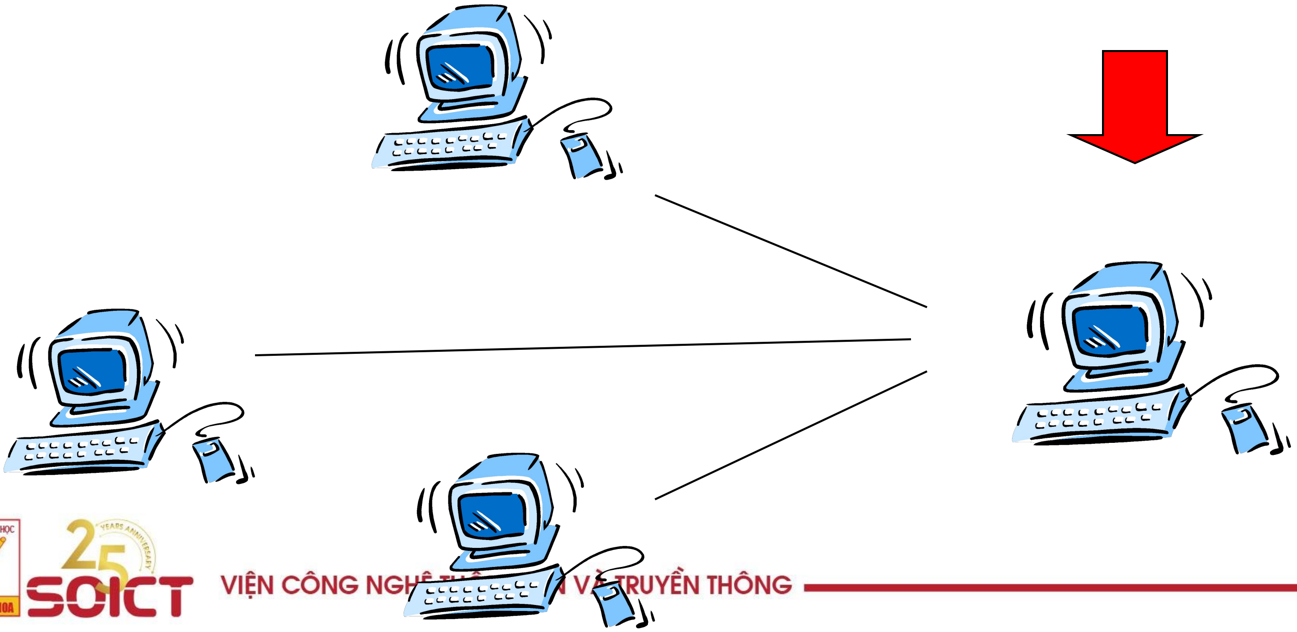
# Programming Introduction

## IT3210 - C Programming Language

## Algorithm

A set of instructions specifying the steps required to accomplish a task

## Program



## What a computer can do?

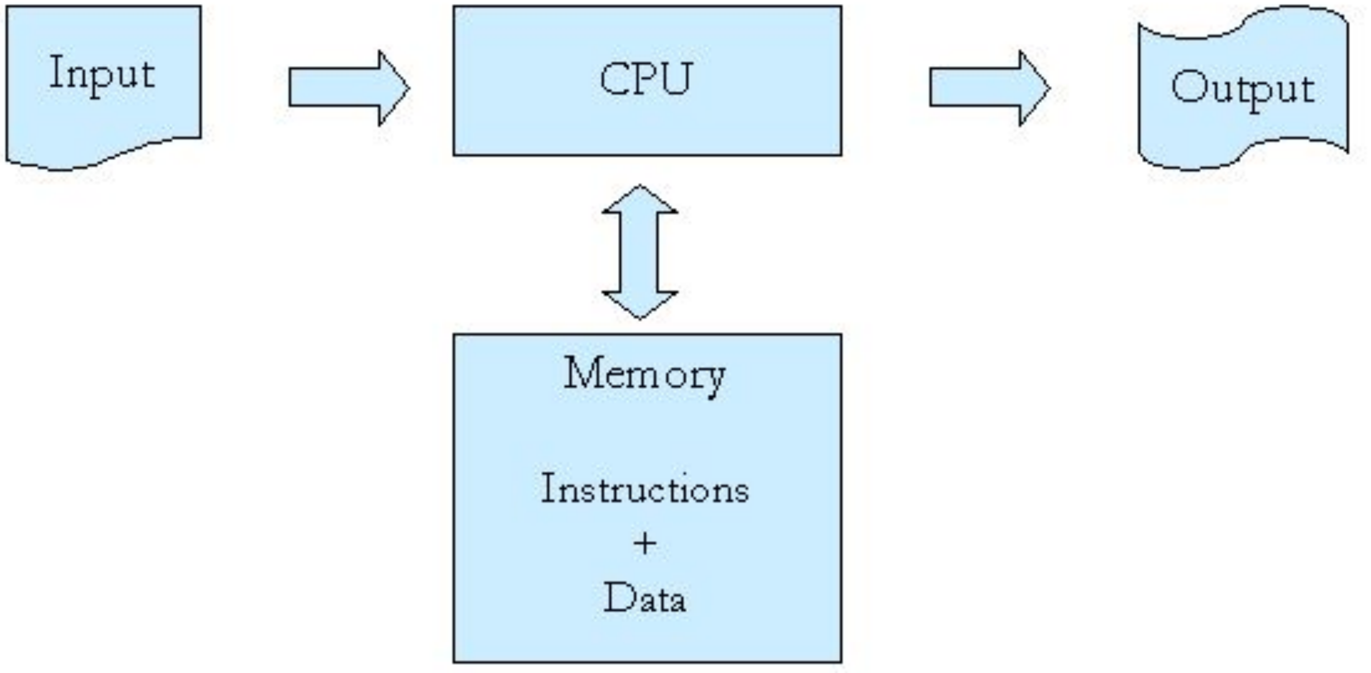
Not much… Computers understand only numbers!

- Store and retrieve numbers (fast and accurate).
- Add, subtract, multiply and divide numbers (also, fast and accurate).
- Compare numbers.
- Follow a list of instructions and jump around in the list

# What else a computer can do?

- More complex calculations can be implemented from a set of simple calculations
- Communicate with peripheral devices to input/output data
