CS 221 Lecture

Tuesday, 20 September 2011

Today's Agenda

1. Announcements
2. DeMorgan's Laws
3. Model of Operation
4. MATLAB: Catch-up Topics
   - Syntax and Semantics
   - Input and Output
   - Strings
   - Conditional Statements in MATLAB
5. Homework Hints

Announcements

- Problem set 1 due tomorrow night (9/21)!
- Lab Quiz 1 next Thursday (9/29)
  - Coverage: Excel and MATLAB fundamentals
  - Preparation: anything you've been asked to do in lab or homework so far
    - Formulas and expressions
    - Conditionals – translate description into boolean expression
- In-class Quiz 1 in two weeks (10/4)
  - Coverage: everything in lecture and lab
  - Prep: practice exam next week

DeMorgan's Laws

- Negation (NOT, ~) distributes over conjunction, but turns it into disjunction:

  $\sim(A \wedge B) = \sim A \vee \sim B$

  NOT(AND(A,B)) = OR(NOT(A), NOT(B))

- Negation distributes over disjunction, turning it into conjunction:

  $\sim(A \vee B) = \sim A \wedge \sim B$

  NOT(OR(A,B)) = AND(NOT(A), NOT(B))

- Negation is its own inverse:

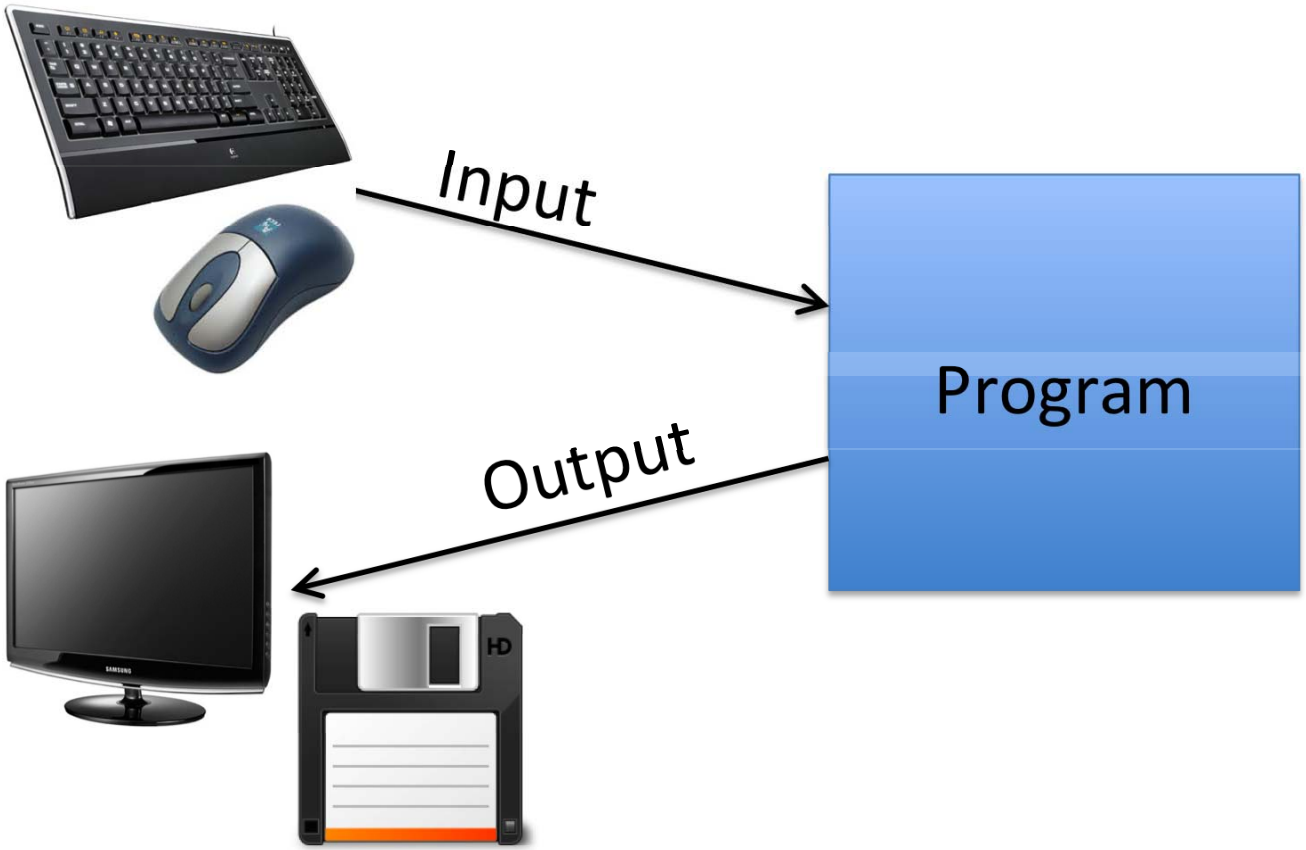
  $\sim\sim A = A$

  NOT(NOT(A)) = A

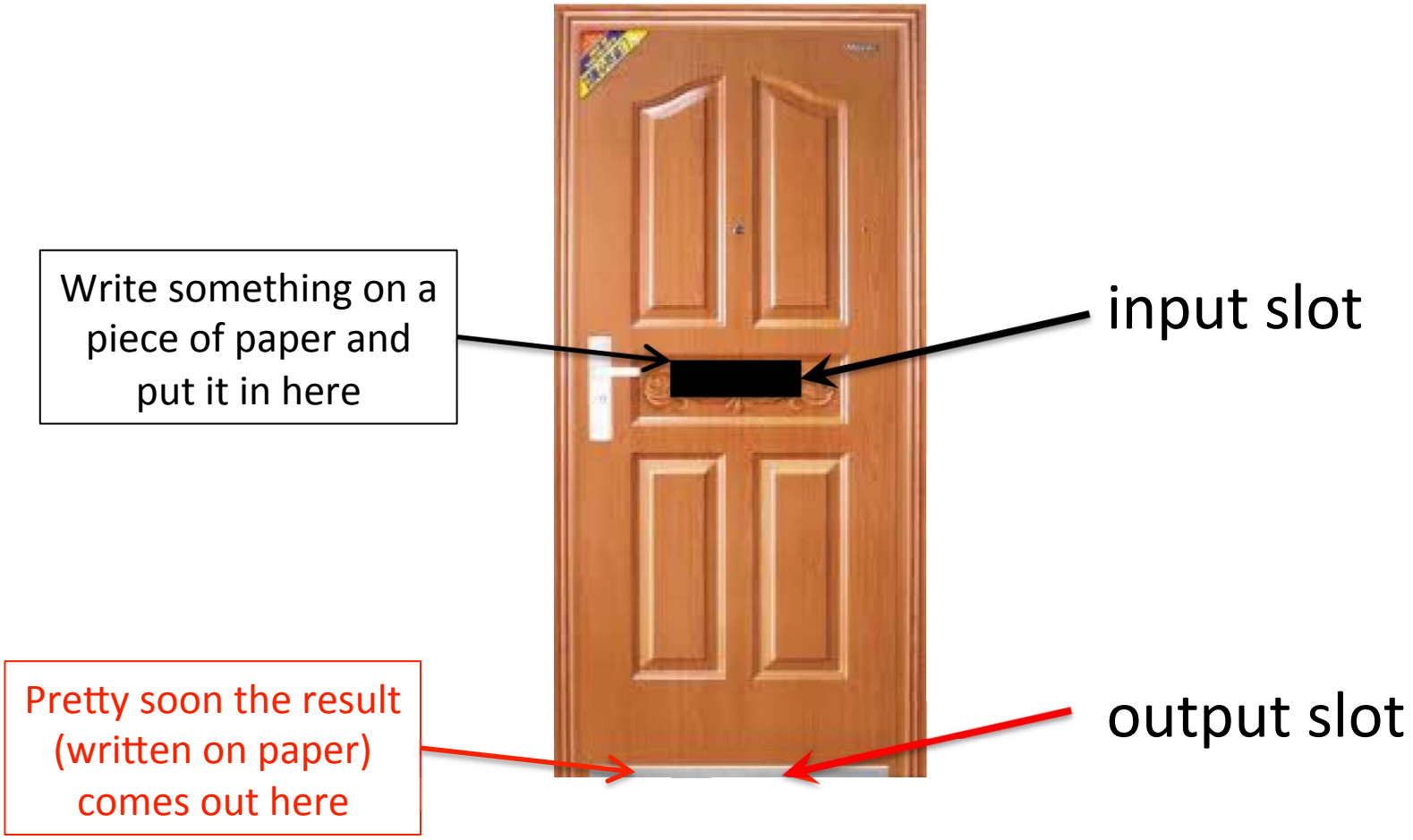
... For any boolean expressions A and B

Thinking About What MATLAB Does

- General model of a running program:



