A resettable variable is defined as follows. There are three new names: value (of type X), set (a procedure with one parameter of type X), and reset (a program). Here are the axioms:

\[
\begin{align*}
\text{value} &= x \\ \text{value}' &= x \iff \text{set } x \\
\text{value}' &= \text{value} \iff \text{set } x. \text{reset} \\
\text{reset}. \text{reset} &= \text{reset}
\end{align*}
\]

Implement this data structure, with proof.

Let value: X be a user's variable, and let old: X be an implementer's variable.

\[
\begin{align*}
\text{set} &= \langle x: X. \text{old} := \text{value}. \text{value} := x \rangle \\
\text{reset} &= \text{value} := \text{old}
\end{align*}
\]

Proof:

\[
\begin{align*}
\text{(value}' &= x \iff \text{set } x) \\
\iff \text{(value}' &= x \iff \text{old} := \text{value}. \text{value} := x) \\
\iff \text{(value}' &= x \iff \text{old}' := \text{value} \land \text{value}' = x) \\
\iff \top \\
\text{(value}' &= \text{value} \iff \text{set } x. \text{reset}) \\
\iff \text{(value}' &= \text{value} \iff \text{old} := \text{value}. \text{value} := x. \text{value} := \text{old}) \\
\iff \text{(value}' &= \text{value} \iff \text{old}' := \text{value}' = \text{value}) \\
\iff \top \\
\text{(reset. reset} &= \text{reset}) \\
\iff \text{(value} := \text{old}. \text{value} := \text{old} \iff \text{value} := \text{old}) \\
\iff \top
\end{align*}
\]