  - Input: mouse, keyboards, joystick.
  - Output: graphic cards, printers
- Everything is doable with numbers

# Von Neumann Architecture



Von Neumann Architecture

## What is a computer program?

- A sequence of instructions aims at solving a specific task
- Instruction is carried out one after the other. No instruction is carried out when the previous instruction is not accomplished
- A program is represented by a programming language.

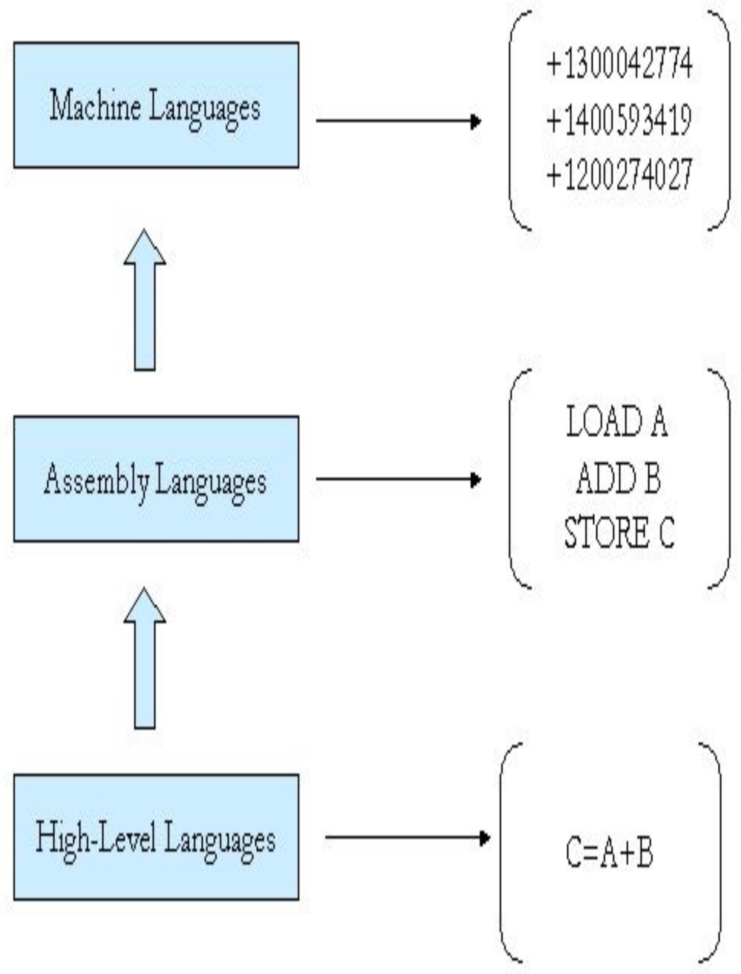
## Programming languages

- **Machine language** is dependent to the computer using machine instructions. Executable programs must be in machine language
- **High level language** is independent to the computer using human algorithm instructions

