COS 126	General Computer Science	Spring 2009

Exam 1 Solutions

1. **Number systems.**

   (a) $44_{10} = 4 + 8 + 32$.

   (b) -20_{10}. Flipping the bits and adding one yields 010100_2 (010011 + 1), which is 20_{10} (4 + 16). Thus, 101100_2 is -20_{10}.

   (c) $FFB3_{16}$. The binary representation of 77_{10} is $0000\,0000\,0100\,1101_2$. (64 + 8 + 4 + 1). Flipping the bits and adding one yields -77_{10} in binary, which is $1111\,1111\,1011\,0011_2$. We convert to hexadecimal, 4 bits at a time: $FFB3_{16}$.

   (d) Equal for all values. Recall that ^ is *exclusive-or* in Java and TOY, not exponentiation. The exclusive-or operator is associative.

2. **Java basics.**

   (a) II and III only.

   (b) It is possible to reach the end of the function without returning any value.

   (c) A set of values and operations on those values. See 1.2 in the textbook or lecture slides.

   (d) I and III only. II is a property of both command-line inputs and standard input.

   (e) All are true except for iv (a Java function can return only one value). See pp. 188–193 of the textbook.

3. **Arrays, loops, functions, and debugging.**

```
public static double[] times(double[][] a, double[] x) {
   int N = x.length;
   double[] y = new double[N];
   for (int i = 0; i < N; i++) {
       double sum = 0.0;
       for (int j = 0; j < N; j++) {
           sum += a[i][j] * x[j];
       }
       y[i] = sum;
   }
   return y;
}
```

4. **TOY.**

(a) 0. The program repeatedly subtracts `R[1]` (which is always `11`) from `R[2]` (initially `55`) until `R[2]` reaches 0.

(b) Does not halt. The load statement in line `11` sets `R[2]` to the value in memory location `10` (which is `7111`). The statement in line `13` jumps to line `11`, which resets `R[2]` to `7111` each time.

(c) `A1F0`. The program sets `R[2]` to the contents of the memory location pointed to by `R[1]` (memory location `11`). Thus, `R[2]` is initialized to `A201`. The subtract statement in line `12` It subtracts `0011` from `R[2]` to get `A1F0`. Since `R[2]` is negative, the jump statement in line `13` does not change the program counter.

5. **Recursive graphics.**

(b)