



Suez University
Faculty of Petroleum and Mining Engineering
BSE225, Spring Term 16-17



Algorithms

Lecture 5 – Monday March 20, 2017

Outline

- What is an Algorithm?
- Representing Algorithms
- Pseudo-code
- Algorithm Implementation
- Types of Algorithms
- Summary

Outline

- What is an Algorithm?
- Representing Algorithms
- Pseudo-code
- Algorithm Implementation
- Types of Algorithms
- Summary

What is an Algorithm?

Computer science, or computing science, is the study of the theoretical foundations of information and computation and their implementation and application in computer systems.

Source: Wikipedia.org - the free encyclopedia

Computer Science is the study of **algorithms**, including:

- Their formal and mathematical properties
- Their hardware realizations
- Their linguistic realizations
- Their applications

What is an Algorithm?

- An algorithm is a step-by-step specification of a method to solve a problem within a finite amount of time.



Muhammad ibn Musa
al-Khwarizmi (780-850)
Mathematician, astronomer,
astrologer and geographer

- An algorithm is a set of ordered steps for solving a problem, such as a mathematical formula or the instructions in a program.

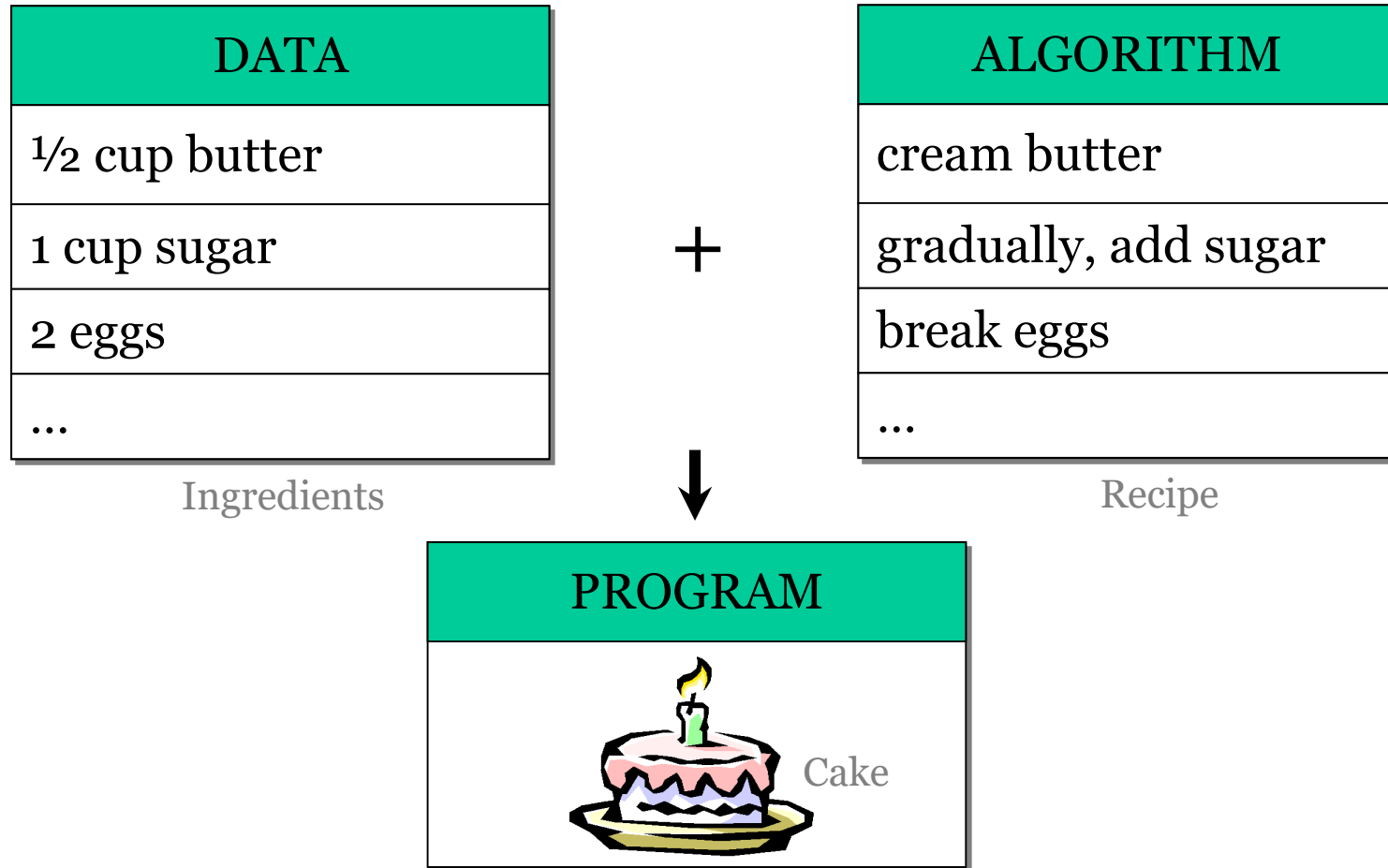
Source: Computer Desktop Encyclopedia.

- An algorithm is a procedure (a finite set of well-defined instructions) for accomplishing some task which, given an initial state, will terminate in a defined end-state.

Source: Wikipedia

What is an Algorithm?

