COS 126 Fall 2006 Exam 1 Solutions

1. Short Answer

(a) -20
(b) 3.0
(c) -2^31 = -2147483648
(d) Infinity
(e) 6.022e23

2. Arrays, Conditional, Loops, and Bugs

1) MainProgram instead of main
2) did not allocate elements of array a
3) keepOn should be initialized to true
4) in the for loop inside the while loop, go to N-1 instead of N
5) temp should be a double, not an int
6) order is wrong in the swap statements
7) missing semicolon after increment of iters

3. Functions

My score will be 9.5 points.
This question is worth 10 points.
Of all the 9.5 points, I will receive 10 of them.

4. Loops, Strings, Conditionals

(a)

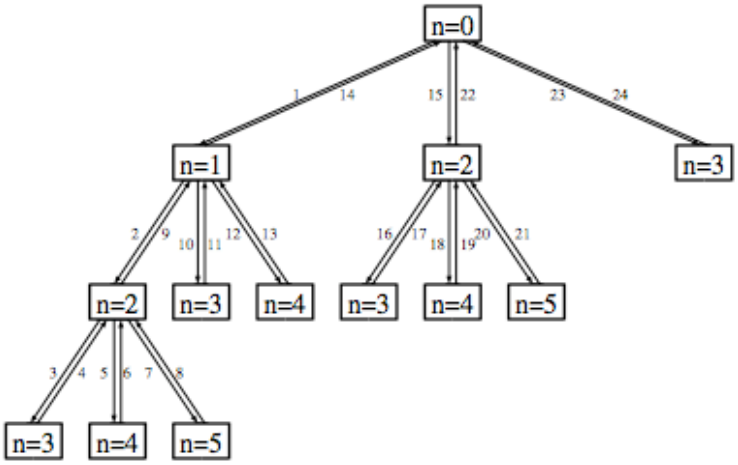
31+121+13
1/1/1/1/
2/+/3/
1/
StringIndexOutOfBoundsException: (or the equivalent)

(b) 2^N (because the string length doubles at each iteration)

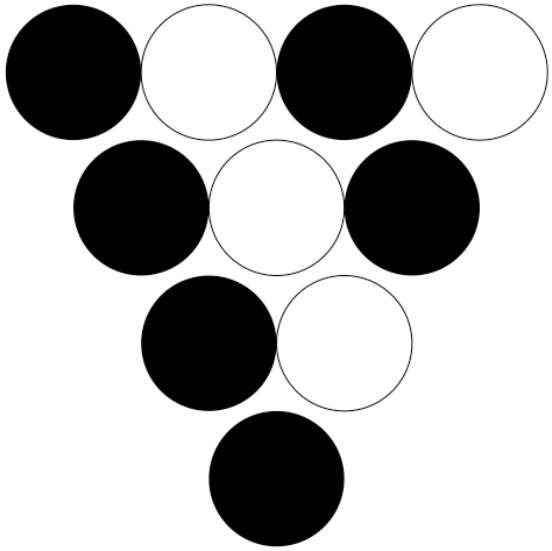
5. I/O

```
public static void findClosest(double X) {
double closest = StdIn.readDouble();
       while (!StdIn.isEmpty()) {
           double next = StdIn.readDouble();
           if (Math.abs(X - next) < Math.abs(X - closest))
               closest = next;
       }
       System.out.println(closest);
   }
```

6. Recursion



7.



8. Recursion

(a) Ack(0, 0) = 1
Ack(0, 3) = 4
Ack(0, 6) = 7
Since the recurrence states that "if m == 0, return n +1", the function computes "n + 1" when m is fixed at 0.

(b) Ack(1, 0) =
Ack(0, 1) = 2
Using the fact that Ack(0, n) = n + 1.

```
Ack(1, 3)                          =
Ack(0, Ack(1, 2))                  =
Ack(0, Ack(0, Ack(1, 1)))          =
Ack(0, Ack(0, Ack(0, Ack(1, 0)))) =
 1   +  1   +  1   +  2           = 5
```

Using the facts that Ack(1, 0) = 2 and that Ack(0, n) computes "n + 1"

```
Ack(1, 6)                          =
Ack(0, Ack(1, 5))                  =
Ack(0, Ack(0, Ack(1, 4)))          =
Ack(0, Ack(0, Ack(0, Ack(1, 3)))) =
 1   +  1   +  1   +  5           = 8
```

Using the facts that Ack(1, 3) = 5 and that Ack(0, n)

computes "n + 1"

In general, Ack(1, n) = Ack(0, Ack(0, ... Ack(1, 0))), with "n" Ack(0, ...) and one Ack(1, 0). Since Ack(1, 0) is 2, and the other "n" Acks add 1 onto the previous result, we get n + 2.

(c) Ack(2, 0) =
Ack(1, 1) = 3
Using the fact that Ack(1, n) = n + 2.

```
Ack(2, 3)                          =
Ack(1, Ack(2, 2))                  =
Ack(1, Ack(1, Ack(2, 1)))          =
Ack(1, Ack(1, Ack(1, Ack(2, 0)))) =
 2   +  2   +  2   +  3           = 9
```

Using the facts that Ack(2, 0) is 3 and that Ack(1, n) computes "n + 2".

```
Ack(2, 6)                          =
Ack(1, Ack(2, 5))                  =
Ack(1, Ack(1, Ack(2, 4)))          =
Ack(1, Ack(1, Ack(1, Ack(2, 3)))) =
 2   +  2   +  2   +  9           = 15
```

Using the fact that Ack(2, 3) is 9 and that Ack(1, n) computes "n + 2".

In general, Ack(2, n) = Ack(1, Ack(1, ... Ack(2, 0))), with "n" Ack(1, ...) and one Ack(2, 0). Since Ack(2, 0) is 3 and the other "n" Acks add 2 onto the previous result, we get 2n + 3.

9. TOY

All of these are no-ops (most use register 0): 10XY, 1XX0, 1X0X, 20XY, 2XX0, 30XY, 3XXX, 40XY, 4XX0, 4X0X, 50XY, 5XX0, 60XY, 6XX0, 70XY, 80XY, A0XY, D0XY (where X and Y are arbitrary nybbles)