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

\[
x := 0 \\
L1: \quad a := x \times 2 \\
\quad b := a < 5 \\
\quad \text{iftrue } b \ \text{goto } L2 \\
\quad a := a + 2 \\
L2: \quad c := a + x \\
\quad b := x < 10 \\
\quad \text{iftrue } b \ \text{goto } L1 \\
\quad \text{return } c
\]
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
```

<table>
<thead>
<tr>
<th>Block</th>
<th>Use</th>
<th>Def</th>
<th>Successors</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>b</td>
<td></td>
<td>2, 9</td>
</tr>
<tr>
<td>7</td>
<td>x</td>
<td>b</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>a, x</td>
<td>c</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>a</td>
<td>a</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>b</td>
<td></td>
<td>5, 6</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>b</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>a</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stmt</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
<th>Iteration 3</th>
<th>Iteration 4</th>
<th>Iteration 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>in</td>
<td>out</td>
<td>in</td>
<td>out</td>
<td>in</td>
</tr>
<tr>
<td>9</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
</tr>
<tr>
<td>8</td>
<td>b</td>
<td>c</td>
<td>b, c</td>
<td>c, x</td>
<td>b, c</td>
</tr>
<tr>
<td>7</td>
<td>x</td>
<td>b</td>
<td>x</td>
<td>b, c</td>
<td>c, x</td>
</tr>
<tr>
<td>6</td>
<td>a, x</td>
<td>x</td>
<td>a, x</td>
<td>x</td>
<td>a, x</td>
</tr>
<tr>
<td>5</td>
<td>a</td>
<td>a, x</td>
<td>a, x</td>
<td>x</td>
<td>a, x</td>
</tr>
<tr>
<td>4</td>
<td>b</td>
<td>a, x</td>
<td>a, b, x</td>
<td>a, x</td>
<td>a, b,x</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>a, b,x</td>
<td>a, x</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>a</td>
<td>a</td>
<td>a,x</td>
<td>a</td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>
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.

```plaintext
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
```

<table>
<thead>
<tr>
<th>Stmt</th>
<th>in</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>a, x</td>
</tr>
<tr>
<td>3</td>
<td>a, x</td>
<td>a, b, x</td>
</tr>
<tr>
<td>4</td>
<td>a, b, x</td>
<td>a, x</td>
</tr>
<tr>
<td>5</td>
<td>a, x</td>
<td>a, x</td>
</tr>
<tr>
<td>6</td>
<td>a, x</td>
<td>c, x</td>
</tr>
<tr>
<td>7</td>
<td>c, x</td>
<td>b, c, x</td>
</tr>
<tr>
<td>8</td>
<td>b, c, x</td>
<td>c, x</td>
</tr>
<tr>
<td>9</td>
<td>c</td>
<td></td>
</tr>
</tbody>
</table>

Simplify a

Simplify b

Simplify c

Simplify x
### Dominators

| $\text{Dom}(1)$ | $\{1\}$ |
| $\text{Dom}(2)$ | $\{1, 2\}$ |
| $\text{Dom}(3)$ | $\{1, 2, 3\}$ |
| $\text{Dom}(4)$ | $\{1, 2, 4\}$ |
| $\text{Dom}(5)$ | $\{1, 2, 4, 5\}$ |

### Immediate Dominators

| $\text{IDom}(1)$ | $\{}$ |
| $\text{IDom}(2)$ | $\{1\}$ |
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 \( a := \phi(a, a) \) at the top of block 2

\[ DF(3) = 4 \]

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

Done.

\[ W = \{2, 4\} \]

\[ DF(2) = \{2\} \]

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

\[ DF(4) = \{2\} \]

Already there

Done.
W = \{4\}

DF(4) = 2

Insert \(c := \phi(c, c)\) 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

\[
\begin{align*}
X_1 & := 0 \\
L1: & \quad a_1 := \phi(a_0, a_4) \\
& \quad b_1 := \phi(b_0, b_2) \\
& \quad c_1 := \phi(c_0, c_2) \\
& \quad a_2 := x_1 \times 2 \\
& \quad b_1 := a_2 < 5 \\
& \quad \text{iftrue } b_1 \text{ goto } L2 \\
& \quad a_3 := a_2 + 2 \\
L2: & \quad a_4 := \phi(a_2, a_3) \\
& \quad c_2 := a_3 + x_1 \\
& \quad b_2 := x < 10 \\
& \quad \text{iftrue } b_2 \text{ goto } L1 \\
& \quad \text{return } c_2
\end{align*}
\]