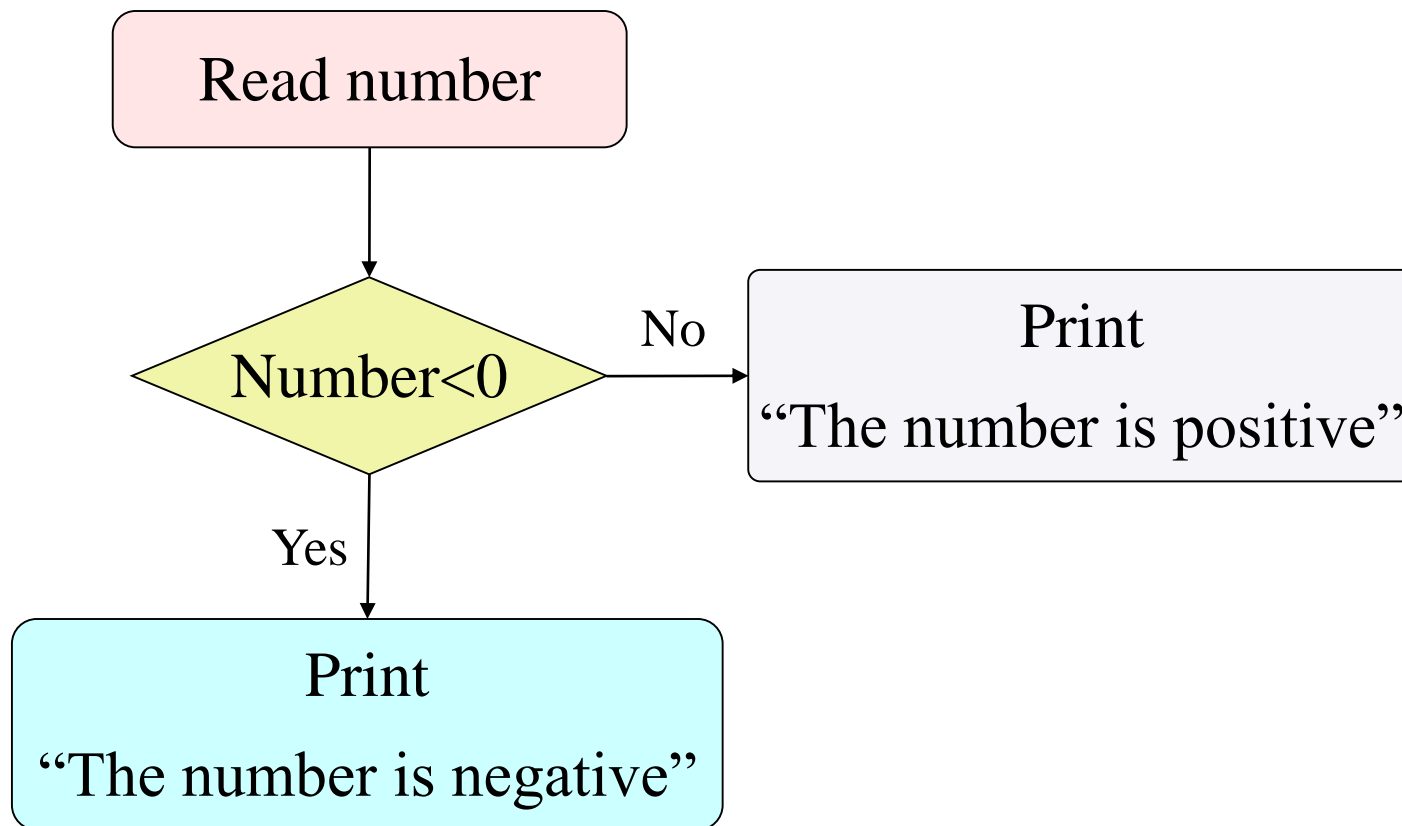
Data + Algorithms = Program



What is an Algorithm?

Problem: Decide whether a number is positive or negative.

Solution:

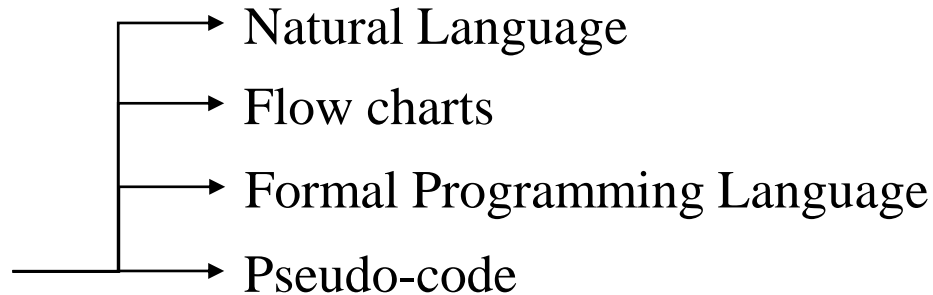


What is an Algorithm?

Problem



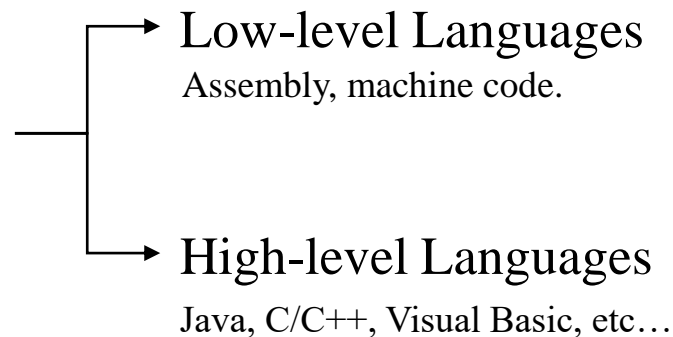
Algorithm Description



Computer Scientist



Algorithm Implementation



Computer Programmer



Solution

Outline

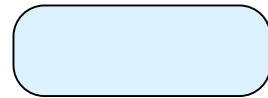
- What is an Algorithm?
- **Representing Algorithms**
- Pseudo-code
- Algorithm Implementation
- Types of Algorithms
- Summary

Representing Algorithms

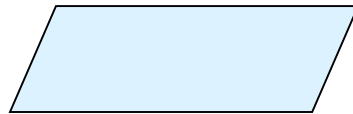
- Flow charts
- Natural Languages
- Formal Programming Language
- Pseudo-code

Representing Algorithms: Flow Charts

A flowchart is a graphical representation of an algorithm.



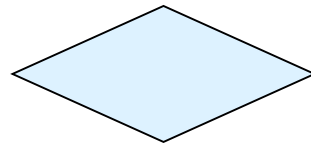
Start and End of an algorithm



Input and Output



Process – Mathematical Operation



Decision – Logical Operation



Connector

Representing Algorithms: Flow Charts

Example-1: Calculate the area of a rectangle.

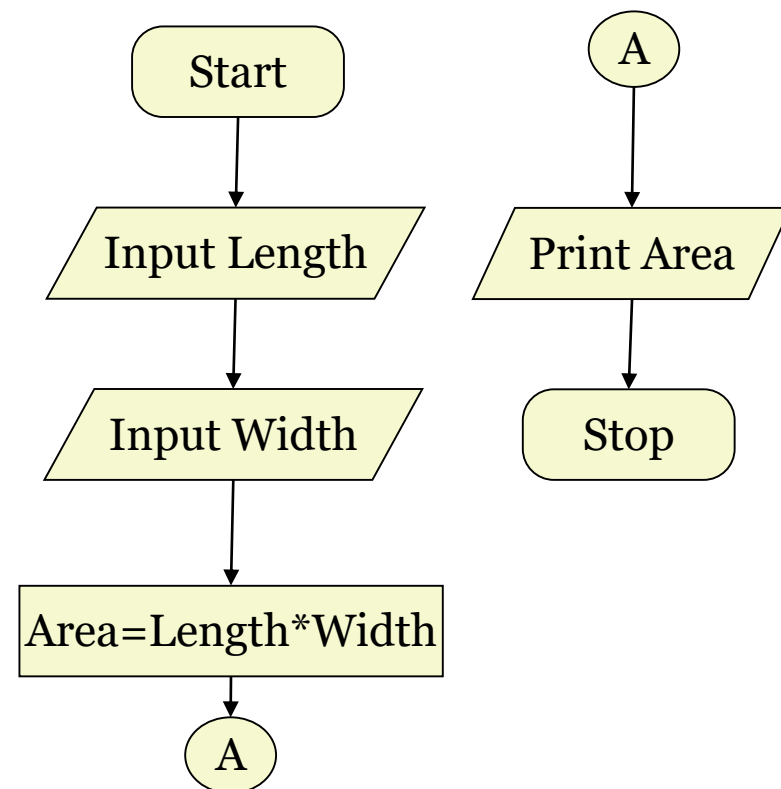
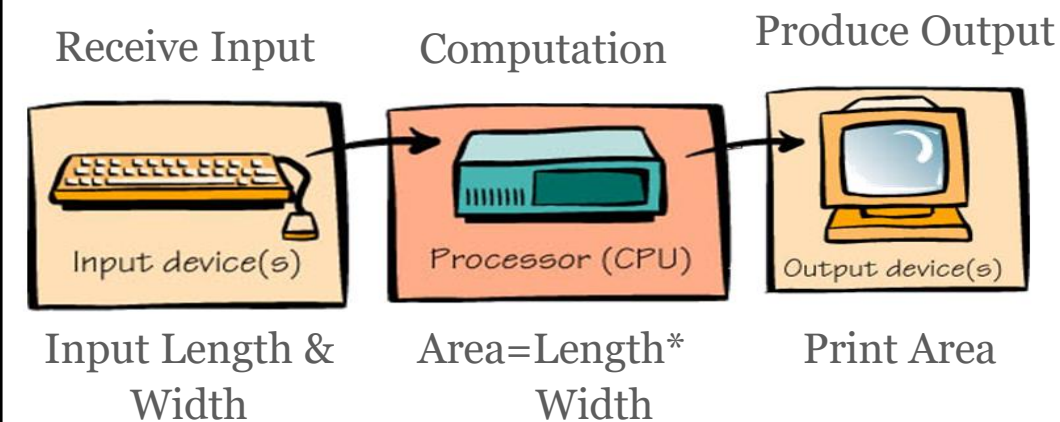
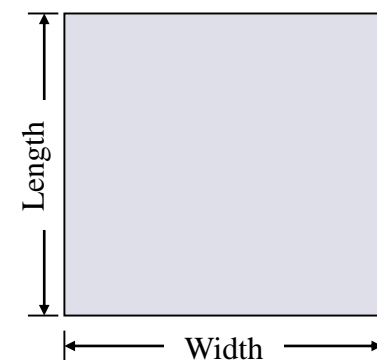
Solution:

