Roll No: _____

CS6843 Program Analysis
MidSem February 29, 2016

Duration: 60 minutes Answer in the same sheet.

1. For the following program, draw the control-flow graph. [3 marks]

```
1 main() {
2  int x = 0;
3  int n = 1;
4  int a = 2;
5  int i = 0;

6   if (n > 0) {
7       for (; i < n; ++i) {
8         a = i * i;
9         if (a < 100) {
10              x += a;
11        } else {
12              x -= a;
13        }
14      }
15   }
16   printf("%d\n", x);
17 }
```

2. For the above program, perform live-variables analysis. Equations are: in(B) = use(B) U (out(B) – def(B)), out(B) = U in(S) where S is a successor of B. Simply fill-up the following table. [4 marks]

	in1	out1	in2	out2	in3	out3	outfinal
0							
1							
2							
3							
4							
5							
6							
7							

3. If your analysis is tracking one bit each for conditions x == 0 and x > 2 and y > 4, find the bit-values after every program statement below. Conservatively, each bit is set to 0. Your analysis does not have any other information, apart from that x and y are unsigned integers. [5 marks]

```
{  0 0 0   }
y = 2;
{          }
x = y + 1;
{          }
x = x – y – 1;
{          }
y = y * y + 4;
{          }
x = y – 1;
{          }
```

4. For the following set of statements, perform Andersen's analysis using constraint graph and show only the final state of the graph (nodes, their attributes and edges). [5 marks]
   *p = q, *a = b, c = a, p = &c, d = &a, q = *a, a = &b, *b = p, b = &p, q = &a.

5. Prof. George claims that while analyzing a program, if for every copy + load + store of the form P = Q, we add Q = P and run Andersen's analysis, we would get the same answer as Steensgaard's analysis run on the original program. Disprove his claim with a counter-example (use as less number of statements as you can). Make sure your counter-example is reasonable. [3 marks]