EECE 231 — Introduction to Programming Using C++ and MATLAB, Sections 3, 4, and 12
Quiz II

Nov 10, 2015

- The duration of this exam is **1 hour 40 minutes**.
- Keep in mind that you need around **10 minutes** at the end of the duration of the exam to **submit** your answers. It is your responsibility to make sure your files are correctly submitted.
- The exam consists of 3 problems for 105 points.
- Plan your time wisely. Do not spend too much time on any one problem. Read through all of them first and attack them in the order that allows you to make the most progress.
- You can use the following material available on Moodle: class presentations, sample code provided by your instructor, assignments submissions and corrections, and previous quizzes and material posted on Moodle by your instructor.
- You are **NOT** allowed to send/receive any **Moodle messages** during the exam.
- You are **NOT** allowed to use the **web**.
- You are **NOT** allowed to use **USB drives** or files previously stored on your machine.
- If you get caught violating the above rules or if you communicate with a person other than the exam proctors during the exam, you will immediately get zero and you will be referred to the appropriate disciplinary committee.
- Active cell phones and any other unauthorized electronic devices are absolutely not allowed in the exam rooms. They should be turned off and put away.
- Submit your solutions each part in a separate file as indicated in the booklet. Include your name and ID number in each file.
- Good luck!

Problem 1 (25 points). Matlab Graphics

Write a Matlab script to produce the plot in Figure 1. This plot is an upper half of a circle centered at $(0, 0)$ and having a radius of 5. A straight line connects the two ends of the half-circle.

Set the axis limits as shown in the figure, and show the grid.

Submit your solution in a file called Prob1.m including your name and ID number in the comments.

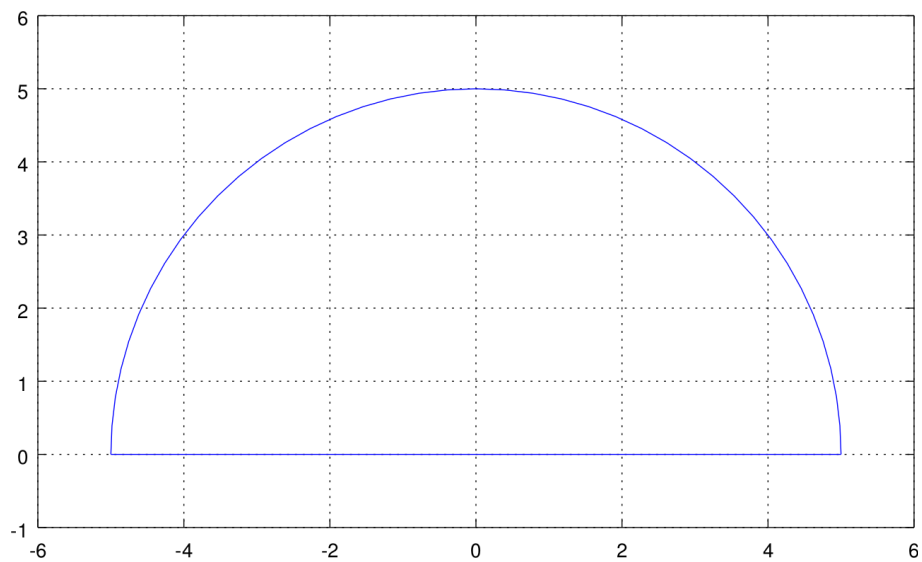

Figure 1: A half-circle with radius equal to 5. The circle is centered at $(0, 0)$. A straight line connects the two ends of the half-circle.

Problem 2 (40 points). Matrix Manipulation in Matlab

a) **(20 points)** The Matlab function *rand()* returns a matrix with random double numbers uniformly distributed on the interval (0, 1).
Write a Matlab function, call it *RandIntMat()*, which takes as arguments the size of a matrix given by M (number of rows) and N (number of columns), and two integer values a and b. The function should return a matrix of size $M \times N$ with random integers between a and b, inclusively.
For example, the following call to the function $RandIntMat(3, 4, 20, 50)$; should return a 3×4 matrix filled with random integers taking values between 20 and 50, inclusively.

```
>> A=RandIntMat(3,4,20,50);
>> A
A =
   21    41    25    38
   40    32    30    39
   44    21    32    45
```

Hint: Your *RandIntMat()* function should call the Matlab functions *rand()* and *round()*.

Submit your function in a file called RandIntMat.m including your name and ID number in the comments.

b) **(20 points)** In Matlab, the 3×5 matrix A, where

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$$

can be constructed using $A = [1, 2, 3, 4, 5; 6, 7, 8, 9, 10; 11, 12, 13, 14, 15];$.

Write a Matlab script that constructs the above A. Your script should use **one for loop only**. For simplicity, you are given the below incomplete code. You need to fill in the missing lines. Mainly, you have to find out the values of *rnum* and *cnum* by relating them to the value of the counter i.

```
A = zeros(3,5);
for i=1:15
    % missing lines
    A(rnum, cnum)=i; % rnum is the row number. cnum is the column number
end
```

Submit your code in a file called Prob2b.m including your name and ID number in the comments.

Problem 3 (40 points). Recursive Functions in C++

a) **(20 points)** Write a C++ recursive function (call it *RecPrintArray*) which takes as parameters an array of integers and its size, and prints the numbers in the array. This function should NOT use any loops (for or while loops). The displaying of the array numbers should be the result of the recursive calls of the function. Include a simple main program to test your *RecPrintArray* function with the array $A[10] = \{2, -1, 3, -5, 1, 8, 7, 6, 9, 8\}$. Your function should print the integers in A from left to right.

b) **(20 points)** Repeat Part(a), but now the recursive function (call it *RecRevPrintArray*) should print the array integers in reverse order, i.e. from last to first. Test your *RecRevPrintArray* function with the array $A[10] = \{2, -1, 3, -5, 1, 8, 7, 6, 9, 8\}$.

Note: The two functions are of the void type, and they should not make any changes to the arrays.

Hint: Consider passing a third parameter to the functions to help with the solution.

Submit both functions in a file called Prob3.cpp which should include your name and ID number in the comments. Include the main program used to test your functions.