Consider the following code:

1: a = b * 2;
L1: 2: if (a >= c) goto L4;
3: d = 3*a + 4;
4: if (d <= c) goto L2;
5: x = 2 * c;
6: y = 7;
7: goto L3;
L2: 8: y = 3 * c;
L3: 9: a = a + 2;
10: goto L1;
L4: 11:

1. What are the basic blocks for this code?
   {1}, {2}, {3, 4}, {5, 6, 7}, {8}, {9, 10}, {11}

2. Draw the basic-block level CFG for this code.

   ![](image)

3. What are the loop headers? What are the back edges?
   The back edge is in red, and the loop header is in blue
4. What are the loop-invariant instructions in this code?
   Instructions 5, 6 and 8 are loop-invariant

5. Which instructions could be moved out of the loops?
   Only instruction 5 can be moved out of the loop. Because both instruction 6 and instruction 8 define y, they cannot be moved.

6. What are the induction variables for the loops in the program? The mutual induction variables?
   Induction variable: a Mutual induction variable: d

7. What does this code look like after applying strength reduction?

   ```
   1:   a = b * 2;
   1':  d' = 3*a + 4;
   L1: 2: if (a >= c) goto L4;
       3:   d = d';
       4: if (d <= c) goto L2;
       5:   x = 2 * c;
       6:   y = 7;
       7:   goto L3;
   L2: 8:   y = 3 * c;
   L3: 9:   a = a + 2;
      9':  d' = d' + 6;
    10:   goto L1;
   L4:11:   
   ```

8. What does this code look like after applying linear test replacement?

   ```
   1:   a = b * 2;
   1':  d' = 3*a + 4;
   L1: 2: if (d' >= 3*c + 4) goto L4;
       3:   d = d';
       4: if (d <= c) goto L2;
       5:   x = 2 * c;
       6:   y = 7;
       7:   goto L3;
   L2: 8:   y = 3 * c;
   L3: 9:   // a = a + 2;
      9':  d' = d' + 6;
    10:   goto L1;
   L4:11:   
   ```