CS1112 (CIS 1121) Fall 2008 Prelim 1 Sept 25 7:30–9:00pm

Q1: (10)	_________	____
Q2: (20)	_________	____
Q3: (20)	_________	____
Q4: (20)	_________	____
Q5: (30)	_________	____
Total: (100)	_________	____

Name: ____
(Legibly print last name, first name, middle name)

NetID: _________

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

______________________________________ (Signature)

Circle your lecture time: 9:05 or 11:15

Circle your section number/instructor's name:

	Tuesday	***Wednesday***
10:10		Sucheta Soundarajan
11:15		Josef Broder
12:20	Sucheta Soundarajan	Josef Broder
1:25	Sucheta Soundarajan	Vivek Maharajh
2:30	Stefan Ragnarsson	Stefan Ragnarsson
3:35	Josef Broder	

Instructions:

- This is a 90-minute, closed-book exam; no calculators are allowed.
- The exam is worth a total of 100 points, so it's about one point per minute!
- Read each problem completely, including any provided code, before starting it.
- Raise your hand if you have any questions.
- Use the backs of pages or ask for additional sheets of paper as necessary.
- Clarity, conciseness, and good programming style count for credit.
- If you supply multiple answers, we will grade only *one*.
- Use only MATLAB code. No credit for code written in other programming languages.
- Assume there will be no input errors.
- **Do not use arrays. Do not write user-defined functions.**
- Do not use `switch`, `try`, `catch`, or `break` statements.
- You may find the following MATLAB predefined functions useful:
  `abs`, `sqrt`, `rem`, `rand`, `floor`, `ceil`, `input`, `fprintf`, `disp`

  Examples: `rem`(5,2) → 1, the remainder of 5 divided by 2
  `rand` → a random real value in interval (0,1)
  `floor`(6.9), `floor`(6) → 6, rounds down to the nearest integer
  `ceil`(8.1), `ceil`(9) → 9, rounds up to the nearest integer

Question 1: (10 points)

Part (a): (3 points)

What are the final values of variables x and y?

```
x= 4;
y= 8;
x= y;
y= x;
```

x

y

Part (b): (3 points)

What are the final values of variables x and y?

```
x= 4;  y= 8;
if x<5
   x= 1;
elseif x<2 && y<10
   y= 11;
end
```

x

y

Part (c): (4 points)

Assume that variables **a**, **b**, and **c** store real values and **a**<**b**. Write an expression that evaluates to *true* if **c** is in the open interval (**a**,**b**) or the interval (**a**,**b**) is not bigger than 0.1.

Question 2: (20 points)

Part (a): (10 points)

For each fragment below, write a **`while`**-loop to produce the same printed output as the given **`for`**-loop. Only the output needs to be the same—any intermediate values need not be the same.

```
for k= 100:-2:5
   disp(k)
end
```

```
for k= 5:2:100
   disp(k)
   k= 200;
end
```

Part (b): (10 points)

Let ***x*** and ***n*** be variables that store positive integer values. Without using the `^` operator or the equivalent built-in function `power`, write a fragment to print the values x, x^2, x^3, …, x^n. Any print format is acceptable.

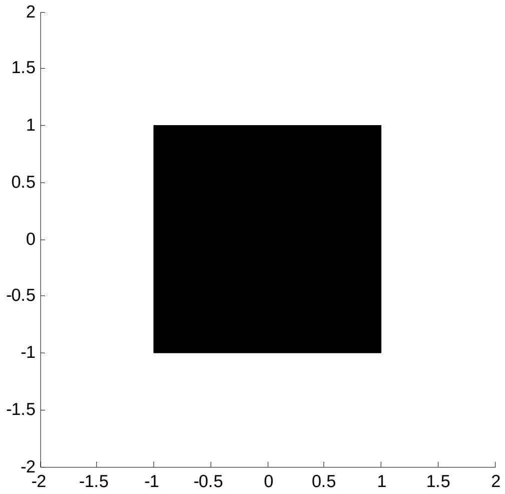
Question 3: (20 points)

The script below produces the figure shown on the right. Add code to the script to add a disk centered at a randomly generated position (x,y). x and y are uniformly random in the range of -1.5 to 1.5. The disk has radius 0.4 (so it is inside the axis limits). The disk should be red if it lies completely inside the black square, blue if it lies completely outside the black square, or magenta if it crosses any edge of the black square. You may assume that the disk is never tangent to any edge.