Inputs: Length & Width

Outputs: Area

Expression:

$\text{Area} = \text{Length} * \text{Width}$



Representing Algorithms: Flow Charts

Example-2: Given a list of corporate income taxes paid by some Canadian industries in 2005. Calculate the average tax paid.

ID	Industry	M\$
1	finance and insurance industries	11,028
2	Agriculture, fishing, hunting, trapping and support activities	495
3	Oil and gas extraction and support activities	3,749
4	Food and soft drink manufacturing	912
5	Clothing, textile and leather manufacturing	160

Source: Statistics Canada, <http://www.statcan.ca/>

Solution:

Inputs: 5 paid taxes $\Rightarrow T_1, T_2, T_3, T_4, T_5$

Outputs: Average tax

Expression: $\text{Average} = (T_1 + T_2 + T_3 + T_4 + T_5) / 5$

Representing Algorithms: Flow Charts

Example-2:

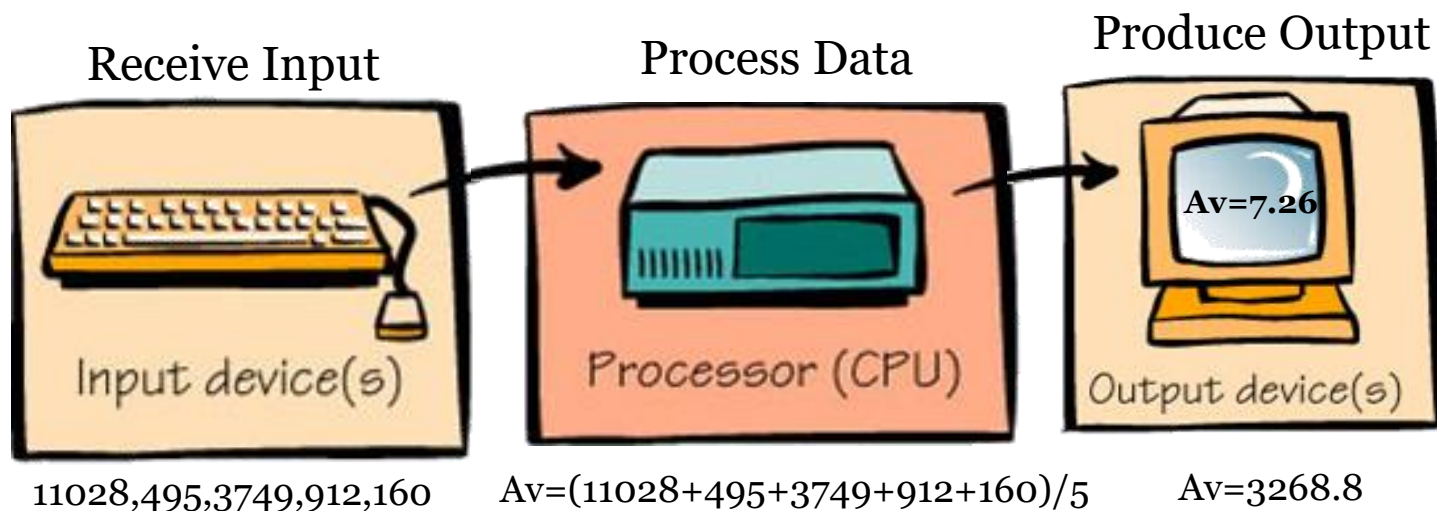
ID	Industry	M\$
1	finance and insurance industries	11,028
2	Agriculture, fishing, hunting, trapping and support activities	495
3	Oil and gas extraction and support activities	3,749
4	Food and soft drink manufacturing	912
5	Clothing, textile and leather manufacturing	160

Inputs: 5 taxes $\Rightarrow T_1, T_2, T_3, T_4, T_5$

Outputs: Average paid tax

Expression:

$$\text{Average} = (T_1 + T_2 + T_3 + T_4 + T_5) / 5$$



Representing Algorithms: Flow Charts

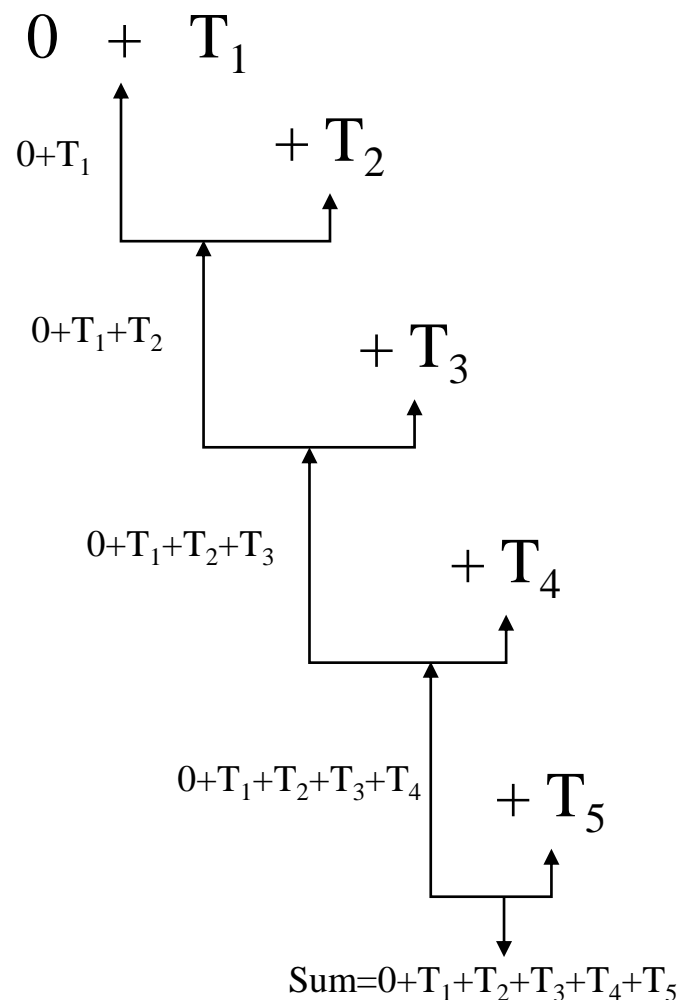
Example-2:

