Programming Fundamental

Lecture 3

Algorithm & Flowchart

By

Asst. Lect. Ali Al-khawaja

Introduction to Algorithms

- A sequence of instructions.
- A procedure or formula for solving a problem.
- It was created mathematician, Mohammed ibn-Musa al-Khwarizmi.
- Often used for calculation, data processing and programming.
- Algorithms can be expressed in any language.

Introduction to Algorithms

- Algorithms for making things will often be divided into sections;
  - the parts/components/ingredients (**inputs**) required to accomplish the task
  - actions/steps/methods (**processing**) to produce the required outcome (**output**).
- For example to build a model car, the parts (**inputs**) are needed plus instructions on how to assemble the car (**processing**) and the result is the car (**output**).

Types of Algorithms

- Two types of Algorithm:
  - Pseudocode
  - Flow chart

Pseudocode

- Pseudocode (which means fake code, because its not really programming code) specifies the steps required to accomplish the task.
- Pseudocode is a type of structured English that is used to specify an algorithm.
- Pseudocode cannot be compiled nor executed, and there are no real formatting or syntax rules.

Pseudocode

Example of Pseudocode:

```
Ready and open subscriber file
Get a record
Do while more records
        If current subscriber subscription count is > 3 then
Output the record
        Get another record
end
```

Pseudocode

Advantages of Pseudocode

- Reduced complexity.
- Increased flexibility.
- Ease of understanding.

Pseudocode

Why is Pseudocode Necessary?

- The programming process is a complicated one.
- You must first understand the program specifications.
- Then you need to organize your thoughts and create the program.
- You must break the main tasks that must be accomplished into smaller ones in order to be able to eventually write fully developed code.
- Writing Pseudocode will save you time later during the construction & testing phase of a program's development.

Pseudocode

How to Write Pseudocode Statements?

1. A computer can receive information
   - Read (information from a file)
   - Get (information from the keyboard)

2. A computer can put out information
   - Write (information to a file)
   - Display (information to the screen)

Pseudocode

How to Write Pseudocode Statements?

3. A computer can perform arithmetic
   - Use actual mathematical symbols or the words for the symbols
   - Example:
     - Add number to total
     - Total = total + number
   - +, -, *, /
   - Calculate, Compute also used

Pseudocode

How to Write Pseudocode Statements?

4. A computer can assign a value to a piece of data

i. to give data an initial value

Initialize, Set

ii. to assign a value as a result of some processing

*x=5+y

iii. to keep a piece of information for later use

Save, Store

Pseudocode

How to Write Pseudocode Statements?

5. A computer can compare two piece of information and select one of two alternative actions

```
IF condition THEN
        some action
ELSE
        alternative action
ENDIF
```

Pseudocode

How to Write Pseudocode Statements?

6. A computer can repeat a group of actions

WHILE condition (is true)
some action
ENDWHILE

FOR a number of times
some action
ENDFOR

Pseudocode

Example 1:

Program Specification:

Write a program that obtains two integer numbers from the user. It will print out the sum of those numbers.

Pseudocode:

Prompt the user to enter the first integer
Prompt the user to enter a second integer
Compute the sum of the two user inputs
Display an output prompt that explains the answer as the sum
Display the result

Pseudocode

Example 2:

Finding average of any three numbers.

We might usually specify the procedure of solving this problem as “add the three numbers and divide by three”. Here, Read (or Ask) and Write (or Say) are implied. However in an algorithm, these steps have to be made explicit. Thus a possible algorithm is:

Pseudocode

Example 2:

Step 1	Start
Step 2	Read values of X, Y, Z
Step 3	S = X + Y + Z
Step 4	A = S / 3
Step 5	Write value of A
Step 6	Stop

Pseudocode

Example 2:

Or you can write like this:

```
Step 1      Start
Step 2          Read values of X, Y, Z
Step 3          S = X + Y + Z
Step 4          A = S / 3
Step 5          Write value of A
Step 6          Stop
```

Pseudocode

Example 3:

Finding square and cube.

```
Step 1      Start
Step 2          Read value of N
Step 3          S = N * N
Step 4          C = S * N
Step 5          Write values of S, C
Step 6          Stop
```

Pseudocode

Example 4:

Finding biggest of two numbers.

```
Step 1        Start
Step 2              Read A, B
Step 3              If A > B, then BIG = A, otherwise
     BIG = B
Step 4              Write BIG
Step 5              Stop
```

Pseudocode

Example 5:

Calculate pay.

Step 1	Start
Step 2	Input hours
Step 3	Input rate
Step 4	pay = hours * rate
Step 5	Print pay
Step 6	End

Pseudocode

Exercise:

Write a Pseudocode for these problems.

