COS 126	General Computer Science	Spring 2011

Exam 1 Solutions

1. **Number systems.**

   (a) 173_{10}

   (b) 31. The last integer printed is 2^{30} because $2^{30} + 2^{30}$ overflows an `int` and results in -2^{31}.

2. **Java basics.**

   (a)

Java expression	*type*	*value*
`1 + 2.0 * 3 + 4.0`	`double`	`11.0`
`(-1 / -1) / 0`	runtime error	–
`(-1.0 / -1.0) / 0.0`	`double`	positive infinity
`Math.sqrt(-2.0)`	`double`	`NaN`
`1 + "+" + 2.0 + "3"`	`String`	`"1+2.03"`
`(double) (10 / 4)`	`double`	`2.0`
`(1.0 <= 2.0 <= 3.0)`	compile-time error	–

   (b) i, iii, iv, v

3. **Loops, conditionals, and arrays.**

   (a) `0.5`

   (b) The smallest difference (in absolute value) between any two values in `a[]`; infinity if no such value.

4. **Input and output.**

   The programs prints out the first number, and then the average (integer division) of each number and its predecessor.

   (a) `2 3 5 7 9 11 10 5`.

   (b) `2 2 4 6 8 10 10 7`.

5. **Functions.**

(a)
```
public static int gcd(int p, int q, int r) {
    return gcd(gcd(p, q), r);
}
```

(b)
```
public static int gcd(int[] values)
```

6. **Recursive graphics.**

(a) 1 4 5 2 6 3 (the unique ordering)

(b) I and III only

- The `drawShadedSquare()` must appear after the lower left and upper right calls because the order 5 square obscures the order 4 recursive patterns in the lower left and upper right; it must appear before the upper left and lower right calls because the order 4 squares are obscured by the order 3 patterns in the upper left and lower right.
- The lower right call must appear before the upper left call because the order 4 pattern is started in the lower right but the order 4 pattern has not yet begun in the upper left.
- The upper right call must appear before the lower left call because of the unfinished order 3 pattern (that is the lower left part of the lower right order 4 pattern).

7. **TOY.**

(a) `0002` and `0008`

(b) `000B` and `0010`

(c) `a = a % b;`