# Machine language

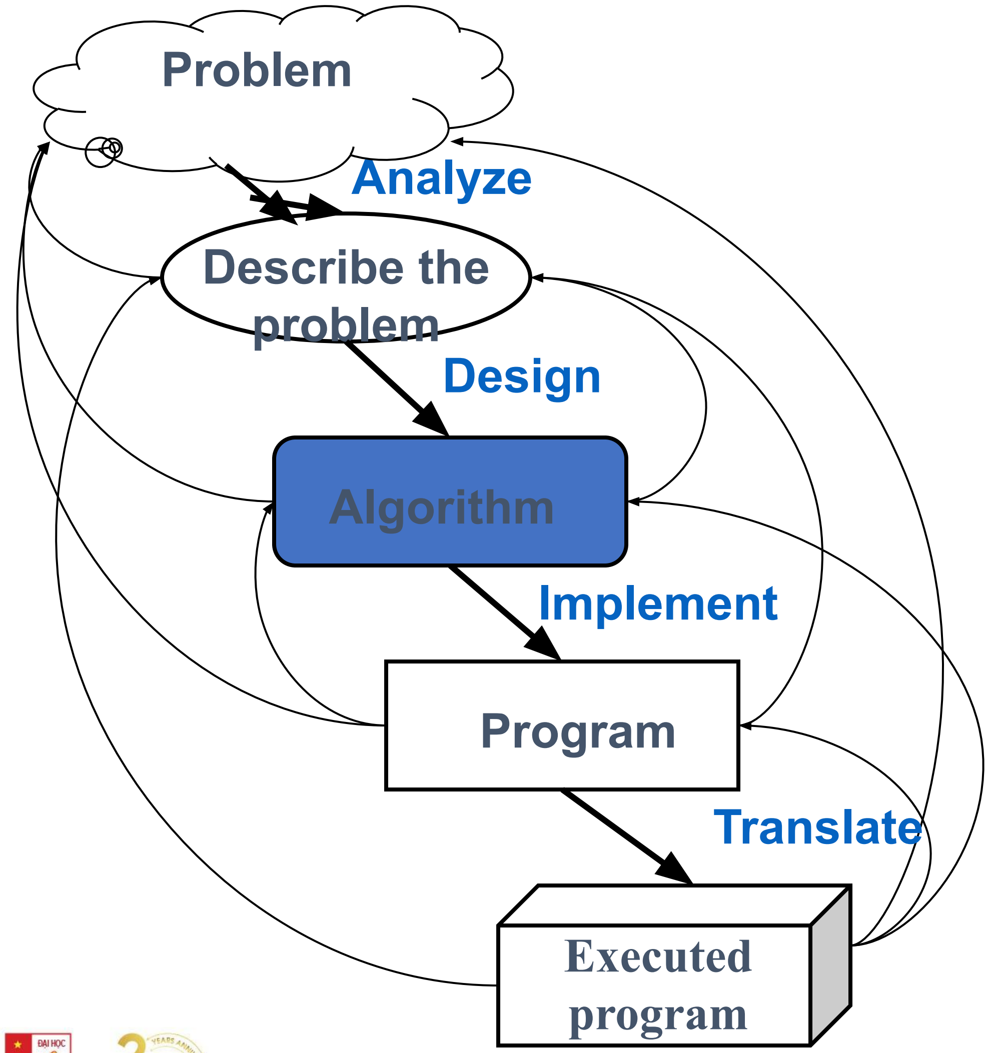
- is a language understandable by computer
- In our view, machine language is only a sequence of 0 and 1.
- There is no common machine language for computer
  - Each micro-processor has its own language
- Human cannot work directly with machine language
- However, computer cannot understand other languages

# High level language



- Assembly – machine language encoded as documents (not convenient)
- Interpretation language (java, perl)
    - A program is translated into machine language during its process
- Translation language (C, pascal)
    - A program is translated into machine language once before process

# The problem-solving process



Rice cooking

Wash rice (0,5kg)
Pour water (1liter) to a casserole
Boil the water
Put rice into the casserole
Turn down heat
Wait 15minutes, take the casserole out

```
washrice(0,5);
pourwater (1);
boilwater();
putintocasserole();
turndownheat();
takecasseroleout();
```

```
01001110101100101010101010010
10101010100110010101010101001
01101001110101010101001001011
10100111101010101111101010100
0110100001101...
```

## Algorithm

- A sequence of instructions specifying the steps required to accomplish some task
- Some examples:
  - Cooking recipe
  - The rules of how to play a game
  - Directions for driving from A to B
  - A car repair manual
  - etc.

