

## Structured Programming II: If Statements

By the end of this class you should be able to:

- implement branching in a program
- describe and use an "if" statement
- use "else" statements
- use "elseif" statements

Text: Section 4.4

## Review Exercise: Temperature Data Analysis

Get into groups of 3 (2 if necessary)

Download → [FWTemperature.txt](#) from the class website

This file contains the daily high and low temperatures at FWA

Row 1 → Day of the Month

Row 2 → Highs for March 2007

Row 3 → Lows for March 2007

Row 4 → Highs for April 2007

Row 5 → Lows for April 2007

Develop and execute a script to:

- load in this file
- Determine and display the number of days in each month where the highs were 60 degrees or higher.
- Determine and display the number of days in each month where the highs were between 50 and 60 degrees

## Start MATLAB and set up this function template m-file (test.m)

```
function y = test(x)
% This is a function template set up as
% convenient way to test program structures.
% y = test(x)
% your name           March 2008
```

## If statement

If  $x$  is greater than or equal to zero  
then  $y$  equals the square root of  $x$

```
if x >= 0
    y = x^0.5;
end
```

Add these lines to your test.m file  
and try it out with several inputs

e.g.  $x = 2$ ,  $x = 0$ ,  $x = -2$

## See Conditional Statements 1 Handout

You cannot use radical in MATLAB,  
use `sqrt(x)` or `x.^0.5`

Tests in Command Window

```
>> test(2)
ans =
    1.4142
>> test(0)
ans =
     0
>> test(-2)
```

Try a vector  $x = [2, 0, -2]$

```
>> test([2, 0, -2])
>>
```

Nothing happens: When test a logical vector in an if statement all cases must be true for the if to go to the true command statements.

Try assigning an output to the last case:

```
>> y = test(-2)
```

??? Output argument "y" (and maybe others) not assigned during call to "C:\Documents and Settings\Administrator\My Documents\MATLAB\test.m (test)".

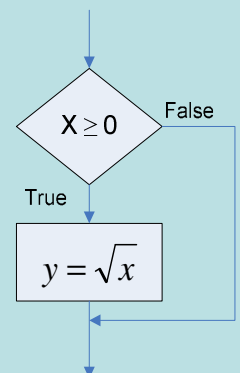
Error in => test at 7  
if  $x \geq 0$

You get an error because for a negative number no value is assigned to the output. We will look at how to fix this with the next program.

## Example Flowchart for this simple if code block

• Diamonds = conditional branch points

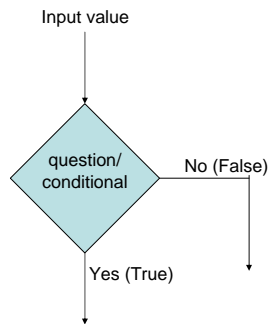
• Rectangles = executable statements



## Flow Charts of Conditionals

Conditionals (e.g. if statements) are represented by diamonds:

- Input comes in the top
- A question or conditional statement is in the diamond
- Two lines leave - one for yes (true), one for no (false)
- One diamond may feed another



## If/else statements

*If  $x$  is greater than or equal to zero then  $y$  equals the square root of  $x$  or else display an error & set  $y$  to NaN*

```
if x >= 0
    y = sqrt(x);
else
    disp('Error: input is negative')
    y = NaN;
end
```

Add this else error message to your test.m file and try

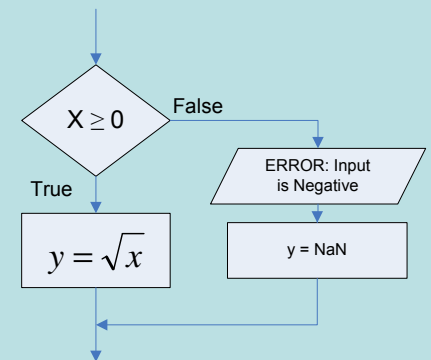
Repeat earlier test cases with this new addition to the m-file in MATLAB:

```
>> test(2), test(0), test(-2)
ans =
    1.4142
ans =
     0
ERROR: Input is Negative
ans =
NaN
```

Note: NaN is a special MATLAB code for "not a number." This code makes sure the output of the function has something in it so errors are not encountered in future use.

## Example Flowchart for this if-else structure

- Parallelograms = Input/Output Steps



## Example Function: eng2PL1.m

Download eng2PL1.m from class website  
This function takes a one word input (in quotes)

Try it with several words including

```
>> eng2PL1('banana')
>> eng2PL1('orange')
```

What does it do that our last translator did not do?

Look at the code - How does it work?