Inputs: 5 taxes $\Rightarrow T_1, T_2, T_3, T_4, T_5$

Expression: $\text{Sum} = T_1 + T_2 + T_3 + T_4 + T_5$,

$$\text{Average} = \text{Sum} / 5$$

Computer is just a **fast stupid machine** so to calculate the average, this machine must be provided by both data and algorithm. A computer with a single processor can perform one operation at a certain time. To calculate the sum of 5 taxes, computer will carry out five additions:

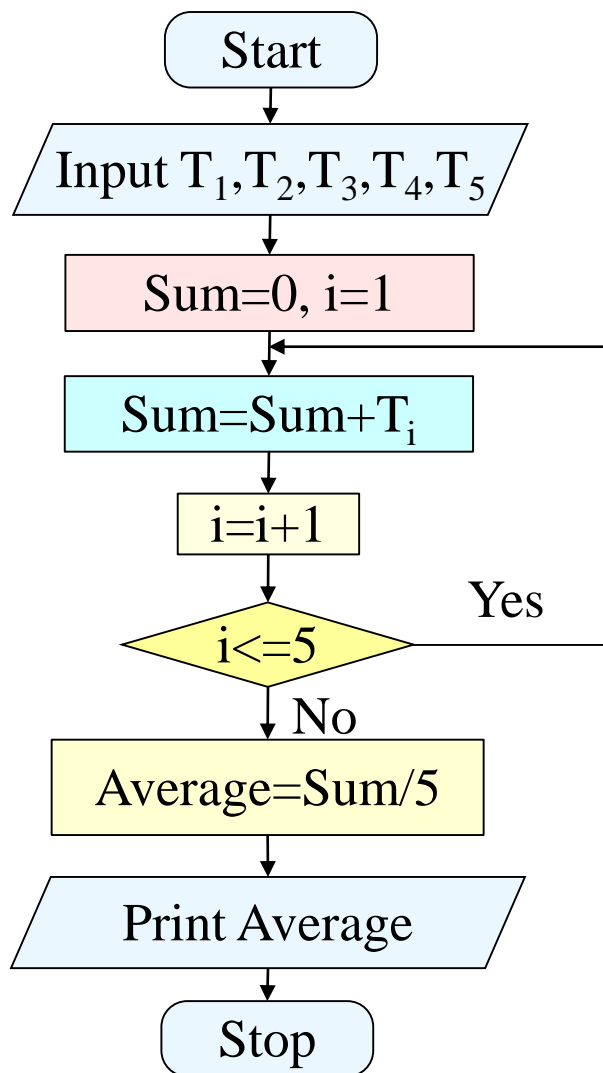


Representing Algorithms: Flow Charts

Example-2:

Iterations	Old Sum	New Sum
1 st Addition i=1	0	$0+T_1$
2 nd Addition i=2	$0+T_1$	$0+T_1+T_2$
3 rd Addition i=3	$0+T_1+T_2$	$0+T_1+T_2+T_3$
4 th Addition i=4	$0+T_1+T_2+T_3$	$0+T_1+T_2+T_3+T_4$
5 th Addition i=5	$0+T_1+T_2+T_3+T_4$	$0+T_1+T_2+T_3+T_4+T_5$

Representing Algorithms: Flow Charts



Iterations	Old Sum	New Sum
1 st Addition i=1	0	$0+T_1$
2 nd Addition i=2	$0+T_1$	$0+T_1+T_2$
3 rd Addition i=3	$0+T_1+T_2$	$0+T_1+T_2+T_3$
4 th Addition i=4	$0+T_1+T_2+T_3$	$0+T_1+T_2+T_3+T_4$
5 th Addition i=5	$0+T_1+T_2+T_3+T_4$	$0+T_1+T_2+T_3+T_4+T_5$

Representing Algorithms: Natural Languages

Get the 5 paid taxes. Initially, set the value of the sum to 0 and the value of the counter i to 1. When these initializations have been completed, begin looping until the value of the variable i becomes greater than 5. First, add T_i to sum. Then add 1 to i and begin the loop all over again. Divide the sum by 5 to compute the average price.

Disadvantages

- too verbose
- unstructured
- too rich in interpretation and meaning (ambiguous)
- imprecise.

Representing Algorithms: Programming Language

```
import java.text.DecimalFormat;
import java.util.Scanner;

public class Average
{
    public static void main (String[] args)
    {
        int sum = 0, Tax, count = 0;
        double average;

        Scanner scan = new Scanner (System.in);

        System.out.print ("Enter paid tax (0 to quit): ");
        Tax = scan.nextInt();

        while (Tax != 0) // 0 to terminate loop
        {
            count++;
            sum += Tax;

            System.out.print ("Enter paid tax (0 to quit): ");
            Tax = scan.nextInt();
        }

        System.out.println ();

        if (count == 0)
            System.out.println ("No values were entered.");
        else
        {
            average = (double)sum / count;

            DecimalFormat fmt = new DecimalFormat ("0.###");
            System.out.println ("The average paid tax is " + fmt.format(average) + "$");
        }
    }
}
```

Disadvantages

- Too many implementation details to worry about such as language syntax, grammar, punctuation, etc.
- Too rigid syntax.

Representing Algorithms: Pseudo-code

- Pseudo is a prefix of Greek origin. It means “false” or fake code.
- Not actually executed on computers.
- Allows us to think out a program before writing the code for it.

BEGIN

get values for paid taxes, T_1, T_2, T_3, T_4, T_5

set Sum to 0

set i to 1

while ($i \leq 5$) {

 set Sum to $\text{Sum} + T_i$

 set i to $i + 1$

}

set Average to $\text{Sum} / 5$

print Average

END

Representing Algorithms: Pseudo-code

- English like constructs (or other natural language) but
- Modeled to look like statements in typical programming languages.

BEGIN

get values for paid taxes, T_1, T_2, T_3, T_4, T_5

set Sum to 0

set i to 1

while ($i \leq 5$) {

 set Sum to $\text{Sum} + T_i$

 set i to $i + 1$

}

set Average to $\text{Sum} / 5$

print Average

END

Outline

- What is an Algorithm?
- Representing Algorithms
- **Pseudo-code**
- Algorithm Implementation
- Types of Algorithms
- Summary

Pseudo-code

Pseudo-code (pseudo is a prefix of Greek origin means “**false**” or fake) is a description of a computer programming algorithm that uses the structural conventions of programming languages, but omits language-specific syntax.

It can also refer to a **high level “language”** whose aim is to generalize the logic and program flow of a computer program.