# Rice cooking algorithm

- Prepare
  - 0,5 kg rice, 1 liter of water
- Steps:
  - Wash rice (0,5 kg)
  - Pour (1 liter) water to a casserole
  - Boil the water
  - Put rice into the casserole
  - Wait until the water is shallow
  - Turn down heat
  - Wait 15minutes, take the ca
- Result:
  - A casserole that contains rice for 5 people

Input

Processing

Output

## Components of an algorithm

- Variables and values
- Instructions
    - Sequences
    - Selections
    - Iterations
    - Procedures

# Values

- Represent quantities, amounts or measurements
- May be numerical or alphabetical values: eg., a people name, a people size, etc.
- Each value usually has an implicit measuring unit
- Example:
  - Value for kg of rice, value for liter of water in the rice cooking algorithm

# Variables

- Containers or places to store values
- Example

*Variable*

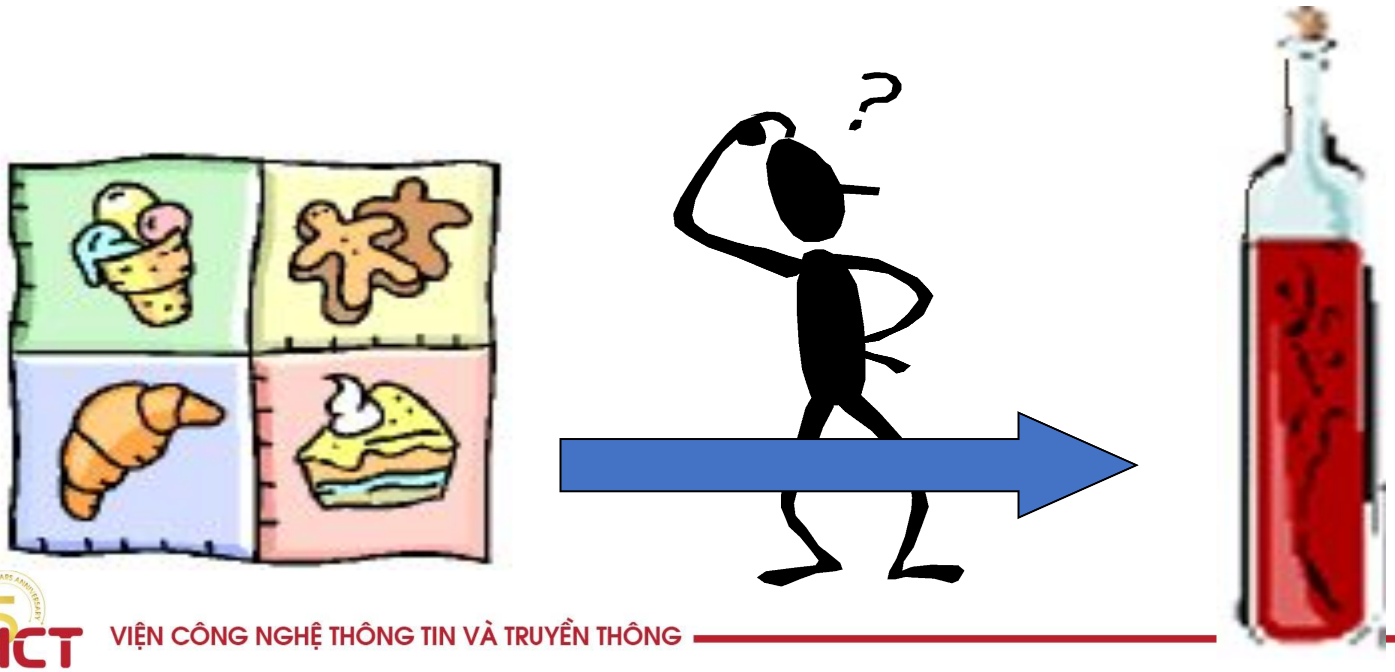
*Values*

This container can be used to store

$\left\{ \begin{array}{l} \text{10 candies} \\ \text{50 g sugar} \\ \text{3 cakes} \\ \text{etc.} \end{array} \right.$

