Summation, Flattening

Define a function \texttt{sum}, which computes the sum of elements of a list of natural numbers.

\[
\texttt{sum} :: \texttt{nat list} \Rightarrow \texttt{nat}
\]

Then, define a function \texttt{flatten} which flattens a list of lists by appending the member lists.

\[
\texttt{flatten} :: 'a list list \Rightarrow 'a list
\]

Test your functions by applying them to the following example lists:

\[
\text{lemma} \quad \texttt{sum \ [2::nat, 4, 8] = x}
\]
\[
\text{lemma} \quad \texttt{flatten \ [[2::nat, 3], \ [4, 5], \ [7, 9]] = x}
\]

Prove the following statements, or give a counterexample:

\[
\text{lemma} \quad \texttt{length (flatten xs) = sum (map length xs)}
\]
\[
\text{lemma} \quad \texttt{sum_append: sum (xs @ ys) = sum xs + sum ys}
\]
\[
\text{lemma} \quad \texttt{flatten_append: flatten (xs @ ys) = flatten xs @ flatten ys}
\]
\[
\text{lemma} \quad \texttt{rev (map rev (rev xs)) = rev (flatten xs)}
\]
\[
\text{lemma} \quad \texttt{rev (map rev xs) = rev (flatten xs)}
\]
\[
\text{lemma} \quad \texttt{list_all (list_all P) xs = list_all P (flatten xs)}
\]
\[
\text{lemma} \quad \texttt{sum (rev xs) = sum xs}
\]

Find a (non-trivial) predicate \texttt{P} which satisfies

\[
\text{lemma} \quad \texttt{list_all P xs \rightarrow length xs \leq sum xs}
\]

Define, by means of primitive recursion, a function \texttt{list_exists} which checks whether an element satisfying a given property is contained in the list:

\[
\texttt{list_exists} :: ('a \Rightarrow \texttt{bool}) \Rightarrow ('a list \Rightarrow \texttt{bool})
\]

Test your function on the following examples:

\[
\text{lemma} \quad \texttt{list_exists (\lambda n. n < 3) \ [4::nat, 3, 7] = b}
\]
\[
\text{lemma} \quad \texttt{list_exists (\lambda n. n < 4) \ [4::nat, 3, 7] = b}
\]

Prove the following statements:

\[
\text{lemma} \quad \texttt{list_exists_append:}
\]
"\text{list_exists } P \ (xs @ ys) = (\text{list_exists } P \ xs \lor \text{list_exists } P \ ys)"

\textbf{lemma} "\text{list_exists} (\text{list_exists } P) \ xs = \text{list_exists } P (\text{flatten } xs)"

You could have defined \texttt{list_exists} only with the aid of \texttt{list_all}. Do this now, i.e. define a function \texttt{list_exists2} and show that it is equivalent to \texttt{list_exists}. 