Pseudo-code

- **Input**

get values for paid taxes, T_1, T_2, T_3, T_4, T_5

or

get T_1, T_2, T_3, T_4, T_5

- **Output**

print Average

Pseudo-code

- **Variables**

A variable is a named storage that holds the data assigned to it until a new value is assigned or the program is finished

Examples:

set the value of **Tax to 3.5** or **set **Tax** to 3.5**

The variable **Tax** holds the value 3.5 after its execution.

set the value of **Tax to **Tax+1****

Same as: add 1 to the value of **Tax** or increment **Tax** (**Tax** is now 4.5)

Pseudo-code

• Conditional Statements

if statement:

```
if <condition> then
operations for the then-part
else
operations for the else-part
endif
```

```
if (my_income >= 10,000) then
    set Tax to 0.10
else
    set Tax to 0.05
endif
```

Conditional operation is a control operation that allow us to alter the normal sequential flow of control in an algorithm. Conditional statements are the “question-asking” operations of an algorithm.

Pseudo-code

• Conditional Statements

Nested if statement:

```
if <first condition> then
    // do first thing
else if <second condition> then
    // do second thing
else
    // do something else
endif
endif
```

```
if (my_income > 100,000) then
    set Tax to 0.17
else if (my_income =100,000) then
    set Tax to 0.12
else set Tax to 0.08
endif
endif
```

Pseudo-code

- **Looping (Iterations)**

while statement:

```
while <condition>
{
    // do some operations
} // end while
```

```
while (years <= 5)
{
    set balance to initial_balance(1+0.05*)years
    set years to years+1
}
```

This means repeat the operations as long as the condition is true

Pseudo-code

- **Examples**

Exmple-1: Write an algorithm in pseudo-code to determine the flying time between two cities given the distance between them and the average speed of the airplane.

$$\mathbf{Time=Distance/Speed}$$

Exmple-2: Write an algorithm in pseudo-code to convert the temperature from Celsius to Fahrenheit.

$$\mathbf{T_f=(9/5)*T_c+32}$$

Pseudo-code

- **Examples**

Exmple-3: Write an algorithm in pseudo-code that gets the radius of a circle as input and calculates the area and circumference.

$$\text{Area} = 3.14 * r^2$$

$$\text{Circumference} = 2 * 3.14 * r$$

Exmple-4: If you invested 10000 pounds with 5% compounded annually. Write an algorithm in pseudo-code that prints out your balance after one year.

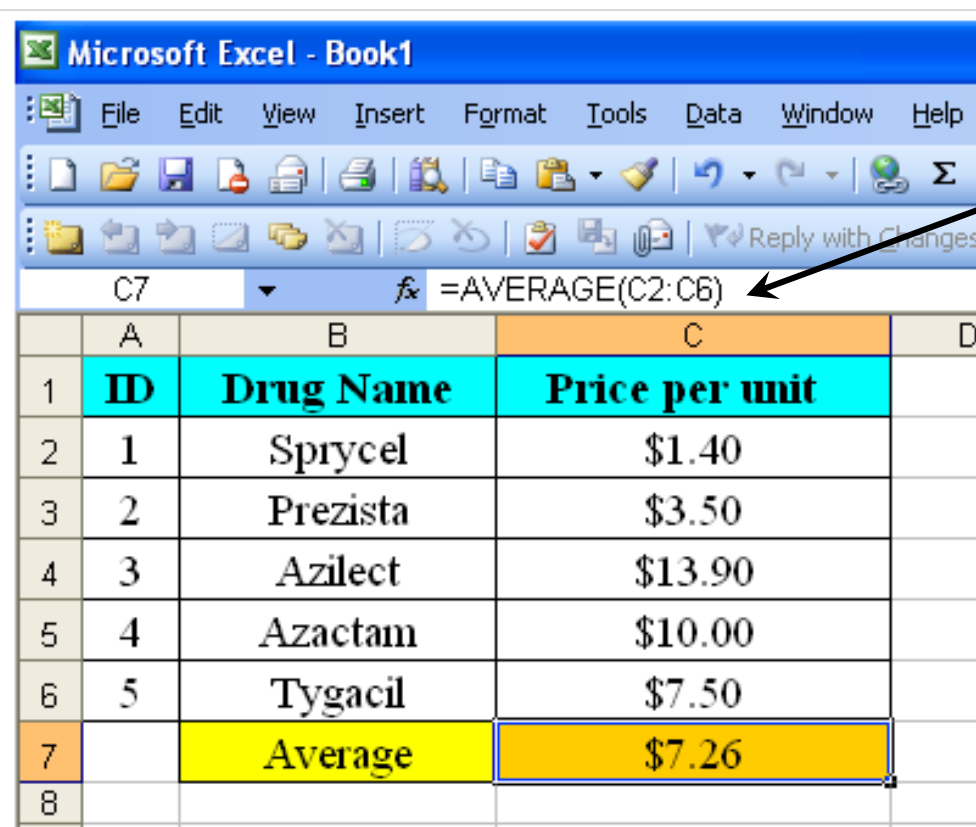
Exmple-5: Write an algorithm in pseudo-code that calculates the net salary of an employee after deducting the 8% of taxes from a gross salary of 2000 L.E.

Outline

- What is an Algorithm?
- Representing Algorithms
- Pseudo-code
- **Algorithm Implementation**
- Types of Algorithms
- Summary

Algorithm Implementation

Problem: Given a list of medical drugs and their prices. Calculate the average price.



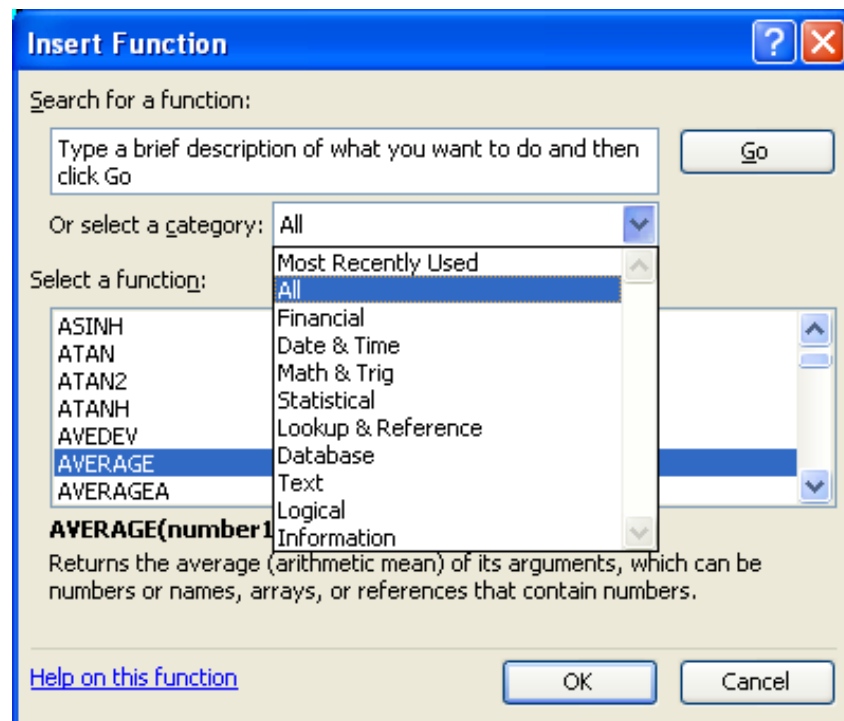
The screenshot shows a Microsoft Excel spreadsheet titled "Microsoft Excel - Book1". The spreadsheet has a menu bar (File, Edit, View, Insert, Format, Tools, Data, Window, Help) and a toolbar. The active cell is C7, and the formula bar shows the formula `=AVERAGE(C2:C6)`. The spreadsheet data is as follows:

	A	B	C	D
1	ID	Drug Name	Price per unit	
2	1	Sprycel	\$1.40	
3	2	Prezista	\$3.50	
4	3	Azilect	\$13.90	
5	4	Azactam	\$10.00	
6	5	Tygacil	\$7.50	
7		Average	\$7.26	
8				

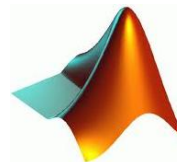
MS Excel has a built-in function called **AVERAGE**, which can be used to calculate the average price in very abstracted way.

