COS126 Exam 1 Mini-Test

1. **Short Answer**

   1. Write the value of `(double) ( 22 / 7 )`.

   2. Write this number using Java's scientific notation, (without using `Math.pow`):
      $6.022 \cdot 10^{23}$

   3. True or False. Any `for` loop can be converted into an equivalent `while` loop.

   4. True or False. Any recursive method can be re-written as a non-recursive method using loops.

   5. True or False. The following condition will compile in Java.
      `(a < b < c)`

   6. You have a program called `Recipe.java` which reads from standard input and writes to standard output. You have compiled it. The command-line to run it so it reads keyboard input and writes to the terminal screen is: `java Recipe`.
      Write the command-line to run it so it reads input redirected from a file named cookbook.txt.

      Write the command-line to run it so it reads input from cookbook.txt and writes to an output file named meal.txt

      Write the command-line to run it so it reads keyboard input and pipes the output to another compiled program named `HungryThing.java`.

2. **Doubles, StdIn, Analysis of Algorithms**

The following takes two command-line arguments x, y; reads from standard input a sequence of point coordinates (xi, yi), and prints what?

```
public class Mystery {
    public static void main(String[] args) {
        double x = Double.parseDouble(args[0]);
        double y = Double.parseDouble(args[1]);

        double bestx = Double.NaN;
        double besty = Double.NaN;
        double bestDist2 = Double.POSITIVE_INFINITY;

        while (!StdIn.isEmpty()) {
            double xi = StdIn.readDouble();
            double yi = StdIn.readDouble();
            double dist2 = (x - xi) * (x - xi) + (y - yi) * (y - yi);
            if (dist2 < bestDist2) {
                bestx = xi;
                besty = yi;
                bestDist2 = dist2;
            }
        }

        // output
        StdOut.printf("Closest point = (%f, %f)\n", bestx, besty);
    }
}
```

Suppose we run the Mystery program as follows:

```
 % java Mystery 1.0 5.0
1.0 3.0
5.0 3.0
9.0 6.0
2.0 6.0
5.0 6.0
<Ctrl-d>
```

a) Fill in the trace table:

x	y	bestx	besty	bestDist2	xi	yi	dist2

2. Continued

   b) What does the program print?

   c) What kind of input would cause `NaN, NaN` to print out?

   d) In general, what does this program do?

   e) Suppose we read in N points. How many comparisons of `dist2` and `bestDist2` will the program make?

3. **Recursion, Debugging (from Spring04, Exam 1, Question 4)**

There's a bug in the following recursive program. You need to find it and fix it.

```
public class Series{
    public static int func(int j){
       if (j==1) return 1;
       return 2*func(j-1)+5*func(j-2);
    }

    public static void main(String[] args) {
        int N=Integer.parseInt(args[0]);
        if (N<0) {
            System.out.println(``invalid argument'');
            return;
        }
        System.out.println(func(N));
    }
}
```

a. Draw the recursion tree for func(3). You only need to draw the tree up to 3 levels, which means the height of the recursion tree should be no greater than 3.

b. From the recursion tree in (a), do you see a problem with the program? Explain what is the problem.

4. **Performance.** The following table gives approximate running times for a program with N inputs for various values of N.

N	time
1000	5 seconds
2000	20 seconds
5000	2 minutes
10000	8 minutes

Which of the following best describes the likely running time of this program for $N = 100,000$?

V. A few minutes

W. A few hours

X. Half a day

Z. A few days