442 A particular program-list has the following operations:

- the operation *mkempty* makes the list empty
- the operation *extend x* joins item *x* to the end of the list
- the operation *swap i j* swaps the items at indexes *i* and *j*
- the expression *length* tells the length of the list
- the expression *item i* tells the item at index *i*

(a) Write axioms to define this program-list.

(b) Implement this program-list, with proof.

After trying the question, scroll down to the solution.

(a) Write axioms to define this program-list.

§ Maybe the clearest definition of the program-list is an implementation of it. I suppose the items are of type X. Although the question talks about lists, I'll use strings. I introduce implementer's variable S of type $*X$.

$mkempty = S := nil$
$extend = \langle x: X \cdot S := S;x \rangle$
$swap = \langle i, j: nat \cdot S := (S \triangleleft i \triangleright S_j) \triangleleft j \triangleright S_i \rangle$
$length = \leftrightarrow S$
$item = \langle i: nat \cdot S_i \rangle$

(b) Implement this program-list, with proof.

§ Using an implementation for the definition does not mean that we are stuck with this implementation. We are still free to implement it differently. If I stick with this implementation, no proof is needed.