COS 126 **General Computer Science** **Spring 2007**

Exam 1 Solutions

1. **Number systems.**

   (a) `57` $= 1 + 8 + 16 + 32$.

   (b) `7D8` $= 7 \times 16^2 + 13 \times 16 + 8$.

   (c) `10110011` To compute -77, first we convert 77 to binary: 01001101. Then, we flip the bits and add 1: 10110010 + 1 = 10110011.

   (d) -2^{31}. The reason is because $+2^{31}$ is not representable as a 32-bit two's complement integer. As a result, `Math.abs(-2147483648)` equals -2147483648.

2. **Nested loops and conditionals.**

   Here are two (equivalent) correct solutions:

   - Swap the order of the two loops.
   - Replace both occurrences of $i - j$ with $j - i$.

3. **Java loops and functions.**

   (a) 4. The corresponding values of `i` are: 0, 1, 8, 729.

   (b) 100. In Java, when you pass an integer to a function, you are passing a copy of that value. If the function changes its copy of that value, this will not change the value of the original variable.

4. **Debugging and arrays.**

   1. The name of the file must match the name of the class.
   2. The argument to `main` should be `String[] args` instead of `void`.
   3. The variable `N` is not declared; it should be of type `int`.
   4. The variable `valcnt` is uninitialized; it should be initialized to `0`.
   5. The variable `found` is declared to be of type `boolean`; it should be of type `boolean[]`.
   6. The loop-continuation condition should be `(valcnt < N)` instead of `(cardcnt > 0)`.
   7. The type of `val` should be `int` instead of `double` since it is used as an index into an array.
   8. The boolean expression in the `if` statement should be `(!found)` or `(found == false)`.
   9. There are missing curly braces around the `if` statement. Without them, `valcnt` gets incremented each time through the `while` loop.
   10. The array access `found[i]` should be `found[val]`. The variable `i` is not defined.

5. **Functions.**

(a)
```
public static boolean areTriangular(int x, int y, int z) {
   if (x >= y + z) return false;
   if (y >= x + z) return false;
   if (z >= x + y) return false;
   return true;
}
```

(b)
- To make code easier to read.
- To make code easier to debug.
- To make code easier to maintain.
- To make code easier to reuse.
- To make code easier to divide up among several programmers.
- To use recursion.

6. **Standard input, standard output, and redirection.**

(a)

```
public class Assignments {
    public static void main(String[] args) {
        int N = StdIn.readInt();

        // as long as there are more students
        while (!StdIn.isEmpty()) {

            // read in student's name
            String name = StdIn.readString();

            // read in the N exam scores and sum them up
            int sum = 0;
            for (int i = 0; i < N; i++) {
               int score = StdIn.readInt();
               sum = sum + score;
            }

            // compute and print out the average
            double average = (double) sum / N;
            System.out.println(name + " " + average);
        }

    }
}
```

(b) `java Assignments < data.txt > output.txt`

7. **Recursive graphics.**

(a) `2 1 4 3`
Switching 3 and 4 makes the overlap go the other way. Switching 1 and 2 means that the order 0 figure draws a shaded circle instead of returning; this effectively increases the depth of the recursion by 1.

(b) It goes into an infinite loop since there is no base case. On most systems, this will result in a `StackOverflowError`, when Java runs out of memory to store the history of function calls.

8. **TOY I.**

(a) 2^{12}. The exact number of bits is: 255×16 (255 16-bit main memory locations, excluding `FF`) $+ 15 \times 16$ (15 16-bit registers, excluding `R0`) $+ 1 \times 8$ (program counter) $= 4328$, which is closest to $2^{12} = 4096$.

(b) d

9. **TOY II.**

(a)
```
13: 2331     R[3] <- R[3] - R[1] = R[3] - 1
15: C418     if (R[4] == 0) pc <- 18
16: D413     if (R[4] >  0) pc <- 13
18: 92FF     write R[2] to standard output
```

(b) `00AA`
The program accesses memory locations `FE`, `FD`, `FC` and so on until it reads a non-positive value. If that value is `0000`, it jumps to line `18` and writes the initial value of `R2` to standard output (`00AA`). Otherwise, it changes `R2` and prints that value out (`00BB`).

(c) There are many correct answers. The easiest is to put a negative value in memory location `FE`.

```
FA:  0000
FB:  0000
FC:  0000
FD:  0000
FE:  9999
```