CS243 Midterm Examination
Winter 2002-2003

You have 1 hour 15 minutes to work on this exam. The examination has 75 points, one point for every minute. Please budget your time accordingly.

Write your answers in the space provided on the exam. If you use additional scratch paper, please turn that in as well.

Your Name: ____________________________

In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

Signature: ____________________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>_______</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>_______</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>_______</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>_______</td>
</tr>
<tr>
<td>Total</td>
<td>75</td>
<td>_______</td>
</tr>
</tbody>
</table>
1. (10 points) Rewrite the assignments in the following loop as affine expressions as loop indices and symbolic constants, where possible.

```
a = a + 4
f = a
c = f + a
b = c + a
c = 2 * a
d = c
g = b + c
```
2. (15 points)
   a. What are the natural loop(s) in the following flow graph? (Show work for partial credit).

   ![Flow Graph Image]

   b. Is the above flow graph reducible? Explain your answer.
3. (25 points) Lazy code motion.

a. Apply the lazy code motion algorithm as described in class to the program above. Show the result of the optimization -- it is not necessary to show any intermediate steps. (Introduce new basic blocks as necessary).
b. Are there redundant operations left after the optimization? If there are, describe an algorithm (briefly) that can eliminate the kind of redundancies left in this example. Show the result of applying your algorithm to the program.
4. (25 points) Consider a language that allows only boolean variables (a boolean variable can hold either the value TRUE or FALSE), and allows only the following assignment statements:

    a = TRUE
    a = FALSE
    a = b
    a = NOT b

Devise a monotone data flow algorithm that determines, for all points in a program, whether each variable can possibly hold a TRUE value.

a. What is the direction of your data flow analysis?

b. Draw a diagram of the lattice for one variable, identifying the top and bottom elements clearly.

c. How do you initialize the information at the entry/exit nodes?

d. How do you initialize the iterative algorithm?

e. Define the transfer function of a basic block.
f. Is your data flow framework distributive?

g. Will your algorithm necessarily converge? If so, why?

h. Explain how you can tell if a variable may have a TRUE value at each program point.