## eng2PL1.m

```
function x = eng2PL1(x)
% Lots o' Comments

% make all letters lower case and set up list of vowels & consonants
x = lower(x);

if any(x(1) == 'aeiou')
    % if x starts in a vowel simply add 'ay'
    x = [x, 'ay'];
else
    % otherwise move leading consonant to the end of the word and add 'ay'
    x = [x(2:end), x(1), 'ay'];
end
```

## If / elseif / else

If  $x$  is greater than zero  
then  $y$  equals the square root of  $x$   
or else if  $y$  is equal to zero  
then display " $x = 0$ " & set  $y$  to zero  
or else  
display "input is negative" & set  $y$  to NaN

## MATLAB Example: Modify your test.m code to match

```
if x > 0
    y = x^0.5;
elseif x == 0
    disp('input is zero')
    y = 0;
else
    disp('ERROR: input is negative')
    y = NaN;
end
```

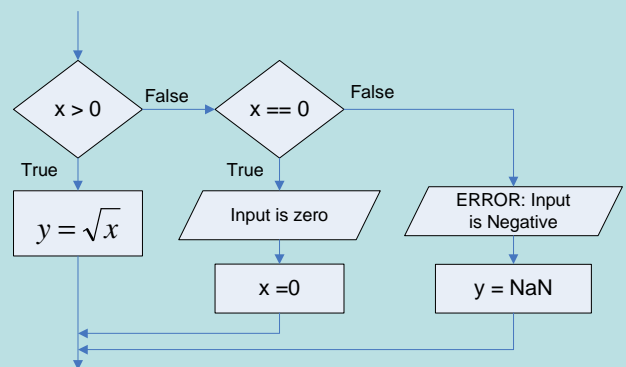
Add this elseif segment (& change the first condition) in your test.m file. Repeat earlier test cases in MATLAB:

```
>> test(2), test(0), test(-2)
ans =
    1.4142
input is zero
ans =
    0
ERROR: Input is Negative
ans =
    NaN
```

What happens if you only put one equal sign in the else if statement?

```
>> test(2), test(0), test(-2)
??? Error: File: test.m Line: 9 Column: 10
The expression to the left of the equals sign is not a valid target for an assignment.
It attempts to do an assignment, which is not legal here.
```

## Example flowchart for if-elseif-else structure



## Nested if: alternative logic for the previous example

```
if x >= 0
    if x == 0
        disp('input is zero')
        y = 0;
    else
        y = sqrt(x);
    end
else
    disp('ERROR input is negative')
    y = NaN;
end
```

This "inner if" is only executed if the "outer if" is true

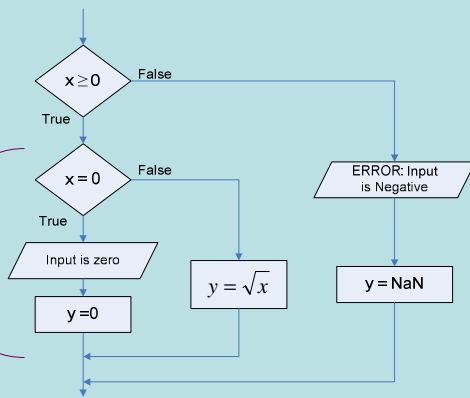
Revise your earlier file to follow this nested if structure. Repeat earlier test cases with this new addition to the m-file in MATLAB:

```
>> test(2), test(0), test(-2)
ans =
    1.4142
ans =
    0
ERROR: Input is Negative
ans =
    NaN
```

This type of nesting can be done in either the "true" section of the if statement (as shown here) or it can be done in the "false" section of the if statement where it would be executed only when the "outer" if is false.

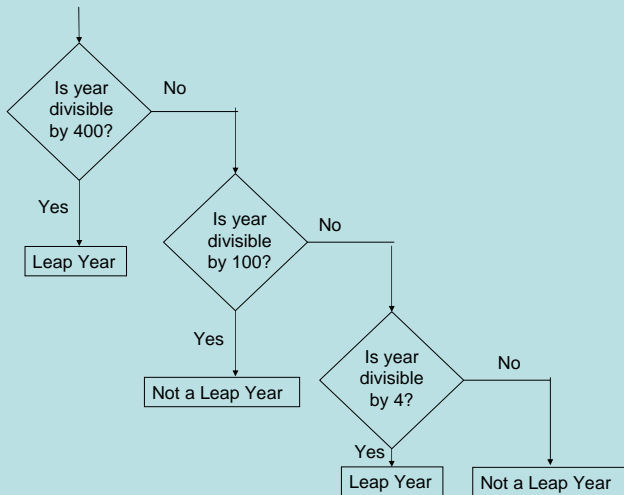
### Example Flowchart for this nested if structure

Again notice this "inner if" is only executed if the "outer if" is true



### Leap Year Exercise:

- In groups of 2 or 3 arrange the cards to form a flow chart for the leap year rules.
- Diamonds represent conditions, Rectangles results.
- The rules:
  1. All years evenly divisible by 400 are leap years
  2. All years evenly divisible by 100 are not leap years (except when condition 1 is met),
  3. Years divisible by 4 but not by 100 are leap years (in addition to condition 1),
  4. All other years are not leap years.



### Another helpful function

mod(numerator, denominator)

try for:

12, 3

11, 3

10, 3

9, 3

Can you see what it does?

How could this be used to determine if a year is evenly divisible by 4.

Create a function that will determine if a given year is a leap year or not.