Algorithm Implementation

Problem: Sometimes we need to implement an algorithm with mathematical and/or logical operations not supported by Excel or other software packages.



Solution: start to create your own program using a high-level programming language.



MATLAB



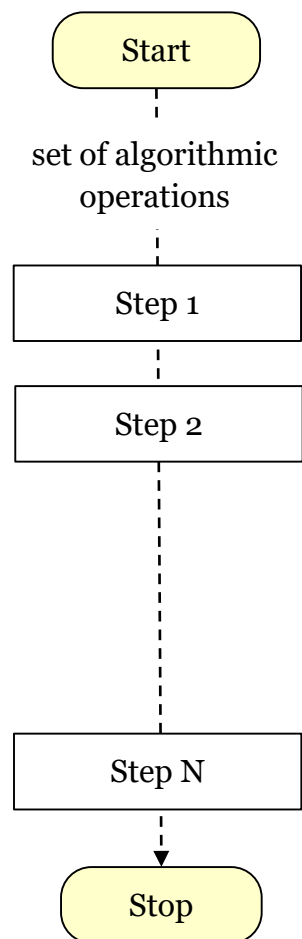
GNU Octave

Outline

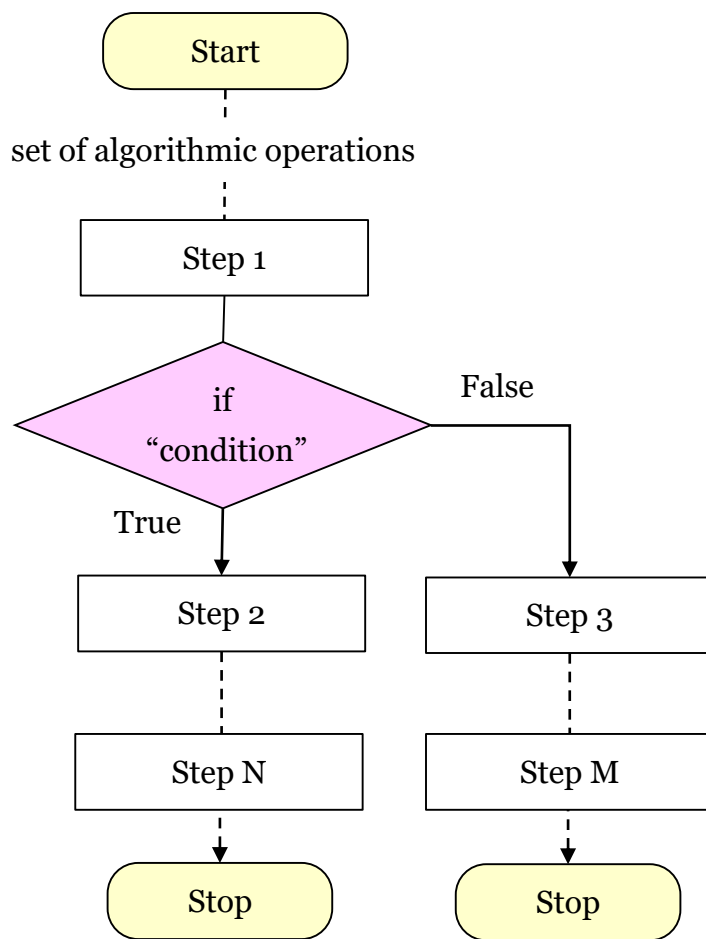
- What is an Algorithm?
- Representing Algorithms
- Pseudo-code
- Algorithm Implementation
- **Types of Algorithms**
- Summary

Types of Algorithms

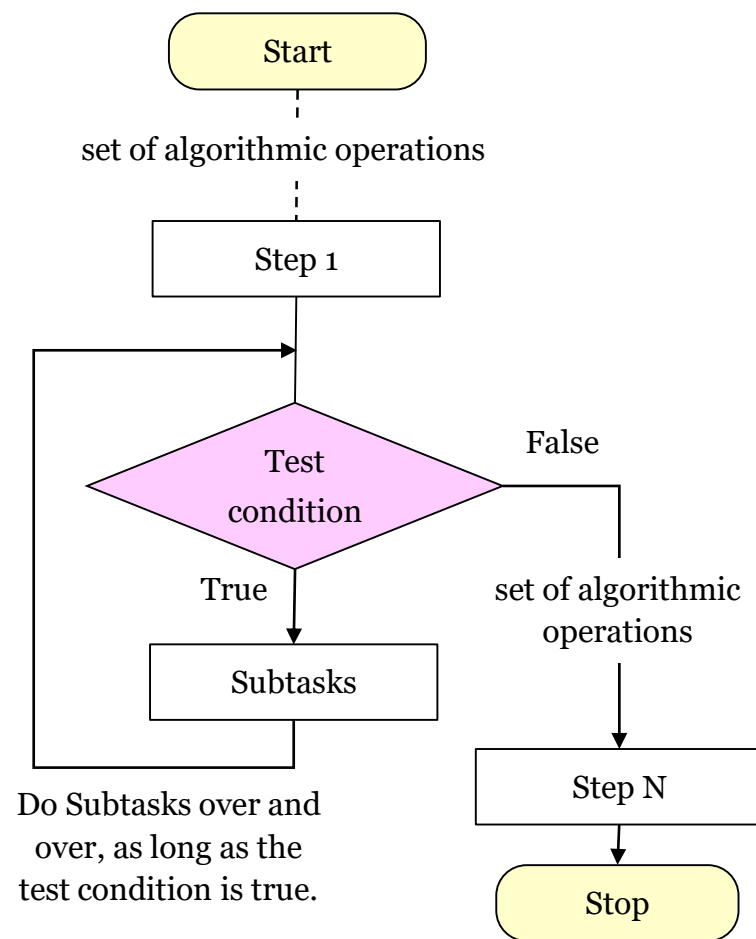
Sequential Algorithm



Conditional Algorithm



Iterative Algorithm



Types of Algorithms

Example: Convert temperature from Celsius to Fahrenheit using the following formula in the following cases:

$$T_f = 9/5 * T_c + 32$$

- a. Convert only one value from Celsius to Fahrenheit,
- b. Convert the Celsius degree to Fahrenheit if and only if the Celsius degree is less than certain value, let's say (90°C)
- c. Convert range of Celsius degrees from (1°C) to (100°C)

Types of Algorithms

Example: Convert temperature from Celsius to Fahrenheit using the following formula in the following cases:

$$T_f = 9/5 * T_c + 32$$

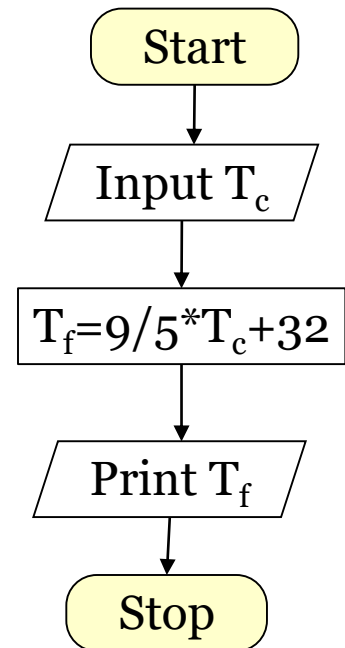
a. Convert only one value from Celsius to Fahrenheit.

