MIT 6.035 Spring 2011 Quiz 2

Full Name: ______________________________

MIT ID: ______________________________

Athena ID: ______________________________

Question:	1	2	3	4	5	6	Total
Points:	10	20	20	15	15	20	100
Score:							

1. We want to optimize the following program snippet written in the Decaf language by eliminating common subexpressions:

```
i = callout("get_int_035");
j = i + 1;
k = i;
l = k + 1;
```

where the `get_int_035` function reads an integer from standard input and returns it.

(a) (5 points) What does the optimized code look like when we use *value numbering* to find common subexpressions?

(b) (5 points) What does the optimized code look like when we use *available expression* analysis to find common subexpressions?

2. We want to compute reaching definitions for the following program:

```
L1:     i = m - 1;
L2:     j = n;
L3:     a = x;
        do {
L4:             i = i + 1;
L5:             j = j - 1;
                if (i < j) {
L6:                     a = y;
                } else {
L7:                     i = z;
                }
        } while(j > m);
```

(a) (5 points) Draw the control flow graph of this program.

(b) (5 points) Compute GEN[n] and KILL[n] for each basic block n.

(c) (5 points) Set up data-flow equations (IN[n] $= \cdots$ and OUT[n] $= \cdots$) for each basic block n.

(d) (5 points) Find out the solution of the data-flow equations.

3. Design a data-flow analysis that determines which expressions are very busy at each program point. An expression is very busy at a program point p if, along every path from p, the expression is always used before a redefinition of any of the variables occurring in it.

   (a) (5 points) Is the data-flow analysis a forward analysis? Or a backward analysis?

   (b) (5 points) Draw the Hasse diagram of the lattice used in the analysis, assuming that there are just three expressions (e_1, e_2 and e_3) in the target program.

   (c) (5 points) What is the confluence operator?

   (d) (5 points) To what value should IN[entry] or OUT[exit] but not both, depending on your answer to (a), be initialized at the beginning of the worklist algorithm? Choose one between IN[entry] and OUT[exit].

4. (a) (10 points) Prove that the *greater than or equal* relation ($\geq$) is a partial order on the set of integers with both positive and negative infinity ($\mathbb{Z} \cup \{-\infty, \infty\}$).

(b) (5 points) Draw the Hasse diagram for it, and mark its greatest element ($\top$) and least element ($\perp$).

5. If your proof is not correct, you will not get a score even when your yes/no answer is correct.
   (a) (8 points) If a lattice has the greatest or least element, is it always unique? Prove or disprove it.

   (b) (7 points) Does a complete lattice always have both greatest and least elements? Prove or disprove it.

6. Alice designed a mysterious data-flow analysis on programs written in the following language:

$$\begin{array}{rcl} \mathrm{S} & \rightarrow & id := \mathrm{E} \mid \texttt{if}\ (\mathrm{E} < \mathrm{E})\ \texttt{then}\ \mathrm{S}\ \texttt{else}\ \mathrm{S} \\ \mathrm{E} & \rightarrow & id + id \mid c \end{array}$$

where id and c denote a variable and a non-negative integral constant. It is known that she modeled a program state at each program point as a function that maps each variable to its value. For example, the $[x \mapsto 1, y \mapsto 2]$ program state means that x and y have 1 and 2 at the program point, respectively. Also, her abstraction function is as follows:

$$\mathrm{AF}([id_1 \mapsto v_1, id_2 \mapsto v_2, \cdots, id_n \mapsto v_n]) = [id_1 \mapsto (v_1 \,\%\, 3), id_2 \mapsto (v_2 \,\%\, 3), \cdots, id_n \mapsto (v_n \,\%\, 3)]$$

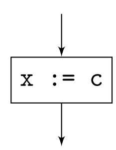
where % is the remainder operator. Let's restore her data-flow analysis from the abstract function.

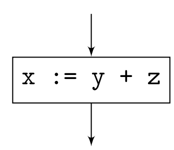
(a) (5 points) Draw the Hasse diagram for the base lattice for her data-flow analysis. Note that the actual lattice is defined using elements of the base lattice as follows:

$$[id_1 \mapsto \widehat{v}_1, id_2 \mapsto \widehat{v}_2, \cdots, id_n \mapsto \widehat{v}_n]$$

where $\widehat{v}_1$, $\widehat{v}_2$ and $\widehat{v}_n$ are elements of the base lattice.

(b) (10 points) Define the transfer functions for the following basic blocks.





(c) (5 points) Will the analysis always produce the meet-over-path solution? Justify your answer by proof sketch or example. If your justification is not correct, you will not get a score even when your yes/no answer is correct.