Assume the availability of function **DrawDisk**:

DrawDisk(-1,0,.4,’m’)

draws a magenta disk with radius 0.4 centered at (-1,0).



```
close all;  figure;  axis equal;  axis ([-2 2 -2 2]);  hold on

DrawRect(-1,-1,2,2,'k')
r= 0.4;  % radius of disk to be drawn

% (x,y) is the position of the center of the disk to be drawn.
% Generate two uniformly random values in (-1.5,1.5) for x and y.
```

```
hold off
```

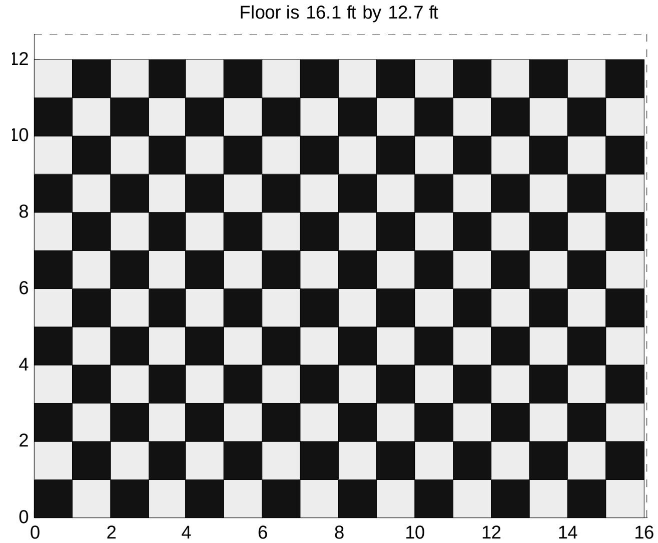
Question 4: (20 points)

Complete the script below to simulate a game in which a person is given two sticks and must select a third from a random bag of sticks until he or she has three sticks that can form a triangle. The lengths of the first two sticks, **a** and **b**, are fixed throughout the game. The sticks in the "random bag" have lengths that are uniformly random in (0,1). The game ends when the player can form a triangle with three sticks or after the player has drawn 50 sticks from the bag, whichever happens first. Display the number of sticks the player has drawn from the random bag by the end of the game.

```
% The two given sticks have lengths a and b.  a and b do not change.
a= rand(1);
b= rand(1);

% Randomly generate c, the length of the third stick, in (0,1).  The
% simulation ends when it is possible to form a triangle with sticks a,b,c
% or after 50 attempts, whichever happens first.
```

Question 5: (30 points)



Complete the script to draw the tiled floor pattern shown on the right using the **DrawRect** function. Each tile is one foot by one foot and only whole tiles are used. The bottom left corner tile is dark. The floor dimensions in the diagram on the right are only an example run of your script—the actual floor dimensions are input by the user. You may assume that the user enters positive values for the dimensions. If the floor is <1 ft^2 then no tile should be drawn.

Assume the availability of function **DrawRect**

```
DrawRect(5,0,1,2,'b')
```

draws a blue rectangle that has width 1 and height 2 with its lower left corner at (5,0). You may use any dark and light color tile combination, e.g., blue ('b') and yellow ('y') tiles.

```
close all;  figure;  axis equal;  hold on

% User inputs floor dimensions
right= input('Enter a positive value: ');
top= input('Enter a positive value: ');

% Draw bounding box from (0,0) to (right,top) to represent floor
plot([0 right],[top top],'k:')
plot([right right],[0 top], 'k:')
title(sprintf('Floor is %.1f ft by %.1f ft', right, top))

% Draw the tiles in the given pattern.  Bottom left corner has a dark tile.
```

```
hold off
```