Input: T_c

Output: T_f

Expression: $T_f = 9/5 * T_c + 32$

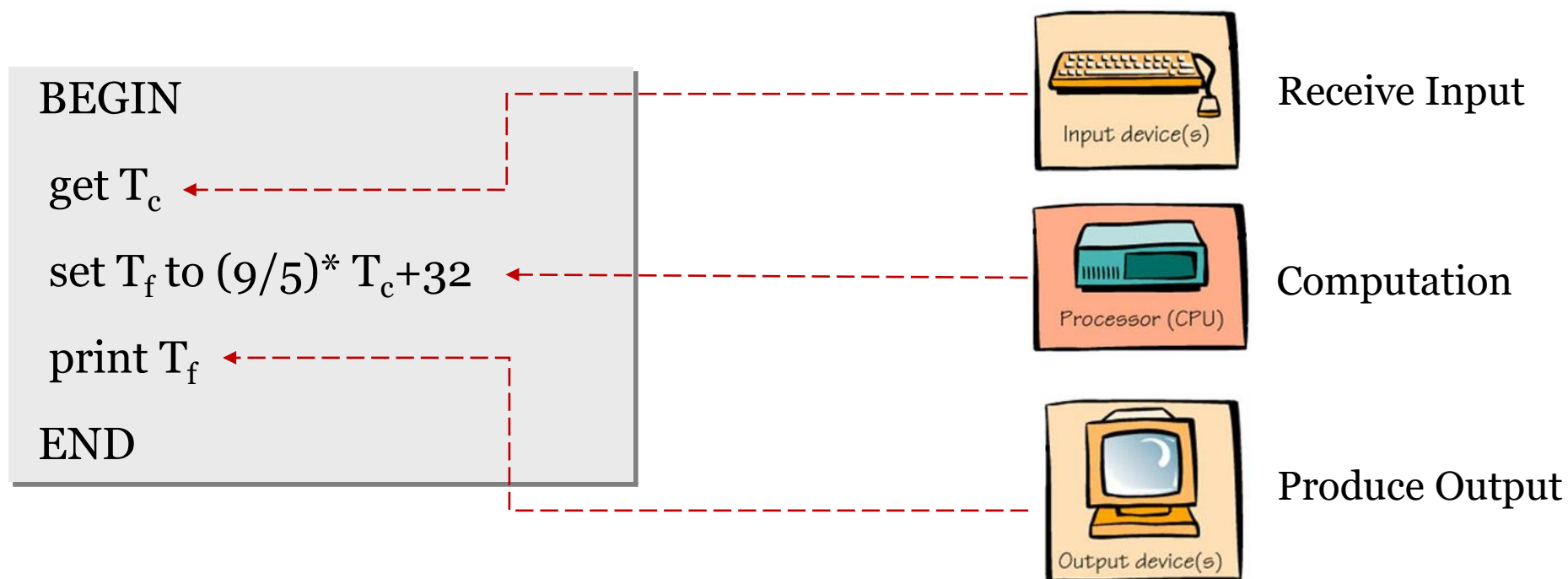
Sequential Algorithm



Types of Algorithms

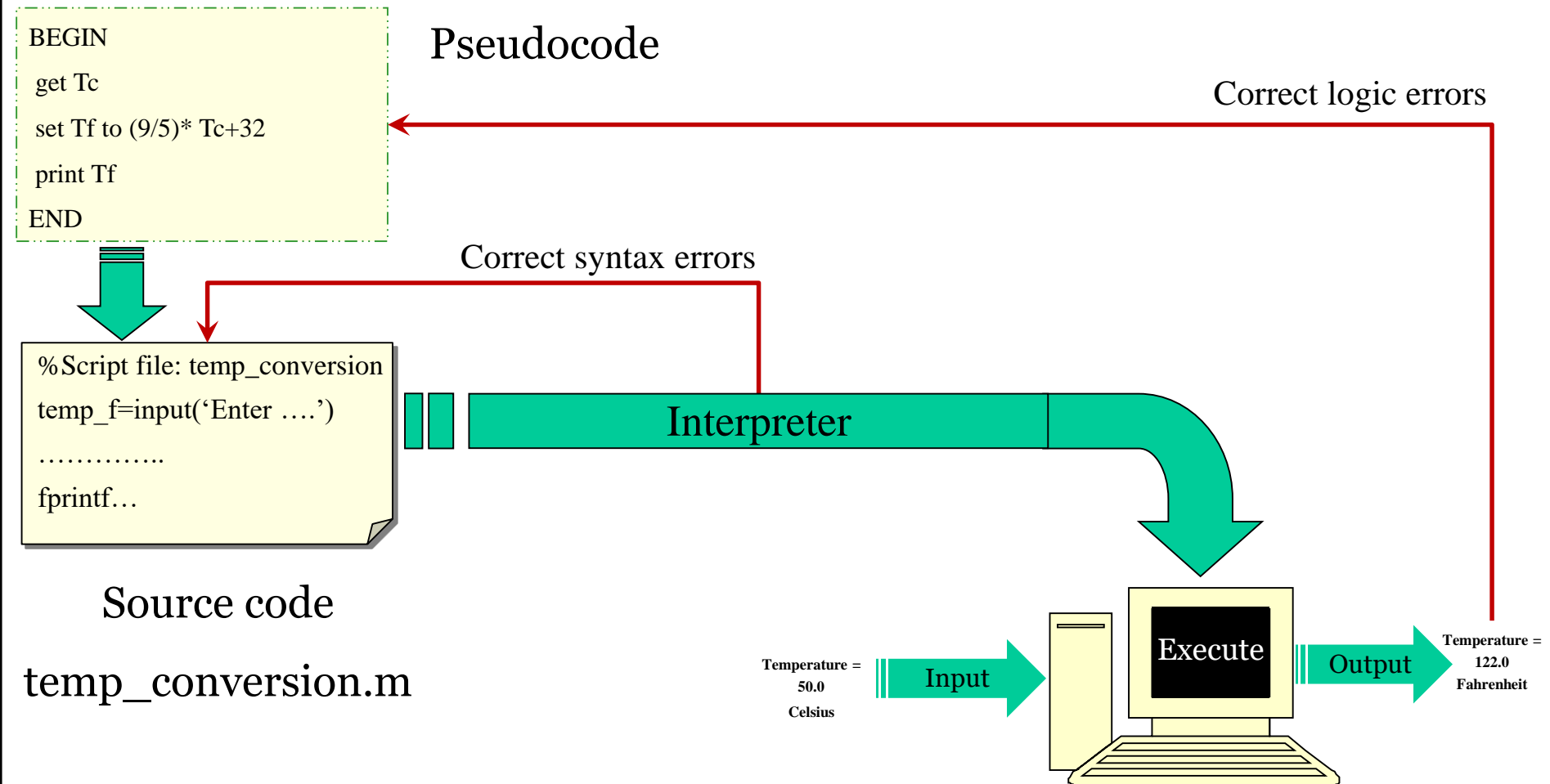
- **Sequential Algorithms**

Sequential algorithm is formed by a list of operations (or steps) arranged in a “linear” fashion, such that the order of these steps is well defined and significant.



