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 \textit{nat} variable used to index \( L \). Define specifications

\[
S = \text{if } L = [n-m;..n] \text{ then } L' = L \text{ else UNFINISHED fi}
\]

\[
A = \text{UNFINISHED}
\]

\[
B = \text{UNFINISHED}
\]

The refinements are

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

\[
A \iff \text{if } i=0 \text{ then ok else } i := i - 1. \text{ if } L \ i = n-m+i \text{ then } A \text{ else } L \ := \ L \ i + 1. \ B \ fi \ fi
\]

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

The proofs are UNFINISHED