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

$$i := 0.$$  
**do**  
**exit when** $i = \#L$.  
$L i := L i + 1.$  
**exit when** $L i = 100.$  
$i := i + 1$  
**od**

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

§ Define $k$ as the first index where $L k = 99$, or $\#L$ if there's no such index.

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

Now the specification $S$ is

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

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

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

We have two refinements to prove.

$$S \iff i := 0 \cdot P$$
$$P \iff \text{if } i = \#L \text{ then } ok$$
$$\text{else } L := i \rightarrow (L i + 1) \mid \text{L. if } L i = 100 \text{ then } ok \text{ else } i := i + 1 \cdot P$$

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

$i = \#L \land ok \Rightarrow P$  
\text{UNFINISHED}

Last refinement, last case.

$i \neq \#L \land (L := i \rightarrow (L i + 1) \mid \text{L. if } L i = 100 \text{ then } ok \text{ else } i := i + 1 \cdot P) \Rightarrow P$  
\text{UNFINISHED}