CS99 **Tuesday November 19, 2002**
Prelim 2 **7:30 PM – 9:00 PM**

_____ _____
(Print last name, first name, middle initial/name) (Student ID)

Statement of integrity: I did not, and will not, break the rules of academic integrity on this exam:

_____
(Signature)

Instructions:

- Skim the entire exam before starting any of the problems.
- Read each problem *completely* before starting it!
- Do not use calculators, reference sheets, or any other material. This test is closed book.
- Solve each problem using MATLAB, except where indicated.
- Use only specified code in each problem.
- Write your solutions directly on the test using blue/black pen or pencil. Clearly indicate which problem that you are solving. You may write on the back of each sheet. If you need scrap paper, ask a proctor.
- Provide only *one* statement, expression, value, or comment per blank!
- Do *not* alter, add, or remove any code that surrounds the blanks and boxes.
- Do *not* supply multiple answers. If you do so, we will grade only one that we will choose.
- Show all work, especially algorithms. Better that you explain how you would solve a problem than to leave it blank.
- You may write additional functions or use pre-defined MATLAB functions/constants wherever you see fit.
- Follow good style! When possible, keep solutions general, avoid redundant code, use descriptive variables, use named constants, indent substructures, avoid breaking out of loops, and maintain other tenets of programming philosophy.
- Comment each control structure, major variable and function (if used), *briefly.*
- Do not dwell on a problem if you get stuck. Do the other problems first!
- Raise your hand if have any questions.

Points:

1. _____ (25 points)

2. _____ (45 points)

3. _____ (15 points)

4. _____ (15 points)

Total: _____/(100 points)

Useful functions

```
function value = randInt(min,max)
% RANDINT Generate a random integer between MIN and MAX
% Ensure that the input it legal:
  if ...
    ~isreal(min)           | ~isreal(max)           | ...  % no complex
    isinf(min)             | isinf(max)             | ...  % no infinite
    isnan(min)             | isnan(max)             | ...  % no NaNs
    ~isnumeric(min)        | ~isnumeric(max)        | ...  % no non-numerical values
    min > realmax          | max > realmax          | ...  % lower bound of integer
    min < -realmax         | max < -realmax         | ...  % upper bound of integer
    floor(min)~=ceil(min) | floor(max)~=ceil(max)  | ...  % no decimal/fraction vals
    min > max                                             % no flipflop of max/min

     error('You are trying to find an illegal random number!');

  end
  value = floor(rand*(max-min+1)) + floor(min);
```

```
function value = readInt(min,max,prompt)
% READINT Get a user-input integer
% readInt is cleaner version of MATLAB's INPUT function.
% You can essentially bang on the keyboard without causing a problem.

% Ensure that the prompt is a string:
  if ~ischar(prompt)
    error('Your prompt should be a string!');
  end

% Get initial input from the user:
  value = str2double(input(prompt,'s'));

% Ensure that the input it legal:
  while ~isreal(value)            | ...  % no complex numbers
        isinf(value)              | ...  % no infinite values
        isnan(value)              | ...  % no NaNs
        ~isnumeric(value)         | ...  % no non-numerical values
        floor(value)~=ceil(value) | ...  % no decimal/fraction values
        value > max               | ...  % lower bound of integer
        value < min                    % upper bound of integer

    % Reprompt the user and get revised input:
      disp('That value is not legal! Please re-enter: ');
      value = str2double(input(prompt,'s'));
  end
```

Problem 1 [25 points] *Loops, Simulation*

Background: Suppose that you have a bag that holds 10 marbles with two different colors. You will write a program that plays a game in which the user tries to extract the exact number of each color without knowing the amounts.

Algorithm: First the bag is filled with a random number of white marbles between 0 and 10, inclusive. The rest of the space is then filled with black marbles. The program then prompts the user to pick a color to extract. If the bag still contains that color, the program extracts the marble and updates the remaining- and taken-marble counts. If the user picks a color marble that the bag no longer contains, the game ends. The game will also end if the user has luckily exhausted both colors. Otherwise, the user must choose the next marble to extract. After the game ends, the program reports the total number of marbles that were extracted.

Requirements: For full credit, you must use **`randInt`** and **`readInt`** functions!

Example session:

```
>> Problem1
Try to take a marble [0 (W), 1 (B)]: 1
Try to take a marble [0 (W), 1 (B)]: 0
Try to take a marble [0 (W), 1 (B)]: 1
Try to take a marble [0 (W), 1 (B)]: 1
Sorry! Out of black marbles!
Total marbles taken: 3
```

Problem 2 [45 points] *Nested Loops, Character Graphics*

Task: Write a program that generates a randomly-sized *nested diamond*. For example, **`Problem2`** might generate a nested diamond of size 3, as shown below:

```
>> Problem2

   /\
  //\\
 ///\\\
 \\\//
  \\//
   \/
```

Requirements: Note that the *size* refers to half the number of rows and half the number of columns. Your solution must use **`randInt`** to generate sizes between 1 and 5, inclusive. You may not use numerical arrays or strings to store more than one character at a time. So, you need to write a nested control structure without the additional help of arrays.

Hints: You might find it useful to break the program into two parts: one to draw the top half and another to draw the bottom.

Problem 3 [15 points] *Characters and Strings*

Task: Complete the function **`randChars`**, which returns an array of size (**`rows`**, **`cols`**) that is filled with randomly generated *lowercase* English letters.

Example Session:

```
>> randChars(3,2)
ans =

dagd
bwea
zqok
```

Hints: I recommend using MATLAB's **`char`** (convert to character) and **`double`** (convert to double) functions. You can actually complete the function with one statement, but you may choose not to do so.

```
function result = randChars(rows,cols)
```

Problem 4 [15 points] *Arrays*

Task: Complete the function **`core`**, which takes as input any array and returns its *inner core*. An inner core is the sub-array just inside the boundary rows and columns.

Example Session:

```
>> x=rand(4,5)
x =

    0.4513    0.0155    0.7586    0.5646    0.8022
    0.1957    0.8909    0.3807    0.7672    0.4710
    0.7871    0.7617    0.3311    0.7799    0.2028
    0.6186    0.9070    0.5041    0.4841    0.5796

>> core(x)
ans =

    0.8909    0.3807    0.7672
    0.7617    0.3311    0.7799
```

Hint: You can also write one statement to complete this function though you may choose not to do so.

```
function result = core(array)
```