Final Exam
CS3300

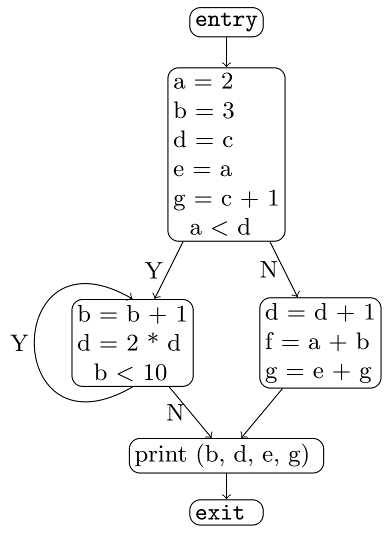
Maximum marks = 50, Time: 3hrs

27-Nov-2012

1. [10] **Control flow**
   For the following code, draw the control flow graph; mark the basic blocks and extended basic blocks. For the first statement in each of the basic blocks, compute the dominator and post-dominator information.

```
     c = a + b
     d = c
     e = d * d
L1:  f = a + c
     g = e
     a = g + d
     if (a < c) {
           h = g + 1
       L2: b = g * a
           if (h < f) goto L1:
     } else {
       f = d - g
       if (f > a)  goto L2
       else c = 2
     }
```

2. [10] **Register Allocation**
   Compute the liveness information, draw the interference graph, and do register allocation using Kempe's heuristic, assuming four registers.



3. [10] **Code Generation**
   Write the tree patterns for the following instructions with their usual meaning:

instruction	form
add	$r_i = r_j + r_k$
mul	$r_i = r_j * r_k$
addi	$r_i = r_j + c$
load	$r_i = M[r_j + c]$
store	$M[r_j + c] = r_i$
MemMove	$M[r_i] = M[r_j]$

   Draw the intermediate-code tree for the assignment statement `x = a[i+1]`. Assume that, all of the variables are located on stack. Generate machine code using the maximal munch method. Argue if your generated code is optimal or optimum. State any assumptions you make.

4. [10] **Machine independent Optimizations**
   Optimize the following code in a step by step manner. At each step, indicate the optimization applied and the resulting code.

```
void foo(){
  q = 2;
  c = q;
  b = c + 3;
  for (i=0; i < 2 * m; ++i) {
    for (j=0; j < 4 * m; ++j) {
      y = T[i] * b;
      S[i, j] = U[i, j] + V[i, j] * y + c;
      goto L2;
      S[i, j] = q + T[i] * c;
       L2: U[i, j] = y - T[i] * c;
      V[i, j] = y - T[i] / c;
    }
  }
}
```

5. [10] **Machine dependent Optimizations**
   Optimize the following snippet of the assembly code and generate optimized code in a step by step manner. At each step, indicate the optimization applied and the resulting code. All the registers are live at the end of the code. State all the assumptions that you make.

```
lw r2 4(r1)
add r6 r2 1
sub r5 r2 r6
lw r3 8(r1)
add r4 r2 r3
sw r3 8(r1)
j L1
nop
nop
add r2 r4 r3
sub r2 r2 1
L1: nop
sw r2 4(r1)
```