ALGORITHMS AND FLOWCHARTS
ALGORITHMS AND FLOWCHARTS

• A typical programming task can be divided into two phases:

  • **Problem solving phase**
    • produce an ordered sequence of steps that describe solution of problem
    • this sequence of steps is called an *algorithm*

  • **Implementation phase**
    • implement the program in some programming language
STEPS IN PROBLEM SOLVING

• First produce a general algorithm (one can use pseudocode)

• Refine the algorithm successively to get step by step detailed algorithm that is very close to a computer language.

• Pseudocode is an artificial and informal language that helps programmers develop algorithms. Pseudocode is very similar to everyday English.
• **Example 1:** Write an algorithm to determine a student’s final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.
PSEUDOCODE & ALGORITHM

Pseudocode:

• Input a set of 4 marks
• Calculate their average by summing and dividing by 4
• if average is below 50
    Print “FAIL”
else
    Print “PASS”
PSEUDOCODE & ALGORITHM

• Detailed Algorithm

• Step 1: Input M1, M2, M3, M4

• Step 2: GRADE ← (M1 + M2 + M3 + M4) / 4

• Step 3: if (GRADE < 50) then

    Print “FAIL”

    else

    Print “PASS”

endif
THE FLOWCHART

• (Dictionary) A schematic representation of a sequence of operations, as in a manufacturing process or computer program.

• (Technical) A graphical representation of the sequence of operations in an information system or program. Information system flowcharts show how data flows from source documents through the computer to final distribution to users. Program flowcharts show the sequence of instructions in a single program or subroutine. Different symbols are used to draw each type of flowchart.
THE FLOWCHART

A Flowchart

• shows logic of an algorithm
• emphasizes individual steps and their interconnections
• e.g. control flow from one action to the next
# FLOWCHART SYMBOLS

## Basic

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Use in Flowchart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oval</td>
<td><img src="oval.png" alt="Oval" /></td>
<td>Denotes the beginning or end of the program</td>
</tr>
<tr>
<td>Parallelogram</td>
<td><img src="parallelogram.png" alt="Parallelogram" /></td>
<td>Denotes an input operation</td>
</tr>
<tr>
<td>Rectangle</td>
<td><img src="rectangle.png" alt="Rectangle" /></td>
<td>Denotes a process to be carried out e.g. addition, subtraction, division etc.</td>
</tr>
<tr>
<td>Diamond</td>
<td><img src="diamond.png" alt="Diamond" /></td>
<td>Denotes a decision (or branch) to be made. The program should continue along one of two routes. (e.g. IF/THEN/ELSE)</td>
</tr>
<tr>
<td>Hybrid</td>
<td><img src="hybrid.png" alt="Hybrid" /></td>
<td>Denotes an output operation</td>
</tr>
<tr>
<td>Flow line</td>
<td><img src="flow_line.png" alt="Flow line" /></td>
<td>Denotes the direction of logic flow in the program</td>
</tr>
</tbody>
</table>
Step 1: Input M1, M2, M3, M4
Step 2: GRADE ← (M1 + M2 + M3 + M4) / 4
Step 3: if (GRADE < 50) then
      Print “FAIL”
      else
      Print “PASS”
      endif
EXAMPLE 2

• Write an algorithm and draw a flowchart to convert the length in feet to centimeter.

Pseudocode:
• Input the length in feet (Lft)
• Calculate the length in cm (Lcm) by multiplying LFT with 30
• Print length in cm (LCM)
Algorithm

• Step 1: Input Lft
• Step 2: Lcm ← Lft x 30
• Step 3: Print Lcm
EXAMPLE 3

Write an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.

Pseudocode

• Input the width (W) and Length (L) of a rectangle
• Calculate the area (A) by multiplying L with W
• Print A
EXAMPLE 3

Algorithm

• Step 1: Input W,L
• Step 2: A ← L x W
• Step 3: Print A
EXAMPLE 4

• Write an algorithm and draw a flowchart that will calculate the roots of a quadratic equation

\[ ax^2 + bx + c = 0 \]

\[ b^2 - 4ac \]

• Hint: \( d = \sqrt{ \ } \), and the roots are: \( x1 = (-b + d)/2a \) and \( x2 = (-b - d)/2a \)
EXAMPLE 4

Pseudocode:
• *Input the coefficients* \((a, b, c)\) *of the quadratic equation*
• *Calculate* \(d\)
• *Calculate* \(x_1\)
• *Calculate* \(x_2\)
• *Print* \(x_1\) *and* \(x_2\)
EXAMPLE 4

• Algorithm:

• Step 1: Input a, b, c

• Step 2: 

• Step 3: 

• Step 4: 

• Step 5: Print x1, x2

\[
b \times b - 4 \times a \times c
\]
DECISION STRUCTURES

The expression $A > B$ is a logical expression: it describes a condition we want to test.

- If $A > B$ is true (if $A$ is greater than $B$), we take the action on left:
  - Print the value of $A$
- If $A > B$ is false (if $A$ is not greater than $B$), we take the action on right:
  - Print the value of $B$
DECISION STRUCTURES

Y
is
A>B
N
Print A
Print B
IF–THEN–ELSE STRUCTURE

• The structure is as follows

If condition then
ture alternative
else
false alternative
endif
IF–THEN–ELSE STRUCTURE

- The algorithm for the flowchart is as follows:

  If $A > B$ then
  
  print $A$

  else
  
  print $B$

  endif
## RELATIONAL OPERATORS

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>≥</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>≤</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>≠</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>
EXAMPLE 5

• Write an algorithm that reads two values, determines the largest value and prints the largest value with an identifying message.

ALGORITHM
Step 1: Input VALUE1, VALUE2
Step 2: if (VALUE1 > VALUE2) then
    MAX ← VALUE1
else
    MAX ← VALUE2
endif
Step 3: Print “The largest value is”, MAX
EXAMPLE 5

START

Input VALUE1, VALUE2

Y

is VALUE1 > VALUE2

N

MAX ← VALUE1

MAX ← VALUE2

Print “The largest value is”, MAX

STOP
NESTED IFS

• One of the alternatives within an IF–THEN–ELSE statement
  • may involve further IF–THEN–ELSE statement
EXAMPLE 6

• Write an algorithm that reads three numbers and prints the value of the largest number.
EXAMPLE 6

Step 1: Input N1, N2, N3

Step 2: if (N1>N2) then
    if (N1>N3) then
        MAX ← N1  \[N1>N2, N1>N3\]
    else
        MAX ← N3  \[N3>N1>N2\]
    endif
else
    if (N2>N3) then
        MAX ← N2  \[N2>N1, N2>N3\]
    else
        MAX ← N3  \[N3>N2>N1\]
    endif
endif

Step 3: Print “The largest number is”, MAX
EXAMPLE 6

- Flowchart: Draw the flowchart of the above Algorithm.
EXAMPLE 7

Write and algorithm and draw a flowchart to

a) read an employee name (NAME), overtime hours worked (OVERTIME), hours absent (ABSENT) and

b) determine the bonus payment (PAYMENT).