Required Problems

1. (a) Define a function \texttt{addFirstA} which takes a list of integers and returns a list in which each element is the sum of the first and corresponding elements of list, without using higher-order functions. For example:

\[
\text{addFirst\ [4,3,2,1] = [8,7,6,5]}
\]

(b) Repeat the problem in part a and write \texttt{addFirstB}, but you should use a higher-order function.

2. Define a function \texttt{commaSeparate :: [String] \to String} that takes a list of strings and returns a single string that contains the given strings in the order given, separated by ", ".

For example,

\[
\begin{align*}
\text{commaSeparate \ [] &= ""} \\
\text{commaSeparate \ ["a", "b"] &= "a, b"} \\
\text{commaSeparate \ ["Monday", "Tuesday", "Wednesday", "Thursday"] &= "Monday, Tuesday, Wednesday, Thursday"}
\end{align*}
\]

3. Write a function \texttt{deleteAll :: (Eq a) \Rightarrow a \to ([a] \to [a])} that takes an item (of a type that is an instance of the Eq class) and a list, and returns a list just like the argument list, but with the each occurrence of the item (if any) removed. For example.

\[
\begin{align*}
\text{deleteAll \ 1 \ [1, 2, 3, 2, 1, 2, 3, 2, 1] &= [2, 3, 2, 2, 3, 2]} \\
\text{deleteAll \ 4 \ [1, 2, 3, 2, 1, 2, 3, 2, 1] &= [1, 2, 3, 2, 1, 2, 3, 2, 1]} \\
\text{deleteAll \ 3 \ [1, 2, 3] &= [1, 2]}
\end{align*}
\]

4. Write a function \texttt{deleteSecond :: (Eq a) \Rightarrow a \to ([a] \to [a])} that takes an item (of a type that has an == function defined for it) and a list, and returns a list just like the argument list, but with the second occurrence of the item (if any) removed. For example.

\[
\begin{align*}
\text{deleteSecond \ 1 \ [1, 2, 3, 2, 1, 2, 3, 2, 1] &= [1, 2, 3, 2, 2, 3, 2, 1]} \\
\text{deleteSecond \ 4 \ [1, 2, 3, 2, 1, 2, 3, 2, 1] &= [1, 2, 3, 2, 1, 2, 3, 2, 1]} \\
\text{deleteSecond \ 3 \ [1, 2, 3] &= [1, 2, 3]}
\end{align*}
\]

5. Write a function \texttt{associated :: (Eq a) \Rightarrow a \to [(a,b)] \to [b]} which takes a value of some type (call this input x) and a list of tuples whose first element is of x's type. It should pull out all elements of list whose second tuple element are the same as x and return a list of these values.

For example:
associated 3 [(3,4), (5,7), (3,6), (9,3)] = [4, 6]
associated 2 [(1,a), (3,c), (2,b), (4,d)] = [b]
associated c (zip [c, c ..] [1, 2 ..]) = [1, 2 ..]