0[9] Let $i$ be an integer variable and let $L$ be a list of integers. What is the exact precondition for $L' = 3 \rightarrow 4$ to be refined by $L := i \rightarrow 4 \mid L$?

\[
\forall L', i' \cdot (L' = 3 \rightarrow 4) \iff L := i \rightarrow 4 \mid L
\]

$\equiv \forall L', i' \cdot (L' = 3 \rightarrow 4) \iff L = i \rightarrow 4 \mid L \land i = i$ one-point twice

$\equiv (i \rightarrow 4 \mid L) = 3 = 4$

$\equiv i = 3 \lor L = 3 = 4$

1[9] Let $a$, $b$, and $c$ be integer variables. Express as simply as possible without using quantifiers, assignments, or dependent compositions

$\equiv a := a + b$. $b := a + b$. $c := a + b$

$\equiv a := a + b$. $b := a + b$. $a' = a \land b' = b \land c' = a + a + b$

$\equiv a' = a + b \land b' = a + b + b \land c' = a + b + a + b + b$

2 The syntax

\[
\text{if } b \rightarrow P \text{ [] } c \rightarrow Q \text{ fi}
\]

where $b$ and $c$ are binary and $P$ and $Q$ are programs, was a popular programming language syntax for several years. It can be executed as follows. If exactly one of $b$ and $c$ is true initially, then the corresponding program is executed; if both $b$ and $c$ are true initially, then either one of $P$ or $Q$ (arbitrary choice) is executed; if neither $b$ nor $c$ is true initially, then execution is completely arbitrary.

(a)[6] Formally specify this construct using the notations of this course.

\[
\text{if } b \rightarrow P \text{ [] } c \rightarrow Q \text{ fi } = b \lor c \Rightarrow b \land P \lor c \land Q
\]

or

\[
\text{if } b \rightarrow P \text{ [] } c \rightarrow Q \text{ fi } = \text{ if } b \land \neg c \text{ then } P \text{ else } \neg b \land c \text{ then } Q \text{ else } b \land c \text{ then } P \lor Q \text{ else } \top \text{ fi } \text{ fi } \text{ fi}
\]

(b)[6] Refine this specification as a program using the notations of this course.

\[
\text{if } b \rightarrow P \text{ [] } c \rightarrow Q \text{ fi } \iff \text{ if } b \text{ then } P \text{ else } Q \text{ fi}
\]

3 Let $a$ be a binary implementer's variable, and let $b$ be a binary user's variable for the operations

\[
\text{set } \equiv a := T
\]

\[
\text{flip } \equiv a := \neg a
\]

\[
\text{ask } \equiv b := a
\]

This theory must be reimplemented using integer variable $i$ in place of $a$, using 0 for $\perp$ and all other integers for T; this is the encoding used in the C language.

(a)[3] What is the data transformer?

\[
a := (i \neq 0)
\]

(b)[9] Transform $\text{flip}$.

\[
\forall a \cdot a := (i \neq 0) \Rightarrow \exists a' \cdot a' := (i \neq 0) \land (a := \neg a)
\]

\[
\equiv \forall a \cdot a := (i \neq 0) \Rightarrow \exists a' \cdot a' := (i \neq 0) \land a \neq a \land b' = b
\]

one-point twice

\[
\equiv (i \neq 0) \lor (i \neq 0) \land b' = b
\]

\[
\iff \text{ if } i = 0 \text{ then } i := 1 \text{ else } i := 0 \text{ fi}
\]