Types of Algorithms

- Sequential Algorithms



Types of Algorithms

- **Sequential Algorithms**

```
>>  
>> temp_c=50  
  
temp_c =  
  
    50  
  
>> temp_f=(9/5)*(temp_c)+32  
  
temp_f =  
  
    122
```

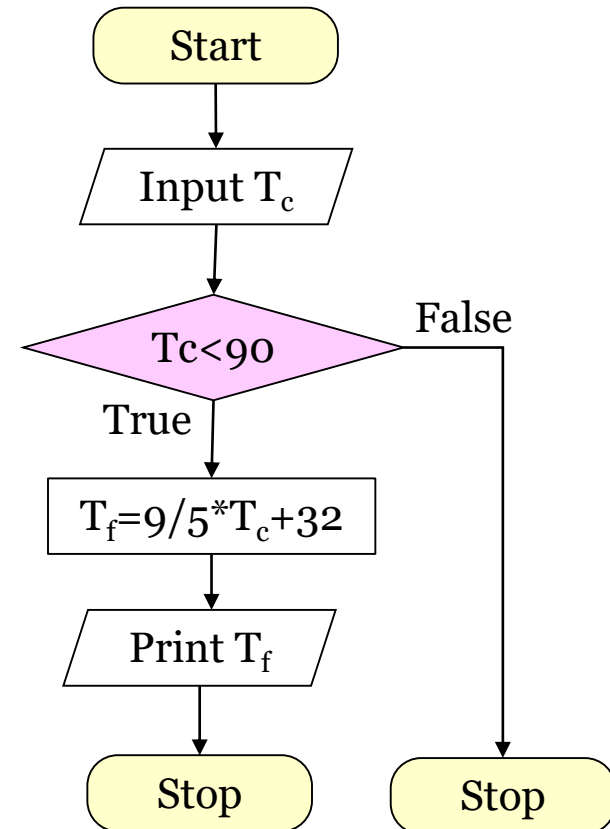
Types of Algorithms

Example: Convert temperature from Celsius to Fahrenheit using the following formula in the following cases:

$$T_f = 9/5 * T_c + 32$$

b. Convert the Celsius degree to Fahrenheit **if and only if** the Celsius degree is less than certain value, let's say (90°C)

Conditional Algorithm



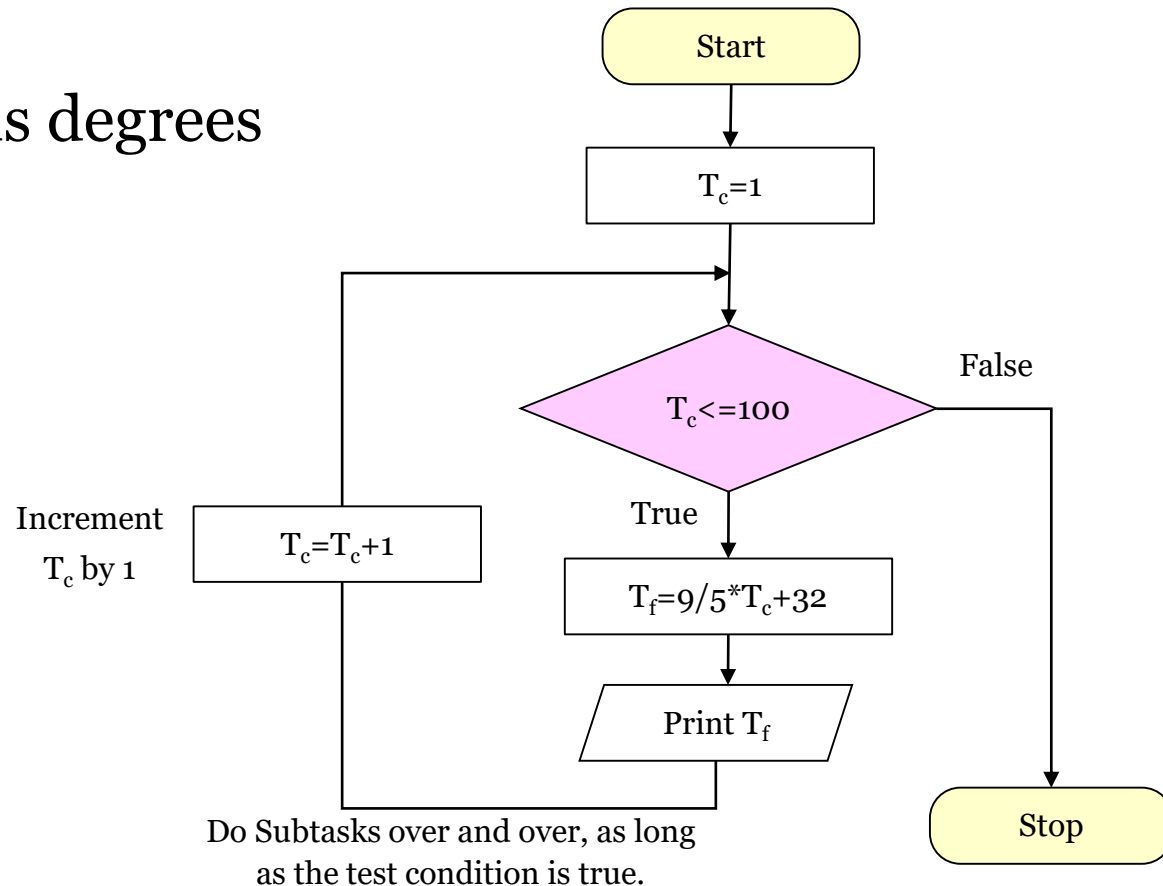
Types of Algorithms

Example: Convert temperature from Celsius to Fahrenheit using the following formula in the following cases:

$$T_f = 9/5 * T_c + 32$$

c. Convert range of Celsius degrees from (1°C) to (100°C)

Iterative Algorithm



Outline

- What is an Algorithm?
- Representing Algorithms
- Pseudo-code
- Algorithm Implementation
- Types of Algorithms
- Sequential Algorithms
- **Summary**

Summary

- An algorithm is a step-by-step specification of a method to solve a problem within a finite amount of time.
- Usually we use “pseudo-code” to describe algorithms.
- Pseudo-code is a notation resembling a programming language but not intended for actual compilation.
- Flowcharts can be thought of as a graphical form of pseudo-code.
- Algorithms can be implemented in any programming language.

Summary

- Sequential algorithm is formed by a list of operations (or steps) arranged in a “linear” fashion, such that the order of these steps is well defined and significant.
- Conditional operation is a control operation that allow us to alter the normal sequential flow of control in an algorithm. Conditional statements are the “question-asking” operations of an algorithm.
- An Iterative operation allows the repetition of a block of statements according to a condition. Iteration is sometimes called looping.