# Type of variables

- Restricted to contain a specific type of value, e.g., only integer number.
- Example : kg (rice) or liter (water)



## Instruction

- Instructions should be:
    - simple
    - unambiguous
    - the system knows the instruction in order to implement it

## Guide about instructions

- Instructions should be simple and unambiguous
- For example:

Wash rice (0,5kg) and then pour water (1 liter) into a casserole and then boil it

- Wash rice (0,5kg).
- Pour water into a casserole (1 liter)
- Boil the water.

✓

## Sequence structure

- is series of instructions to be carried out one after the other
- Example:
  - Wash rice (0,5 kg).
  - Pour water into a casserole (1 liter)
  - Boil the water.
  - Put rice into the casserole
  - Wait until the water is shallow
  - Turn down heat
  - Wait 15 minutes, take the casserole out

## Selection

- Is an instruction that decides which of two possible sequences is executed
- It is based on a condition (true/false)

```
if …
then …
else …
```

## Example about rational number

input an integer

selection condition

Carry out when the condition is true

Carry out when the condition is false

```
input N
if (N ≠ 0)
then
{
   output 1/N
}
else
{
   output “infinitive"
}
```

# Question ?

Do these two algorithms give the same output?

## Algorithm 1

```
input N
if (N ≠ 0)
then
{
        output 1/N
}
else
{
      output “infinitive"
}
```

## Algorithm 2

```
input N
if (N ≠ 0)
then
{
      output 1/N
}

output “infinitive"
```

Algorithm 2 returns both two outputs when N ≠ 0

## Iteration

- Repeat an instruction (or a group of instructions) while (or maybe until) some true or false condition occurs.
- Two kinds of iteration:
    - Test the condition each time <u>before</u> repeating the instruction
    - Test the condition each time <u>after</u> executing the instruction

## Example

Print the odd numbers from 1 to 100

Create variable num with the initial value of 1

The loop is carried out while the condition num <=100 is true

Output num with the current value for each loop and increase num by 2

```
num = 1
while (num <= 100)
do
{
    output num
    num = num + 2
}
```

# Question ?

Do these two algorithms give the same output?

Algorithm 1

```
num = 1
while (num <= 100)
do
{
    output num
    num = num + 2
}
```

Algorithm 2

```
num = 1
while (num <= 100)
do
{
    num = num + 2
    output num
}
```

Algorithm 2 lists all odd numbers from 3 to 101

## Example: Sum of a set of integer values

Find the differences between the two algorithms below:

**Algorithm 1**

```
a = 0
sum = 0
while (a > 0) do
{
        input a
        sum = sum + a
}
output sum
```

**Algorithm 2**

```
sum = 0, a = 0
do
{
        input a
        sum = sum + a
} while (a <>0)
output sum
```

## Procedure

- Is a series of instructions with a name
- You can
    - refer to it (by name)
- Procedure is used in structured programming to divide a program into smaller parts with different names
    - Procedure
    - Function
    - Sub-routine

# Example

```
Procedure RiceCooking
{
    Wash rice (0,5 kg)
    Pour (1 little) water to a
        casserole
    Boil the water
    Put rice into the casserole
    Wait until the water is shallo
    Turn down heat
    Wait 15minutes, take the
        casserole out
}
```

```
Procedure DinerPreparing
{
    RiceCooking
    Boiling vegetabl
    Frying meat
    Setting the table
}
```

Calling procedure

Declaring procedure

## Exercises

1. Write an algorithm to solve the following equation:

   $a*x + b = c.$

2. Write an algorithm to:
   - Input values for 3 variables a,b,c
   - Print out the variable that has the highest value and the variable that has the lowest value. Print out the value for these two variables.

3. Write an algorithm to:
   - Input a value for a variable n >=1
   - Find all numbers <=n that satisfy the following condition
     - Divide 3 remain 2 and divide 5 remain 3

## Summary

- The problem solving process
- Problem → Algorithm → Program
- Programming language
  - High level language vs. machine language
- Components of an algorithm
  - Variables and values
  - Instructions:
    - Sequences, selections, iterations, procedures

ĐẠI HỌC
BÁCH KHOA

25 YEARS ANNIVERSARY
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

## Thank you for your attentions!