1. $S = (A + B + C) / Y$
2. Convert from Celsius to Fahrenheit.

   (Multiply by 9, then divide by 5, then add 32)
3. Area of Circle (πr^2)
4. Volume of Sphere ($\frac{4}{3}\pi r^3$)
5. Average speed = $\frac{\text{Distance traveled}}{\text{Time taken}}$

Flowchart

- A **flowchart** is a graphical representation of an algorithm.
- These flowcharts play a vital role in the programming of a problem and are quite helpful in understanding the logic of complicated and lengthy problems.
- Once the **flowchart** is drawn, it becomes easy to write the program in any high level language.

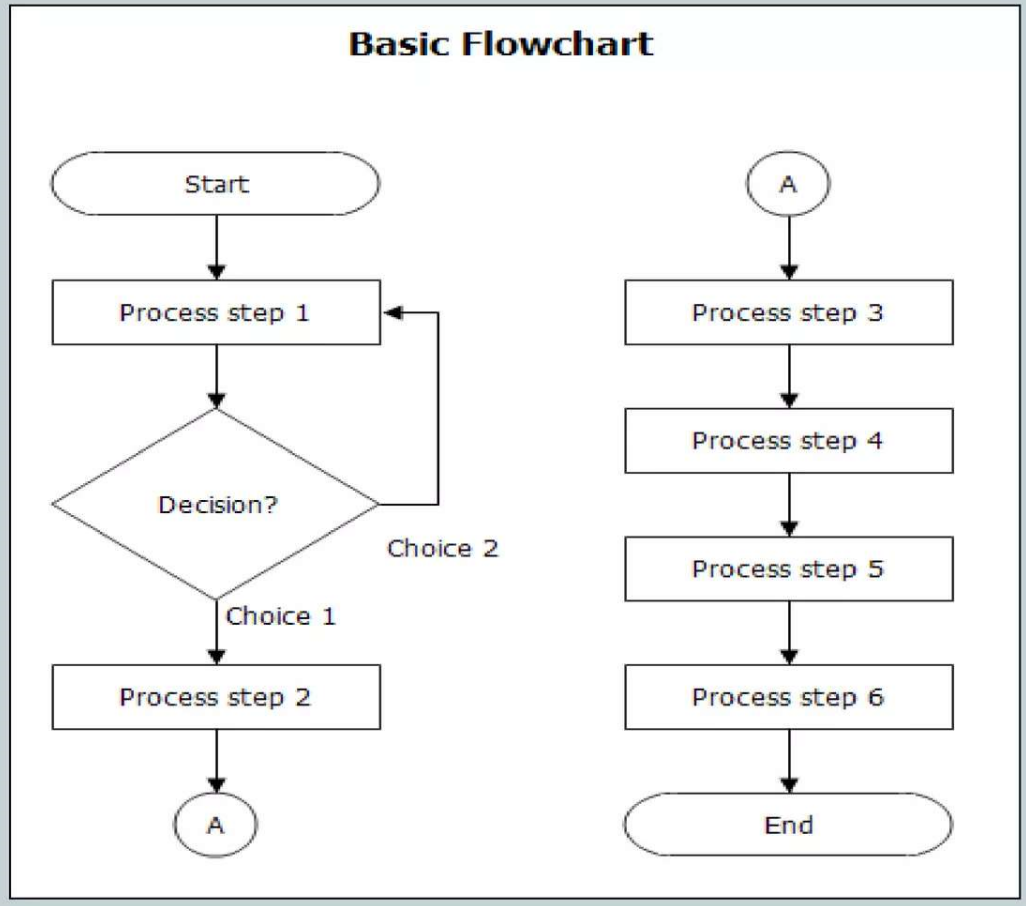
Flowchart

Flowchart

- A flowchart can therefore be used to:
  - Define and analyze processes
  - Build a step-by-step picture of the process for analysis, discussion, or communication
  - Define, standardize or find areas for improvement in a process

Flowchart

Flowchart



Flowchart

Flowchart Symbols

Start and **End** symbols

- Represented as lozenges, ovals or rounded rectangles
- Usually containing the word "Start" or "End", or another phrase signalling the start or end of a process, such as "submit enquiry" or "receive product".

Flowchart

Flowchart Symbols

Arrows

- Showing what's called "flow of control" in computer science.
- An arrow coming from one symbol and ending at another symbol.
- Represents that control passes to the symbol the arrow points to.

Flowchart

Flowchart Symbols

Processing steps

- Represented as rectangles.
- Examples: “Add 1 to X"; "replace identified part"; "save changes" or similar.

