CIS (4|5)61
Winter 2010 Midterm 2

Write your name at the bottom of each page before you begin.

1. [10 points]

Local variables can be allocated in llvm using code like this:

```
%p = alloca i32*, align 4
```

However, while we can allocate references to objects this way, we cannot generally use `alloca` to allocate the objects themselves. Why not? What *really bad thing* could happen if we did? (Try to be more specific than "the program could crash.")

2. [10 points] Cool has the type rule

$$\frac{O, M, C \vdash e_1 : \mathit{Bool} \quad O, M, C \vdash e_2 : T_2}{O, M, C \vdash \text{while } e_1 \text{ loop } e_2 \text{ pool} : \mathit{Object}} \quad \text{[Loop]}$$

Why isn't it

$$\frac{O, M, C \vdash e_1 : \mathit{Bool} \quad O, M, C \vdash e_2 : T_2}{O, M, C \vdash \text{while } e_1 \text{ loop } e_2 \text{ pool} : T_2} \quad \text{[Loop]}$$

3. [10 points] For this question, we imagine a computer with the following instruction set:

LDA r_i,x	;; The address of location x is loaded into register r
LDC r_i,k	;; $r_i := k$ (k is a constant)
LD r_i,x	;; r_i := Mem[x]
ST r_i,r_k	;; Mem[r_k] := r_i
ADD r_i,r_j,r_k	;; $r_i := r_j + r_k$
SUB r_i,r_j,r_k	;; $r_i := r_j$ - r_k
BP label	;; Branch to label if the last result was positive
BZ label	;; Branch to label if the last result was zero
B label	;; Branch to label (regardless of the condition)

Assume an unlimited set of registers, and assume that a, b, and c are integer variables that have already been allocated in the activation record. Show the register SSA code you might generate for the following (which could be Java or C or something similar with short-circuit evaluation of boolean expressions):

```
if (a > b && b > c) {
   a = a + b;
}
```

4. [10 points] A *live variables* analysis is based finding the least fixed point solution to a set of data flow equations, where the following equation is associated with each node in the control flow graph (except the exit node, which has an empty live set):

$$\text{live}(n) = \bigcup_{m \in \text{succ}(n)} \text{live}(m) - \text{kill}(n) + \text{gen}(n)$$

(where $-$ and $+$ are set difference and union, respectively, and $\text{succ}(n)$ are the control flow successors of node n).

What is the set of live variables at node (1) (entry to the procedure) in the diagram below? What is the live set at node (2)?