Model of Operation



What's Happening Behind the Door

Genie

Chest of Drawers containing values

What the Genie Does

When a command comes through the slot:

- Parse it to see if it is understandable (syntactically valid)
- If valid:
  - carry out the command; output the result
- else:
  - output an error message describing the problem

Syntax and Semantics

- Syntax: the structure of the commands the Genie (program) understands
  - Think grammar
  - Example: "sentence not this"

    "brilliant gathering slowly icicle"
- Semantics: the meaning of a properly-structured command

MATLAB Commands

Commands come in three flavors:

- Built-in commands that control MATLAB
  - Examples: clc, format, help, who ...
- Names of scripts
  - Script: file that contain a sequence of commands
- Expressions
  - Made up of:
    - constants
    - variable names
    - operators – including functions like sqrt() and mod()

Variables in MATLAB

- Variables: labeled "drawers" for holding values
  - Names must begin with a letter
  - Any combination of letters, numbers and underscores
    - Tip: don't use variable names longer than 63 characters
  - Case sensitive
    - ThisIsAValidVariableName
    - car47velocity
    - initial_condition
  - Invalid names:
    - 2ndDerivative
    - map-distance
    - tax_rate_%

Interpreting Commands

- if the command contains an operator:
  (it is an expression)
  - evaluate the operands (subexpressions)
  - apply the operator to the results (operands)
- elseif the command is a numeric constant
  - assign the value of the constant to the variable ans
- elseif the command is the name of a variable
  - assign the value of the variable to the variable ans
- else
  - look for a built-in command or script (m-file) that matches the word; if one exists, execute it, else print error message end
- end

Assignment

- Note: "=" is the assignment operator
  - It is not the same thing as equality!
- Assignment expressions have the form:

  <variable name> = <expression>

  Assignment is not commutative!
  - Variable name must be on LHS
  - RHS evaluated first
  - RHS expression may contain <variable name>

    e.g., t = t+1 or x = x - y

Semicolon

- Normally the Genie produces output that describes what assignment was done
- Putting a semicolon after the expression suppresses that output
- But the assignment still happens

Scripts

- A script or m-file is a file containing a sequence of MATLAB commands
  - suffix: .m
- When the Genie executes a script, it behaves as if each command had come through the slot individually – in order, one at a time
- Scripts save typing by allowing you to repeat the same computation over and over

Input and Output

- The input() function allows interaction with the user
  - used inside a script to get the input
  - Use like this:

    <variable> = input('Please enter something: ');
- The simplest way to do output is either:
  - omit the semicolon, or
  - use the disp( ) function:

    disp(a^2+b^2)

    disp('Sorry, your input was invalid.')
  - disp() prints a carriage return (new line) at the end
  - Print multiple items of the same type by enclosing them in square brackets:

    disp([a^2 + b^2, 2*pi])

Strings

- Almost all programming languages have some kind of string data type
  - A string is a finite sequence of characters
  - Strings are useful for interacting with the user
- In MATLAB, strings are represented as one-dimensional arrays of characters
- String constants are enclosed in single quotes

  'this is a string constant' '123456'
- Variables can have values that are strings
  - E.g., prompt = 'Please enter a number between 1 and 10'

Outputting Strings

- When MATLAB produces output, it usually automatically converts numbers to printable format (strings)
- Sometimes you must use the num2str( ) built-in function to convert a number into a string

  You need to do this to print both numbers and strings with one call to disp( ):

  ```
  disp(['the answer is: '  num2str(result)]);
  ```

Inputting Strings

- Note that input( ) evaluates the string read from the keyboard!
  - If you enter 3*2, it returns 6, *not* '3*2'
- To get a string from the keyboard, give input( ) a second argument:

```
inputstring = input('Please enter your first name: ', 's');
```

(see help for input)

