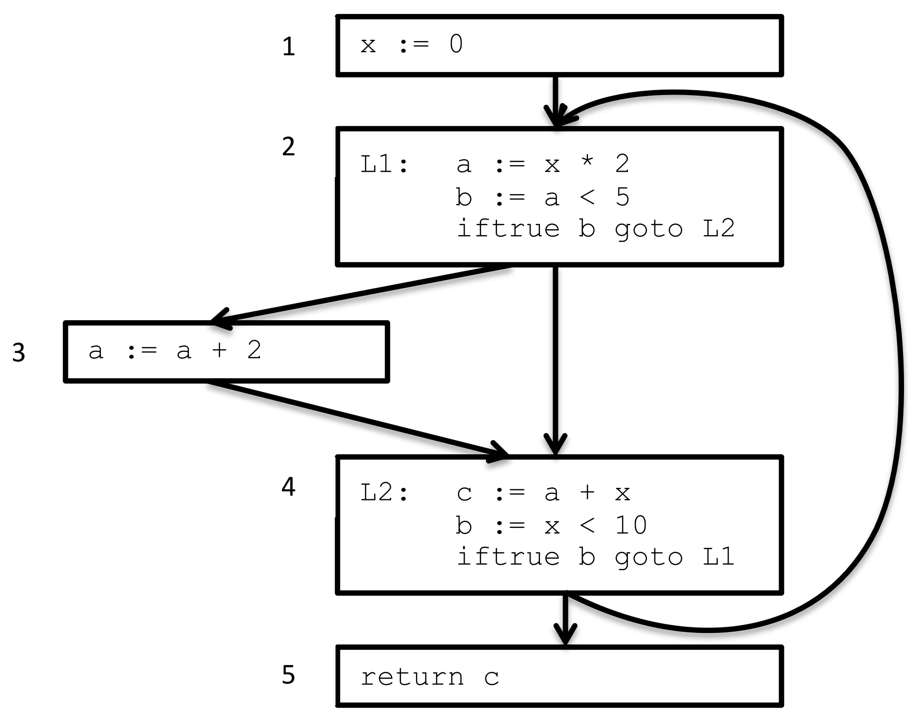
CS 2210 – Homework 4
Due: Monday, April 22, 2013 at the start of class

1.) Identify the basic blocks in the following sequence of IR code and construct the Control Flow Graph:

```
      x := 0
L1:   a := x * 2
      b := a < 5
      iftrue b goto L2
      a := a + 2
L2:   c := a + x
      b := x < 10
      iftrue b goto L1
      return c
```



2.) Perform liveness analysis on the variables in the above code statement by statement. Show each iteration of the algorithm in terms of live-in and live-out.

```
1:          x := 0
2:    L1:   a := x * 2
3:          b := a < 5
4:          iftrue b goto L2
5:          a := a + 2
6:    L2:   c := a + x
7:          b := x < 10
8:          iftrue b goto L1
9:          return c
```

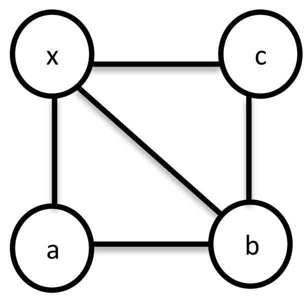
Block	**Use**	**Def**	**Successors**
9	c		
8	b		2, 9
7	x	b	8
6	a, x	c	7
5	a	a	6
4	b		5, 6
3	a	b	4
2	x	a	3
1		x	2

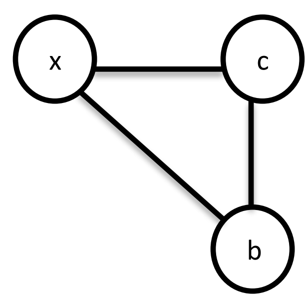
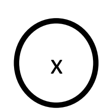
	Iteration 1		**Iteration 2**		**Iteration 3**		**Iteration 4**		**Iteration 5**	
Stmt	**in**	**out**	**In**	**out**	**in**	**out**	**in**	**out**	**in**	**out**
9	c		c		c		c		c	
8	b	c	b, c	c, x	b, c, x	c, x	b, c, x	c, x	b, c, x	c, x
7	x	b	x	b, c	c, x	b, c	c, x	b, c, x	c, x	b, c, x
6	a, x	x	a, x	x	a, x	x	a, x	c, x	a, x	c, x
5	a	a, x	a, x	a, x	a, x	a, x	a, x	a, x	a, x	a, x
4	b	a, x	a, b, x	a, x	a, b, x	a, x	a, b, x	a, x	a, b, x	a, x
3	a	b	a	a, b, x	a, x	a, b, x	a, x	a, b, x	a, x	a, b, x
2	x	a	x	a	x	a, x	x	a, x	x	a, x
1		x		x		x		x		x

