Let \( a \), \( b \), and \( x \) be natural variables. Variables \( a \) and \( b \) are implementer's variables, and \( x \) is a user's variable for the operations

- \( \text{start} \equiv a := 0 \), \( b := 0 \)
- \( \text{step} \equiv a := a+1 \), \( b := b+2 \)
- \( \text{ask} \equiv x := a+b \)

Reimplement this theory replacing the two old implementer's variables \( a \) and \( b \) with one new natural implementer's variable \( c \).

(a) What is the data transformer?

\[ c = a+b \]

(b) Using your data transformer, transform \( \text{step} \).

\[
\forall a, b \cdot c = a+b \Rightarrow \exists a', b' \cdot c' = a'+b' \land (a := a+1 \land b := b+2) \text{ replace program}
\]

\[
\equiv \forall a, b \cdot c = a+b \Rightarrow \exists a', b' \cdot c' = a'+b' \land a' = a+1 \land b' = b+2 \land x' = x \text{ one-point}
\]

\[
\equiv \forall a, b \cdot c = a+b \Rightarrow c' = a+1+b+2 \land x' = x
\]

\[
\equiv \forall a, b \cdot b = c-a \Rightarrow c' = a+b+3 \land x' = x \text{ one-point for } b
\]

\[
\equiv \forall a \cdot c' = c+3 \land x' = x
\]

\[
\equiv c' = c+3 \land x' = x
\]

\[
\equiv c := c+3
\]