Our little SKI programming language consisted of the constants $S$, $K$, and $I$, and all combinations of them. We had three computation rules:

\[ Ix \rightarrow x \quad Kxy \rightarrow x \quad Sxyz \rightarrow (xz)(yz) \]

Recall that we defined $T$ (true) and $F$ (false) in our language as the programs $K$ and $KI$. Using these, we can define programs that do logic. Here are definitions of some new constants and their computation rules:

\[ V_{xyz} \rightarrow zxy \quad R_{xyz} \rightarrow yzx \quad H_{xy} \rightarrow yx \]

We can define logical negation (not) as $VFT$, logical conjunction (and) as $RF$, and logical disjunction (or) as $HT$. What we mean by this is that we can combine (apply) these programs to the truth values above, apply the reduction rules, and get the appropriate truth value. For example, negation should take one truth value and compute its negation, so we would expect $VFTT$ computes to $F$ and $VFTF$ computes to $T$. Show that this is the case. Then show the cases for conjunction and disjunction. I.e., show the following reductions:

\[
\begin{align*}
VFTT & \sim F & HTFF & \sim F \\
VFTF & \sim T & RFFF & \sim F & HTFT & \sim T \\
RFTT & \sim T & HTTT & \sim T
\end{align*}
\]

You should only use the 6 computation rules given above, and the fact that $T = K$ and $F = KI$.

As an example, here is the second case for disjunction:

\[
RFFT \rightarrow FTF \quad \text{by } R \text{ computation rule}
\]

\[
= (KI)K(KI) \quad \text{substituting for } T \text{ and } F
\]

\[
\rightarrow I(KI) \quad \text{by } K \text{ computation rule}
\]

\[
\rightarrow KI \quad \text{by } I \text{ computation rule}
\]

\[
= F \quad \text{by definition of } F
\]

See the class notes (http://www.cse.psu.edu/~hannan/fys) for the examples we did in class.

**Bonus:** For the adventurous, see if you can define $V$, $R$, and $H$ in terms of just $S$, $K$, and $I$. I.e., can you find some programs $P_V$, $P_R$, and $P_H$, constructed from only $S$, $K$, and $I$ such that

\[
\begin{align*}
P_{Vxyz} & \sim zxy \\
P_{Rxyz} & \sim yzx \\
P_{Hxy} & \sim yx
\end{align*}
\]

(Warning: this is a very difficult problem)