You are given a sorted list of \( m \) different numbers, all in the range \( 0,..n \). Write a program to find the lexicographically next sorted list of \( m \) different numbers, all in the range \( 0,..n \).

Here is the last sorted list of 5 different numbers all in the range 0,..10.

\[ [5; 6; 7; 8; 9] \]

At index \( i \), the largest possible item is \( n-m+i \). Strategy: find the last item that is below its maximum, increase it by 1, then fill up the following items in increasing order. For example, if the sorted list of 5 different numbers in the range 0,..10 is

\[ [2; 4; 7; 8; 9] \]

the last item that is below its maximum is the 4. So increase the 4 to 5, then fill up the rest and get

\[ [2; 5; 6; 7; 8] \]

To find the last, search from the end back toward the beginning. To make the specification implementable, we have to decide what to do if we are given the last list; I choose that we leave it as is.

Let \( L \) be a list variable whose initial value is the given sorted list of length \( m \) with items all in \( 0,..n \). Let \( i \) be a nat variable used to index \( L \). Define specifications

\[
S = \begin{cases} 
    \text{if } L = [n-m;..n] \text{ then } L' = L \text{ else } \text{UNFINISHED} & \text{fi} \\
    A = \text{UNFINISHED} \\
    B = \text{UNFINISHED}
\end{cases}
\]

The refinements are

\[
S \Leftarrow i := m. \ A
\]

\[
A \Leftarrow \begin{cases} 
    \text{if } i = 0 \text{ then } \text{ok} & \text{the given list is the last} \\
    \text{else } i := i-1. \text{ if } Li = n-m+i \text{ then } A \text{ the item is max} \\
    \text{else } Li := Li+1. \ B \text{ fi fi}
\end{cases}
\]

\[
B \Leftarrow i := i+1. \text{ if } i = m \text{ then } \text{ok else } Li := L(i-1)+1. \ B \text{ fi}
\]

The proofs are UNFINISHED