MATH.2720 Introduction to Programming with MATLAB
Logical Variables and Programming Structures, Part 1

A. Logical Variables and Operations

So far we have encountered two MATLAB data types: numerical and character (strings). MATLAB also handles logical variables. A logical variable can take on only 2 values: 1 (true) or 0 (false). For example, the variable `s` defined by

```
>>s = 1 < 2
```

has the value 1 (true) because 1 is less than 2. The variable `r` defined by

```
>>r = 1 < 1
```

has the value 0 because 1 is not less than 1.

New logical variables can be created from existing variables using the operations `AND`, `OR`, and `NOT`. The action of these operations is described in the following table, in which A and B are assumed to be previously defined logical variables.

Logical Operation	MATLAB Syntax	Description
AND	`A & B` or `and(A, B)`	`A & B` = 1 if `A`=1 and `B`=1. Otherwise, `A & B` = 0 If `A` and `B` are arrays with the same dimensions, `A & B` is an array obtained by element-by-element `AND`ing.
OR	`A | B` or `or(A, B)`	`A | B` = 1 if `A`=1 or `B`=1 or both. `A | B` = 0 if both `A` and `B` = 0 If `A` and `B` are arrays with the same dimensions, `A | B` is an array obtained by element-by-element `OR`ing.
NOT	`~A` or not(A)	`~A`=1 if `A` = 0 and `~A` = 0 if `A` = 1. If `A` is an array,`~A` is an array obtained by element-by-element `NOT`ing.
Exclusive OR	`xor(A, B)`	`xor(A, B)` = 1 if exactly one of `A` and `B` is true. Otherwise `xor(A, B)` = 0

Below is a table showing the MATLAB syntax for the relational operators. Note the use of the double equal sign to test for equality.

Relational Operator	MATLAB Syntax
$<$	`<`
$\leq$	`<=`
$>$	`>`
$\geq$	`>=`
$=$	`==`
$\neq$	`~=`

Here are some additional MATLAB commands you may find useful.

MATLAB Command	Description
`all(A)`	`all(A)` = 1 if all elements of logical array `A` are true. Otherwise `all(A)` = 0
`any(A)`	`any(A)` = 1 if any element of logical array `A` is true. Otherwise `any(A)` = 0 if all elements of `A` are false.
`find(A)`	`find(A)` returns an array containing the indices of the true elements of array `A`

Try these commands to see what they do.

```
>>A = [1 0 1 0]
>>B = [0 1 0 1]
>>A & B
>>A | B
>>~A
>>v = -2:2
>>w = v.^2
>>A = v==w
>>B = v>=w
>>find(v<w) %What do these numbers mean?
```

B. `If` Programming Structures

Sometimes a procedure you want to implement will take different forms depending on the input. MATLAB provides several ways to do this.

(i) *if - end*

(From Gilat, *MATLAB: An Introduction with Applications.*) Workers get paid 1.5 times their normal hourly wage for each hour they work beyond 40 hours in a week. Create and run a script file containing the following commands to calculate a worker's weekly pay. Remember, you don't have to type in the comments. They are there just to tell you what the commands are doing.

```
t = input('Please enter the number of hours worked ');
h = input('Please enter the hourly wage in $ ');
pay = t*h;
if t>40
    pay = pay + (t-40)*0.5*h; % This command is only executed if t > 40
end
fprintf('The worker''s pay is $ %5.2f\n', pay)
```

(ii) *if - else - end*

The *if - else - end* structure provides a way of dealing with two different cases. Create a script file containing the following commands to calculate a worker's weekly pay another way.

```
t = input('Please enter the number of hours worked ');
h = input('Please enter the hourly wage in $ ');
if t<=40
    pay = t*h; % If t<40, pay is calculated this way
else
    pay = 40*h + (t-40)*1.5*h; % If t>=40, pay is calculated this way
end
fprintf('The worker''s pay is $ %5.2f\n', pay)
```

(iii) *if - elseif - else - end*

If you have to deal with more than two different cases, you can use the *if - elseif - else - end* structure. Create a script file containing the following commands to generate a letter grade from a numerical average.

```
avg = input('Please enter your numerical grade average ');
if (avg < 0) | (avg > 100)
    disp('Please enter a number between 0 and 100')
    avg = input('Please enter your numerical grade average ');
end
if avg > 90
    grade = 'A'; % If avg >90 this command is executed.
elseif avg > 80
    grade = 'B'; % If 80 < avg <= 90 this command is executed.
elseif avg > 70
    grade = 'C'; % If 70 < avg <= 80 this command is executed.
else
    grade = 'F'; % If avg <= 70 this command is executed.
end
fprintf('Your letter grade is %s\n', grade)  % The %s format is for strings (characters).
```

C. The `switch - case` Programming Structure

The `switch - case` programming structure is sometimes more convenient to use than an `if` structure. Create and run a script file containing the following commands to see how `switch - case` works. The file contains code to convert temperature values from one scale to another.

```
T = input('Enter temperature value to be converted ');
Units_in = input('Enter units of the temperature value you entered (C, K, F) ','s');
% The 's' at the end of this command tells MATLAB that the input is a character string
Units_out = input('Enter units you want to convert to (C, K, F) ','s');
err = 0;
switch Units_in
case 'C'
    T_K = T + 273;
case 'K'
    T_K = T;
case 'F'
    T_K = 273 + 5*(T - 32)/9;
otherwise
    err = 1;
end
switch Units_out
case 'C'
    T_out = T_K - 273;
case 'K'
```

```
    T_out = T_K;
case 'F'
    T_out = 9*(T_K - 273)/5 + 32;
otherwise
    err = 1;
end
if err
    disp('Error in entering current or new temperature scale')
else
    fprintf('Temperature = %5.1f %s\n',T_out,Units_out)
end
```

Practice Problems.

1. Go to the *Class Handouts etc.* link on our course web page. Right click on *Dr. Colby's data file for Use in Class on February 20*, and save the file to your desktop.

   In MATLAB, click on the *Import Data* icon on the toolbar. Click on the file UMLdata.

   Highlight column D, then click on *Import Selection*.

   (This column contains the maximum recorded temperature at UMass Lowell for the dates 1/1/2014 - 6/30/2014.) Use the `find` and `length` commands to calculate the number of days the maximum temperature was greater than or equal to 60°F. (Answer: 66).

2. Create a script file that asks the user for his or her average grade. If the grade is greater than or equal to 60, print the message "Congratulations! You pass." If the grade is less than 60, print the message "Sorry, you do not pass."

3. Create a script file that asks the user how many numbers on his/her lottery ticket match the winning number and then calculates the user's prize. If no numbers match, the prize is \$0; if one number matches, the prize is \$5; if two numbers match, the prize is \$50; and if three numbers match the prize is \$1,000.

4. (From Attaway, *MATLAB: A Practical Introduction to Programming and Problem Solving*, 2nd ed.) Whether a storm is a tropical depression, tropical storm, or hurricane is determined by the average sustained wind speed. A storm is a tropical depression if the winds are less than 38 miles per hour, a tropical storm if the winds are between 39 and 73 miles per hour, and a hurricane if the wind speeds are greater than or equal to 74 miles per hour. Write a script file that asks the user for the wind speed and prints out which type the storm is.

5. Create a function file using the `switch-case` structure that takes a student's quiz score (1 - 5) as input and produces a letter grade as ouptut, according to the following table.

Score	5	4	3	2	1
Grade	A	B	C	D	F