1. Consider the pseudo code program:

```
begin
  a := readint();
  b := readint();
  c := a + b;
  if (a > b)
    d := c;
    e := 2;
    f := d + e;
  else
    d := 0;
    if (a == b)
      d := 1;
    endif
    e := 1;
    f := d + e;
  endif
  endif
  writeint(e);
  writeint(f);
end
```

Assign registers to the variables $a$ to $f$ using graph coloring. That is, from the live ranges of variables in the CFG representation of the program (show the CFG) determine the conflict graph and the minimum number of registers.
2. Consider the following block of three-address code:

\[
\begin{align*}
\text{r1} &= \text{r1} + \text{r2} \\
\text{r2} &= \text{r9} + \text{r1} \\
\text{r4} &= \text{r2} \\
\text{r6} &= \text{r2} + 1 \\
\text{r9} &= \text{r6} \\
\text{r6} &= \text{r4} + 3 \\
\text{r8} &= \text{r2} + \text{r9}
\end{align*}
\]

Apply local forward copy propagation and show the result, then local backward copy propagation and show the final result.

3. Partition the following fragment of three-address code into a CFG with basic blocks:

\[
\begin{align*}
\text{k} &= 4 \\
\text{n} &= 1 \\
\text{i} &= \text{k} + 7 \\
\text{if k > 0 goto L2}
\text{L1: i} &= \text{i - 1} \\
\text{n} &= 2 \times \text{number} \\
\text{if i != 0 goto L1}
\text{goto L3}
\text{L2: n} &= 2 \times \text{k}
\text{L3: halt}
\end{align*}
\]

Also draw the dominator tree of the CFG.