Final Review Fall09

These questions are designed to help you think about course material, not to show you actual exam questions. The exam will be about 40-70 percent multiple choice, as discussed in class. The remainder will be short answer or coding.

Research shows that the best way to study for an exam is with other people. In group study, the people who start knowing more learn more, so don't think that it is only valuable to study with people who know more than you do.

Never discuss an answer with the group until after you write it down!

When studying, always take the time to **explain** why an answer is correct. This will help you solidify your knowledge.

1. Answer any question from the first midterm review. Answer any question from the second review except questions specifically about project 1. Be sure to review such timeless favorites as writing simple loops, functions, recursive functions, and Unix commands.

Project 2

I will not ask questions about callback, buttonDown, or nested function definitions.

2. Given a description of any function from the project, show how to create test parameters and use them in a call to the function.
3. Given a description of any function from the basic project and sample input, show how to call the function and show its output (if any).
4. Show how to draw a hex and save its handle. Also show how to create a Hex using its constructor and then get it to draw itself.
5. Show how to write constructors and methods from the project.
6. Use repmat, bwlabel, and find to show their operation on simple matrices. (Remember, find is only as useful as the masks you make.) Also review intersect and union.
7. Write a function that takes one parameter. It returns the value you pass in, but if you call it without an argument it returns 21 (hint: use nargin).

Classes

8. Create a class Bear that stores a Bear's temperature and SAT scores. A Bear should be able to walk (prints "I'm walking").
9. Create a class Animal with a run method (prints "I'm running"). Alter your bear class so that it can use any methods from Animal without having to copy them into Bear.
10. Show how to create two different Bears in the interpreter, and make them walk and run.

Search and Sorting

11. What are the differences between linear and binary search?

12. Given an algorithm resembling binary search show the sequence of calls/parameters that will be generated.

13. Given a small unsorted array, explain/draw how selection sort will turn it into a sorted array (for example show the comparisons and swaps for [4 5 3 1])

14. What are the advantages of using quick sort over selection sort? Describe how quick sort works using a recursive function.

15. Think about how selection sort and quick sort work. Imagine that you had a hundred people, and you wanted to sort them by their last name. Which sort would be easier to implement if there was one leader controlling the group, but each person could act on their own. Explain/defend your answer.

16. Show how selection sort works by writnig each pass of the sort on a matrix.

17. Calculate the number of comparisons performed on a matrix of given size by binary search, selection sort, and quicksort.

18. What are the preconditions for binary search? Explain why.

19. Given search or sort code, fill in any missing code. Alternatively, if is coded incorrectly, show exactly what will result.

0.0.1 MATLAB functions/programming

20. Convert a simple function with a loop to its vectorized form.

21. Given a vector m = [1 4 6 7 3 9], give a Matlab expression that will evaluate to m with 7 removed.

22. Write a single statement that will remove all numbers less than 6 from a vector.

23. Explain the difference in memory usage between Matlab's default representation of numbers and the representation of a mask.

24. Demonstrate the use of Matlab functions **any, all**.

25.

26. Write a recursive and an iterative function to display a triangle of asterisks:

```
⋆
⋆⋆
⋆⋆⋆
⋆⋆⋆⋆
```

27. Given a recursive function, show what it displays when passed certain parameters (trace the function as on Midterm 1). For example, you could be given a function that makes a geometric shape.

28. Write a recursive function to calculate a simple numeric function, like factorial or exponent.

29. Create, and access the fields of a structure.

30. Create an array of a structures.

31. Sum all of the values of a field in an array of a structures (for example all of the credits taken by an array of student structures).

32. Write a function to compare the time the cpu spends performing two functions 100 times each. Do not plot the data, just display the times.

33. Given some data in matrices, show how to plot it using the plot function.

34. Show how to grow a Matlab matrix. In general, is this a good idea? Why or why not?

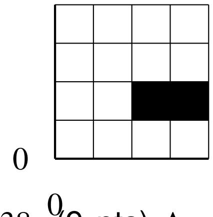
35. (3 pts) Show how to create, at the command line given, a Matlab structure "bear" with four legs.

>>____________________

36. (3 pts) Add another field to the bear above to represent that the bear has 42 teeth.

>>____________________

37. (5 pts) Suppose you had an existing figure in Matlab, and wanted to add a patch to it. **Write the single patch** command that would add the patch shown below to the figure. Assume that the lower left corner of the figure (not the patch) is (0,0) and the upper right is (4,4).



38. (9 pts) Assume you are **given** a Matlab function **"spaces"** that takes a single parameter and prints that number of spaces (and nothing else). Use this function to complete the three blanks in the following definition of a recursive function **"myDiag"** that performs as follows:

```
>> myDiag(5)
     *
    *
   *
  *
 *
```

```
%Think about what three things this function needs to do in the else case!
function [] = myDiag(n)
  if (n == 0)
    return;
  else

    ____________________________________

    ____________________________________

    ____________________________________
  end
end %myDiag
```

39. (7 pts) Explain how to scale and shift so that you can generate a random number within a specified range. Feel free to write Matlab examples to illustrate your explanation.

0.1 In the Interpreter:

40. (3 pts) Show how to extract `[5 6 7]` from the matrix m shown.
`>> m = [1 2 3 4 5 6 7 8];`

41. (3 pts) Show how to extract `[5 6; 9 10; 13 14]` from the matrix m shown.
`>> m = [1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16];`

42. In one or two lines of Matlab code, create a mask and use it to remove all the sevens from the matrix m.

43. In three or four lines of Matlab code, write Matlab code that makes and uses a mask to print 'spam' (only once) if there are any sevens in a matrix m.

44. Solve the previous problem using a for loop.

45. Write a Matlab expression to find the number of sevens in a matrix m without using a loop.

46. For a given matrix m, show the result of the expression flipud(m')