225 (flatten) Write a program to flatten a list. The result is a new list just like the old one but without the internal structure. For example,

L = [[3; 5]; 2; [5; [7]; [*nil*]]]
L' = [3; 5; 2; 5; 7]

Your program may employ a test $L\ i$: *int* to see if an item is an integer or a list.

After trying the question, scroll down to the solution.

§ Define $flat\ L$ to mean that list L is flat.

$$flat\ L \;=\; \forall i\colon 0,..\#L\cdot L\ i\colon int$$

or more succinctly

$$flat\ L \;=\; L\colon [*int]$$

Define $sim\ L\ M$ to mean that lists L and M have the same items in the same order, though they may have different internal structure. Formally, it is easier to define sim for all strings. Let s , t , and u be strings, and let i and j be integers.

$$sim\ s\ s$$
$$sim\ s\ t \;=\; sim\ t\ s$$
$$sim\ s\ t \wedge sim\ t\ u \;\Rightarrow\; sim\ s\ u$$
$$sim\ s\ nil \;=\; s{=}nil$$
$$sim\ (i;\ s)\ (j;\ t) \;=\; i{=}j \wedge sim\ s\ t$$
$$sim\ (s;\ [t];\ u)\ (s;\ t;\ u)$$

Define specifications P and Q as

$$P \;=\; flat\ L' \;\wedge\; sim\ L\ L'$$
$$Q \;=\; L'[0;..k]{=}L[0;..k] \wedge flat\ (L'[k;..\#L']) \wedge sim\ (L[k;..\#L])\ (L'[k;..\#L'])$$

Then the refinements are

```
$P \Leftarrow k:= 0.\ Q$
$Q \Leftarrow$ if $k{=}\#L$ then $ok$
                 else if $L\ k\colon int$ then $k:= k+1.\ Q$
                      else $L:= L[0;..k]\ ;;\ L\ k\ ;;\ L[k+1;..\#L].\ Q$ fi fi
```