Flowchart

Flowchart Symbols

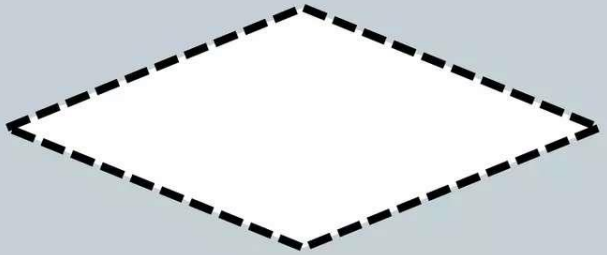
Input/Output

- Represented as a parallelogram.
- Examples: Get X from the user; display X.

Flowchart

Flowchart Symbols

Conditional or decision



- Represented as a diamond (rhombus).
- These typically contain a Yes/No question or True/False test.

Algorithms

Flowchart Symbols

Display

- Indicates a process flow step where information is displayed to a person (e.g., PC user, machine operator).

Flowchart

Rules for Flowchart

1. Every flow chart has a START symbol and a STOP symbol.
2. The flow of sequence is generally from the top of the page to the bottom of the page. This can vary with loops which need to flow back to an entry point.
3. Use arrow-heads on connectors where flow direction may not be obvious.
4. There is only one flow chart per page.

Flowchart

Rules for Flowchart

5. A page should have a page number and a title.
6. A flow chart on one page should not break and jump to another page
7. A flow chart should have no more than around 15 symbols (not including START and STOP).

Flowchart

Advantages of Using Flowcharts

- **Communication**: Flowcharts are better way of communicating the logic of a system to all concerned.
- **Effective analysis**: With the help of flowchart, problem can be analysed in more effective way.
- **Proper documentation**: Program flowcharts serve as a good program documentation, which is needed for various purposes.
- **Efficient Coding**: The flowcharts act as a guide or blueprint during the systems analysis and program development phase.

Flowchart

Advantages of Using Flowcharts

- **Proper Debugging**: The flowchart helps in debugging process.
- **Efficient Program Maintenance**: The maintenance of operating program becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part.

Flowchart

Basic Control Structures

- Sequence
- Selection
- Loop

Flowchart

Basic Control Structures

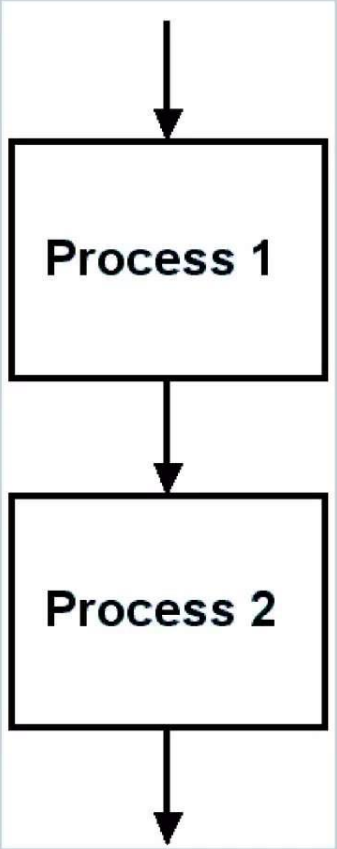
Sequence

- Steps that execute in sequence are represented by symbols that follow each other top to bottom or left to right.
- Top to bottom is the standard.