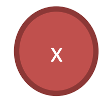
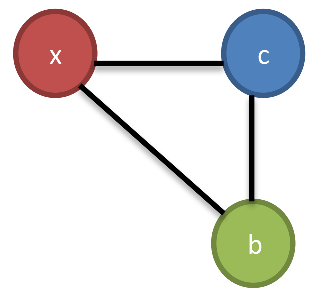
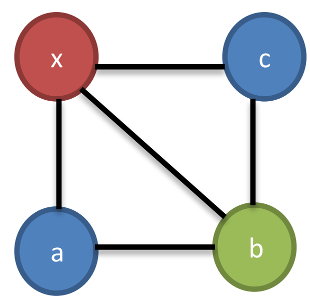
3.) Construct the interference graph and perform register allocation using K=3 registers. Show the order that simplify removes the nodes from the graph and then the resulting colors as it is rebuilt.

```
1:          x := 0
2:    L1:   a := x * 2
3:          b := a < 5
4:          iftrue b goto L2
5:          a := a + 2
6:    L2:   c := a + x
7:          b := x < 10
8:          iftrue b goto L1
9:          return c
```

Stmt	in	out
1		x
2	x	a, x
3	a, x	a, b, x
4	a, b, x	a, x
5	a, x	a, x
6	a, x	c, x
7	c, x	b, c, x
8	b, c, x	c, x
9	c	



<table>
<tr><td>Simplify a</td><td>
</td></tr>
<tr><td>Simplify b</td><td>
</td></tr>
<tr><td>Simplify c</td><td>
</td></tr>
<tr><td>Simplify x</td><td></td></tr>
</table>

<table>
<tr><td>Select x, color red</td><td>
</td></tr>
<tr><td>Select c, color blue</td><td>
</td></tr>
<tr><td>Select b, color green</td><td>
</td></tr>
<tr><td>Select a, color blue</td><td>
</td></tr>
</table>

4.) Convert the above code into SSA Form.

Dominators

Dom(1) = {1}

Dom(2) = {1, 2}

Dom(3) = {1, 2, 3}

Dom(4) = {1, 2, 4}

Dom(5) = {1, 2, 4, 5}

Immediate Dominators

IDom(1) = {}

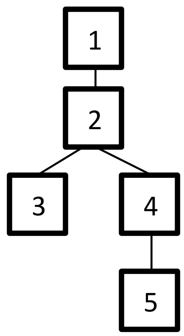
IDom(2) = {1}

IDom(3) = {2}

IDom(4) = {2}

IDom(5) = {4}

IDom Tree



Dominance Frontier

Block 1: No preds

Block 2: runner = 1, IDom(2) = 1

    Done.

    Runner = 4

        DF(4) = {2}

    Runner = 2

        DF(2) = {2}

    Runner = 1

    Done.

Block 3: 1 Pred

Block 4: Runner = 2, IDom(4) = 2

    Done.

    Runner = 3

        DF(3) = {4}

    Runner = 2

Done.

Block 5: 1 Pred

Inserting Phi Functions

Defsites:

X = {1}

A={2, 3}

B = {2, 4}

C = {4}

W = 1

DF(1) = {}

Done.

W ={2, 3}

DF(2) = 2

Insert a := ϕ(a, a) at the top of block 2

DF(3) = 4

Insert a := ϕ(a, a) at the top of block 4

Done.

W = {2, 4}

DF(2) = {2}

Insert b := ϕ(b, b) at the top of block 2

DF(4) = {2}

Already there

Done.

W = {4}

DF(4) = 2

Insert c := ϕ(c, c) at the top of block 2

Note that our algorithm assumes that all variables are defined in the entry block, hence the phi functions that seem to be unnecessary in block 2. They are dead code and dead-code elim will remove them.

Numbering

```
       X1 := 0
L1:    a1 := φ(a0, a4)
       b1 := φ(b0, b2)
       c1 := φ(c0, c2)
       a2 := x1 * 2
       b1 := a2 < 5
       iftrue b1 goto L2
       a3 := a2 + 2
L2:    a4 := φ(a2, a3)
       c2 := a3 + x1
       b2 := x < 10
       iftrue b2 goto L1
       return c2
```