1. Define the property $\text{Len}$ (length of a list) as an Idris data type.

\[
\text{data Len : List a → Nat → Type where}
\]
\[
\text{Empty : Len [] 0}
\]
\[
\text{NonEmpty : Len xs n → Len (x∷xs) (S n)}
\]

2. Prove the following theorem about $\text{Len}$.

\[
\text{lenAdd : Len xs n → Len ys m → Len (xs++ys) (n+m)}
\]
\[
\text{lenAdd \{xs=[]\} Empty q = q}
\]
\[
\text{lenAdd \{xs=z∷zs\} (NonEmpty p) q = NonEmpty (lenAdd \{xs=zs\} p q)}
\]
Exercises

3. **Prove the lemma** LenS.

\[ \text{lenS : Len } xs \text{ n } \rightarrow \text{Len } (x::xs) \text{ (S n)} \]

\[ \text{lenS } = \text{NonEmpty} \]

4. **Prove the lemma** LenSF.

\[ \text{lenSF : Flen } xs = n \rightarrow \text{Flen } (x::xs) = S n \]

\[ \text{lenSF } p = \text{rewrite } p \text{ in Refl} \]