ComS 207: Programming I
Homework 8
Out: Wed. Nov 1, 2006
Due: Fri. Nov 10, 2006 (Before the start of class. Online submission *ONLY*)

Student Name:

Recitation Section:

Some of these problems came from your textbook. The text is reproduced here for your convenience.

1. **What is the Output?**

   For each of the following four recursive methods, try to figure out what will be the output of the recursive call with the specified arguments. Try to do this by hand first as there will be something like that on the final exam.

   This code was posted on: http://www.cs.princeton.edu/introcs/23recursion/

```
public class Recursion {

    public static void mystery1(int a, int b) {
        if (a <= b) {
            int m = (a + b) / 2;
            System.out.print(m + " ");
            mystery1(a, m-1);
            mystery1(m+1, b);
        }
    }

    public static void mystery2(int n) {
        if (n > 0) {
            System.out.print(n + " ");
            mystery2(n-2);
            mystery2(n-3);
            System.out.print(n + " ");
        }
    }
    public static void mystery3(int n) {
        if (n == 0 || n == 1) return;
        mystery3(n-2);
        System.out.print(n + " ");
        mystery3(n-1);
    }

    public static String mystery4(int n) {
        if (n <= 0) return "";
        return mystery4(n-3) + n + " " + mystery4(n-2) + n + " ";
    }

    public static void main(String[] args) {
        int N = 6;
        mystery1(0, N);
        System.out.println();
        mystery2(N);
        System.out.println();
        mystery3(N);
        System.out.println();
        System.out.println(mystery4(N));

    }

}
```

Your answer:

mystery1:

mystery2:

mystery3:

mystery4:

2. **Programming Projects**

Choose ***three of the following five*** programming projects and implement them. Your grade will NOT depend on which ones you choose. Just pick the ones that you like.

(a) [Exercise 11.5 (p. 609)]

Modify the method that calculates the sum of the integers between 1 and N shown in this chapter. Have the new version match the following recursive definition: The sum of 1 to N is the sum of 1 to (N/2) plus the sum of (N/2 + 1) to N. Trace your solution using an N of 7.

(b) [Exercise 11.7 (p. 609)]

Write a recursive method to reverse a string. Explain why you would not normally use recursion to solve this problem.

(c) [Programming project 11.2 (p. 610)]

Design and implement a program that implements Euclid's algorithm for finding the greatest common divisor of two positive integers. The greatest common divisor is the largest integer that divides both values without producing a remainder. An iterative version of this method was part of the RationalNumber class presented in Chapter 6. In a class called DivisorCalc, define a static method called gcd that accepts two integers, num1 and num2. Create a driver to test your implementation. The recursive algorithm is defined as follows:

```
> gcd (num1, num2) is num2 if num2 <= num1 and num2 divides num1

> gcd (num1, num2) is gcd (num2, num1) if num1 < num2

> gcd (num1, num2) is gcd (num2, num1 % num2) otherwise
```

(d) [Programming project 11.7 (p 610)]

In the language of an alien race, all words take the form of Blurbs. A Blurb is a Whoozit followed by one or more Whatzits. A Whoozit is the character 'x' followed by zero or more 'y's. A Whatzit is a 'q' followed by either a 'z' or a 'd', followed by a Whoozit. Design and implement a recursive program that generates random Blurbs in this alien language.

(e) **[Programming project 11.9 (p 611)]**

Design and implement a recursive program to determine and print the Nth line of Pascal's Triangle, as shown below. Each interior value is the sum of the two values above it.

Hint: use an array to store the values on each line.

```
                   1
                 1   1
               1   2   1
            1   3   3    1
          1   4   6   4   1
        1   5  10    10   5   1
      1   6  15  20    15   6   1
    1   7  21  35    35  21   7   1
  1   8  28  56  70  56  28    8   1
1   9  36  84 126 126  84  36    9   1
```

3. *** For Advanced (or Bored) Students Only!**

   [Programming project 11.6 (p 610)]

   Design and implement a recursive program that solves the Non-Attacking Queens problem. That is, write a program to determine how eight queens can be positioned on an eight-by-eight chessboard so that none of them are in the same row, column, or diagonal as any other queen. There are no other chess pieces on the board.

   Hint: For 8-queens there are a total of 92 solutions. If we exclude symmetry, there are 12 solutions.

4. **What to Submit**

   ONLINE ELECTRONIC SUBMISSION ONLY USING WebCT!!!

   PLEASE, DO NOT SUBMIT PRINTOUTS.

   For part 1 submit a text file with your answers.

   DEADLINE: Before the start of class on Nov 10, 2006.

   IMPORTANT: Once again, no late homeworks will be accepted.

   That's it. Good Luck!