Using item comparison but not list comparison, write a program to determine whether one list comes before another in the list order.

Let the lists be $L$ and $M$. We need binary variable $s$ to record the answer, and natural variable $n$ as index.

$$s' = (L < M) \iff n := 0. \ n \leq \#L \land n \leq \#M \implies s' = (L[n;..\#L] < M[n;..\#M])$$

$$n \leq \#L \land n \leq \#M \implies s' = (L[n;..\#L] < M[n;..\#M]) \iff$$

if $n = \#M$ then $s := \bot$
else if $n = \#L$ then $s := \top$
else if $L[n] < M[n]$ then $s := \bot$
else if $L[n] > M[n]$ then $s := \bot$
else $n := n + 1$. $$n \leq \#L \land n \leq \#M \implies s' = (L[n;..\#L] < M[n;..\#M]) \iff \text{Fi fi fi fi fi}$$

First refinement, right side:

$$n := 0. \ n \leq \#L \land n \leq \#M \implies s' = (L[n;..\#L] < M[n;..\#M]) \iff n = \#M \land (s := \bot)$$ substitution law

$$\iff 0 \leq \#L \land 0 \leq \#M \implies s' = (L[0;..\#L] < M[0;..\#M])$$

$$\iff s' = (L < M)$$

Last refinement, by cases. First case:

$$(n \leq \#L \land n \leq \#M \implies s' = (L[n;..\#L] < M[n;..\#M])) \iff n = \#M \land (s := \bot)$$ portation, expand assignment

$$\iff n = \#M \leq \#L \land s' = \bot \land n' = n \implies s' = (L[n;..\#L] < M[n;..\#M])$$ context

$$\iff n = \#M \leq \#L \land s' = \bot \land n' = n \implies L[#M;..\#L] \geq M[#M;..\#M]$$ (any list) \geq \text{nil}

$$\iff \top$$

Second case:

$$(n \leq \#L \land n \leq \#M \implies s' = (L[n;..\#L] < M[n;..\#M])) \iff n + \#M \land n = \#L \land (s := \top)$$ portation, expand assignment

$$\iff n = \#L < \#M \land s' = \top \land n' = n \implies s' = (L[n;..\#L] < M[n;..\#M])$$ context

$$\iff n = \#L < \#M \land s' = \top \land n' = n \implies L[#L;..\#L] < M[#L;..\#M]\ [\text{nil}] < \text{any nonempty list}$$

$$\iff \top$$

Middle case:

$$(n \leq \#L \land n \leq \#M \implies s' = (L[n;..\#L] < M[n;..\#M])) \iff n + \#M \land n + \#L \land L[n] \leq M[n] \land (s := \top) \land M[n;..\#M]$$ portation, expand assignment

$$\iff n < \#L \land n < \#M \land L[n] < M[n] \land s' \land n' = n \implies s' = (L[n;..\#L] < M[n;..\#M]) \land s'$$ context $s'$ weaken antecedent

$$\iff n < \#L \land n < \#M \land L[n] < M[n] \land s' \land n' = n \implies L[n;..\#L] < M[n;..\#M]$$ the segments are nonempty and the first item of $L[n;..\#L]$ is less than the first item of $M[n;..\#M]$

$$\iff \top$$

Next-to-last case:

$$(n \leq \#L \land n \leq \#M \implies s' = (L[n;..\#L] < M[n;..\#M])) \iff n + \#M \land n + \#L \land L[n] > M[n] \land (s := \bot) \land M[n;..\#M]$$ portation, expand assignment

$$\iff n < \#L \land n < \#M \land L[n] > M[n] \land s' = \bot \land n' = n \implies s' = (L[n;..\#L] < M[n;..\#M])$$ context $s'$ weaken antecedent

$$\iff n < \#L \land n < \#M \land L[n] > M[n] \land \lnot s' \land n' = n \implies L[n;..\#L] \geq M[n;..\#M]$$ weaken antecedent
\( \iff n \not< L \land n \not< M \land L n > M n \Rightarrow L[n;\#L] \geq M[n;\#M] \) the segments are nonempty and the first item of \( L[n;\#L] \) is greater than the first item of \( M[n;\#M] \)

\( \equiv \top \)

Last case:

\( (n \leq \#L \land n \leq \#M \Rightarrow s' = (L[n;\#L] < M[n;\#M])) \)

\( \iff n+1 \not\in L \land n+1 \not\in M \land L n = M n \)

\( \land (n := n+1. \ n \leq \#L \land n \leq \#M \Rightarrow s' = (L[n;\#L] < M[n;\#M])) \)

\( \Rightarrow s' = (L[n+1;\#L] < M[n+1;\#M]) \) discharge

\( \equiv n \not< L \land n \not< M \land L n = M n \land s' = (L[n+1;\#L] < M[n+1;\#M]) \)

context \( s' \) and weaken antecedent

\( \Rightarrow (L[n+1;\#L] < M[n+1;\#M]) = (L[n;\#L] < M[n;\#M]) \) the segments are nonempty and their first items are equal so the remaining items determine the order of the segments

\( \equiv \top \)