CS 6340 – Fall 2009 – Problem Set 1 **Name**_____

Assigned: August 18, 2009
Due: August 25, 2009

At the beginning of class on the due date, submit your neatly presented solution with this page stapled to the front (50 pts).

1. Given the following C program, construct control flow graphs (CFGs) for the program (make sure you understand the semantics of the *break* and *continue* statements). In the first CFG, use maximal basic blocks; in the second CFG, place each statement in its own basic block. In both cases, label the nodes in the graphs with the numbers of the program statements (10).

```
    main()
    {
       int sum, i, j;

1.     sum = 0;
2.     i = 1;
3.     while (i <= 5) {
4.         scanf("%d",&j);
5.         if (j < 0)
6.               continue;
7.         sum = sum + j;
8.         if (sum > 10)
9.               break;
10.        i = i + 1;
11.    }
12.    printf("sum is %d", sum);
    }
```

2. For the CFG you created in (1)
    a. Compute and show the dominator and postdominator trees for the graph (10).
    b. Use T1-T2 analysis to determine whether the graph is reducible; show all your work (10).
    c. Show the depth-first presentation of the graph, determine the depth of the CFG using that depth-first presentation, and explain how you determined the depth (10).

3. Given the following program (in a Pascal-like language), construct two control flow graphs for the program. In the first, use basic blocks; in the second, place each statement in its own basic block. In both cases, label the nodes in the graphs with the numbers of the program statements (10).

```
 procedure sqrt(real x):real
    real x1,x2,x3,eps,errval;

    begin
1.    x3 = 1
2.    errval = 0.0
3.    eps = .001
4.    if (x <= 0.0)
5.       output("illegal operand");
6.       return errval;
7.    else
8.       if (x < 1)
9.          x1 = x;
10.         x2 = 1;
11.      else
12.         x1 = eps;
13.         x2 = x;
14.      endif
15.      while ( (x2-x1) >= 2.0*eps )
16.         x3 = (x1+x2)/2.0
17.         if ( (x3*x3-x)*(x1*x1-x) < 0)
18.            x2 = x3;
19.         else
20.            x1 = x3;
21.         endif;
22.      endwhile;
23.      return x3;
24.   endif;
25. end.
```