Flowchart

Basic Control Structures

Sequence



Flowchart

Basic Control Structures

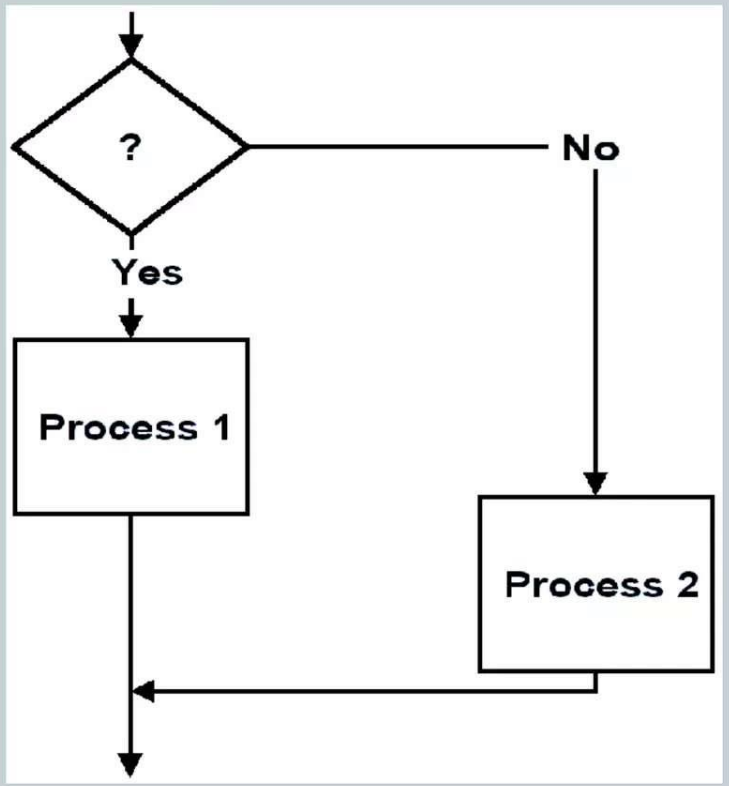
Selection/Branching

- Once the condition is evaluated, the control flows into one of two paths.
- Once the conditional execution is finished, the flows rejoin before leaving the structure.

Flowchart

Basic Control Structures

Selection/Branching



Flowchart

Basic Control Structures

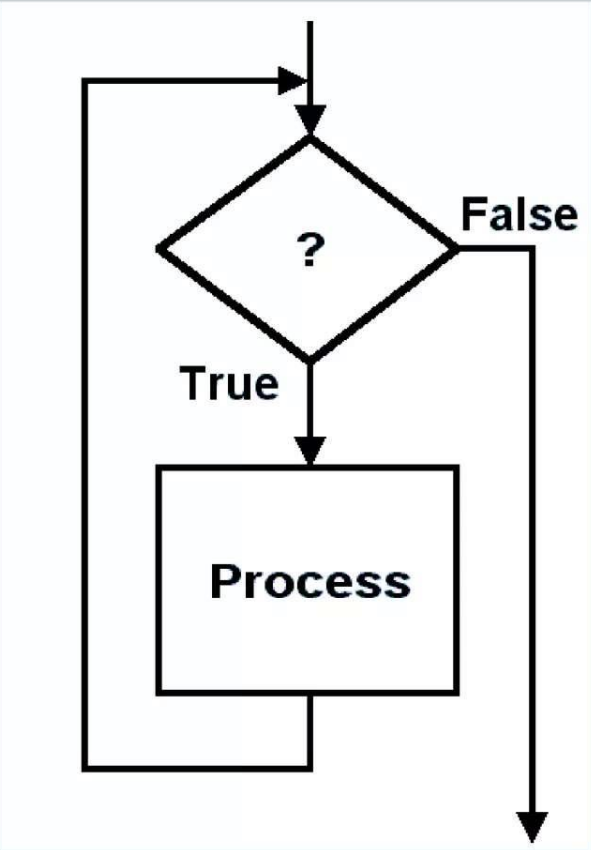
Loop

- Either the processing repeats or the control leaves the structure.
- Notice that the return line joins the entry line before the question.

Flowchart

Basic Control Structures

Loop



Flowchart

Example 1:

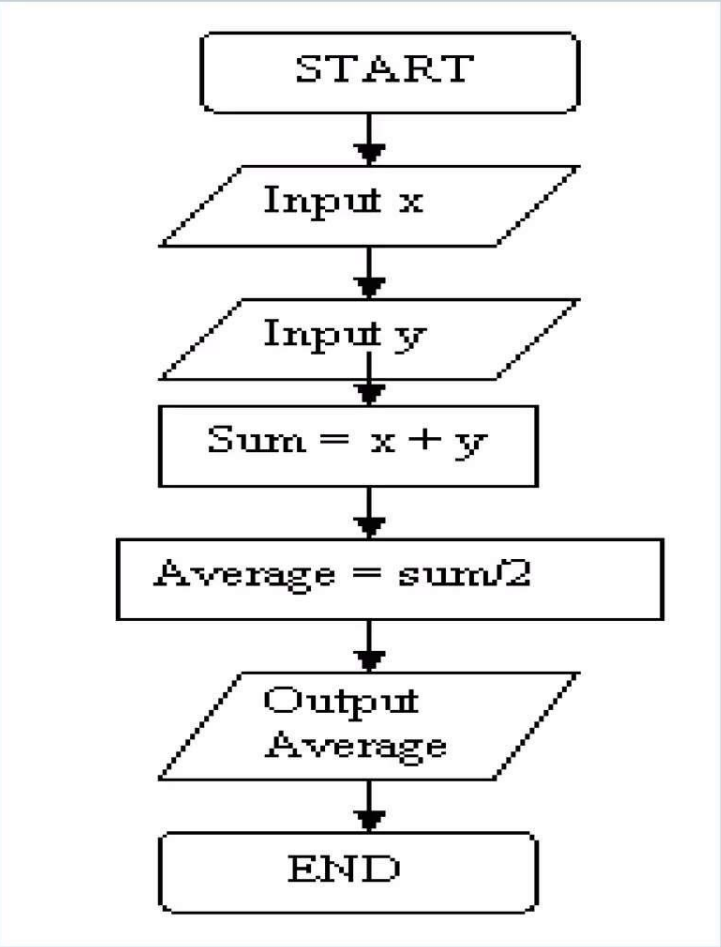
Algorithm:

