Given natural list variable \( L \), index variable \( i \), and time variable \( t \), increase each list item by 1 until you have created item 100. The time is bounded by \#L. The program is

\[
i := 0, \\
\text{do } \text{ exit when } i = \#L. \\
\quad L\ i := L\ i + 1. \\
\quad \text{exit when } L\ i = 100. \\
\quad i := i + 1 \odo
\]

Write a formal specification, and prove it is refined by the program.

\[\text{§ Define } k \text{ as the first index where } L\ k = 99 \text{ or } \#L \text{ if there's no such index.}
\]

\[\neg (\exists j: 0,..k: L\ j = 99) \land (L\ k = 99 \lor k = \#L)\]

Now the specification \( S \) is

\[
(\forall j: 0,..k: L'\ j = L\ j + 1) \\
\land (L\ k = 99 \land L'\ k = 100 \land (\forall j: k+1,..\#L: L'\ j = L\ j) \lor k = \#L) \\
\land t' \leq t + \#L
\]

Define loop specification \( P \) to be like \( S \) but from index \( i \) rather than from 0.

\[
(\forall j: i,..k: L'\ j = L\ j + 1) \\
\land (L\ k = 99 \land L'\ k = 100 \land (\forall j: k+1,..\#L: L'\ j = L\ j) \lor k = \#L) \\
\land t' \leq t + \#L - i
\]

We have two refinements to prove.

\[S \iff i := 0. \quad P\]

\[P \iff \text{if } i = \#L \text{ then ok else } L := i \rightarrow L\ i + 1 \mid L. \quad \text{if } L\ i = 100 \text{ then ok else } i := i + 1. \quad P \fi\]

The first is easy: replacing \( i \) by 0 in \( P \) we obtain \( S \). We prove the last refinement by cases. First case.

\[i = \#L \land \text{ok } \Rightarrow P \]

\[= \top \]

Last refinement, last case.

\[i \neq \#L \land (L := i \rightarrow L\ i + 1 \mid L. \quad \text{if } L\ i = 100 \text{ then ok else } i := i + 1. \quad P \fi) \Rightarrow P \]

\[= \top \]

\[= \top \]

\[= \top \]