You are given a text $t$ of characters drawn from the alphabet “x”, “(”, “)”, “[”, “]”. Write a program to determine if $t$ has its brackets properly paired and nested.

§ Introduce variable $n: \text{nat}$, and a stack. Define predicate $P: \text{text} \to \text{bin}$ so that $P s$ means that $s$ has its brackets properly paired and nested. Here are its axioms.

$$
\begin{align*}
P & \quad \text{""} \\
P & \quad \text{"x"} \\
P a & = \quad P ("( ; a ; ")) \\
P a & = \quad P ("[ ; a ; "]") \\
P a \land P z & \Rightarrow P (a ; z)
\end{align*}
$$

Now we need two specifications.

$$
\begin{align*}
R & = \quad (\text{isempty' } \land n' \leftrightarrow t \equiv P t) \\
Q & = \quad \text{(defined later)}
\end{align*}
$$

Here are the refinements.

$$
\begin{align*}
R & \iff \text{mkempty. } n:= 0. \quad Q \\
Q & \iff \quad \text{if } n \leftrightarrow t \text{ then } \text{ok} \\
& \quad \text{else if } t_n = \text{"x" } \text{ then } n:= n+1. \quad Q \\
& \quad \text{else if } t_n = (" \text{ then } \text{push "}) \text{. } n:= n+1. \quad Q \\
& \quad \text{else if } t_n = [" \text{ then } \text{push ["]} \text{. } n:= n+1. \quad Q \\
& \quad \text{else if } \text{isempty } \text{ then } \text{ok} \\
& \quad \text{else if } t_n = \text{top } \text{ then } \text{pop. } n:= n+1. \quad Q \\
& \quad \text{else ok } fi \ fi \ fi \ fi \ fi
\end{align*}
$$

I have used a stack, and for the purpose of executing the program, the stack can be implemented any way that is correct. But for the purpose of defining specification $Q$, I implement it as follows. Let $s$ be a text-valued implementer's variable.

$$
\begin{align*}
\text{mkempty} & = \quad s:= \text{nil} \\
\text{isempty} & = \quad \leftrightarrow s = 0 \\
\text{push} & = \quad \langle c : \text{char} \to s:= c ; s \rangle \\
\text{top} & = \quad s_0 \\
\text{pop} & = \quad s:= s_1 ; \leftrightarrow s
\end{align*}
$$

Now I can define specification $Q$.

$$
Q = \quad P (t_0 ; n ; s) \Rightarrow R
$$

And finally we can prove the refinements. UNFINISHED