Input: two numbers x and y
Output: the average of x and y
Steps:

1. input x
2. input y
3. sum = x + y
4. average = sum /2
5. output average

Flowchart

Example 1:



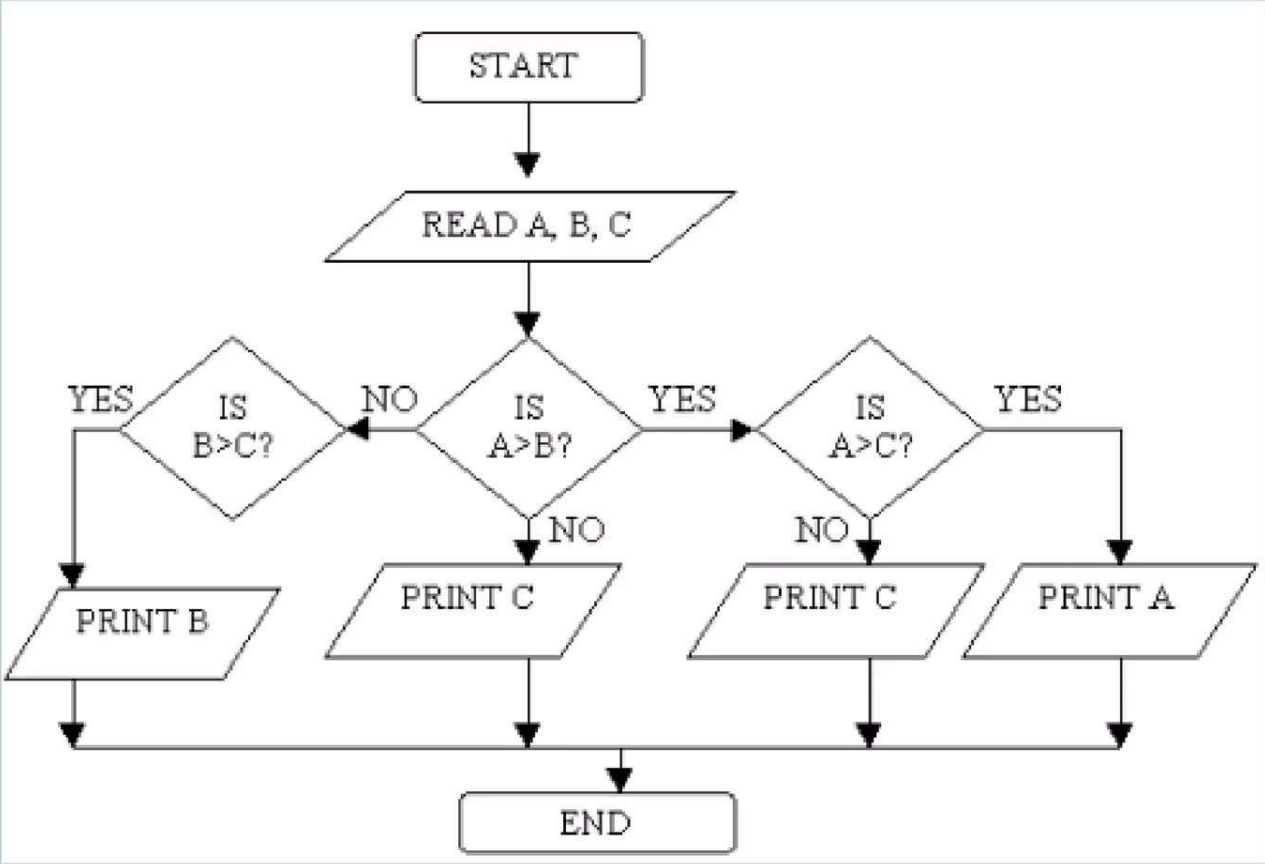
Flowchart

Example 2:

Draw a flowchart to find the largest of three numbers A, B, and C.

Flowchart

Example 2:



Flowchart

Exercise:

Draw a flowchart diagram.

1. A program that compare the first number and second number and display which one is the biggest.

2. Login screen to check the Username and Password.