Operations on Strings

- Concatenation: strcat(str1,str2) returns a single string that consists of str1 followed by str2
- Comparing strings
  - Don't use "==" to compare strings
  - Use strcmp(str1, str2) instead
  - It returns true if they are identical, false otherwise

Conditional Commands (If-statements)

- if <boolean expression>
  <command>
  end
  - Executes <command> if <boolean expression> evaluates to true (nonzero), otherwise does nothing

Other forms of if-statements

```
if  <boolexp>
     <command1>
else
     <command2>
end
```

Meaning:

if <boolexp> evaluates to true (nonzero):
execute <command1>

otherwise, (i.e., <boolexp> evaluates to false (zero)):
execute <command2>

Note well: no boolean expression after else! (Why?)

Nested if-statements

Sometimes you need to test a bunch of conditions:

```
if  score >= 90
     grade = 'A';
else
     if score >= 80
          grade = 'B';
     else
          if score >= 70
               grade = 'C';
          else
               grade = 'E';
          end
     end
end
```

Using if-statements

The "elseif" form of if-statement just makes this cleaner:

```
if  score >= 90
    grade = 'A';
elseif score >= 80
    grade = 'B';
elseif score >= 70
    grade = 'C';
else
    grade = 'E';
end
```

- Only one "end" is required
- Less indentation

Example

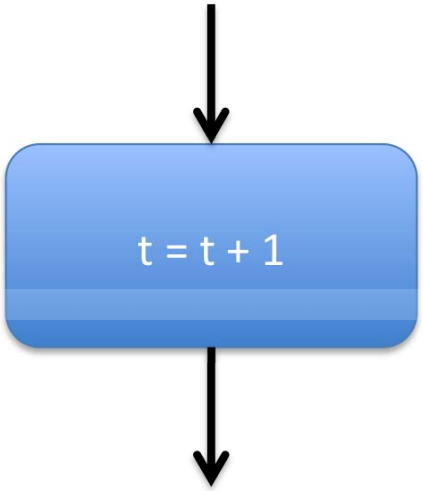
Write a script to compute the square or cube of a given number. The script should use the input() function to ask the user "square or cube?" and act according to the value input.

How to approach this?

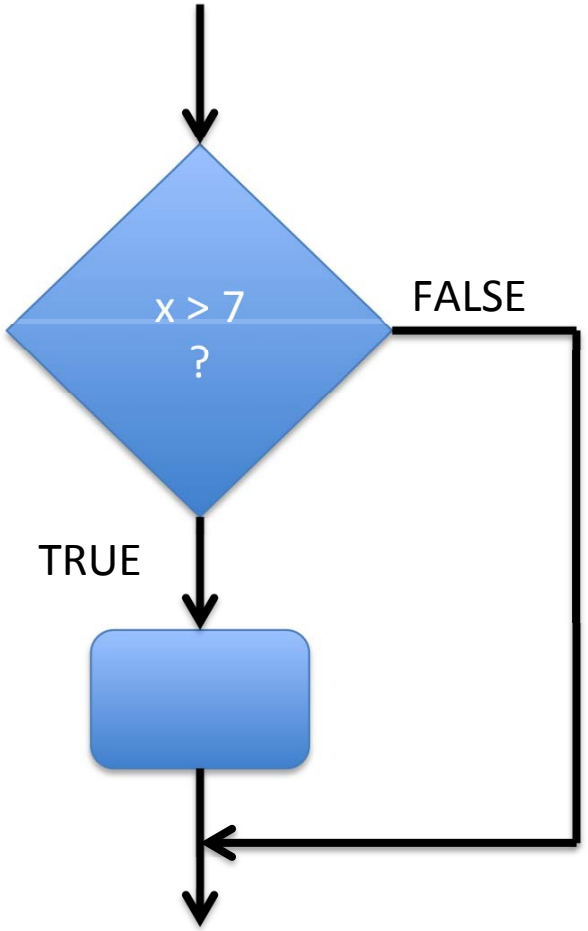
Flowcharts

- Flowcharts are a graphical way to describe computations
- They show the sequence of steps carried out by an script
- Useful for thinking about conditional statements

Flowcharts



Assignment
(or other command)



5. Homework Hints

- Use COUNT(), not COUNTIF()
  - AVERAGE() may also be OK