1. (a) Define a function \( \text{addFirstA} \) 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 \( \text{addFirstB} \), but you should use a higher-order function.

2. Define a function \( \text{commaSeparate} :: \{\text{String}\} \to \text{String} \) 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 \( \text{deleteAll} :: (\text{Eq a}) \Rightarrow a \to ([a] \to [a]) \) 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 each occurrence of the item (if any) removed. For example.
\[
\begin{align*}
\text{deleteAll} \ 1 \ [1, 2, 3, 2, 1, 2, 3, 2, 1] & = [2, 3, 2, 2, 3, 2] \\
\text{deleteAll} \ 4 \ [1, 2, 3, 2, 1, 2, 3, 2, 1] & = [1, 2, 3, 2, 1, 2, 3, 2, 1] \\
\text{deleteAll} \ 3 \ [1, 2, 3] & = [1, 2]
\end{align*}
\]

Do this (a) using a list comprehension, and (b) by writing out the recursion yourself. Submit both solutions (and please call the first one \( \text{aDeleteAll} \) and the second version \( \text{bDeleteAll} \), so you don’t have to put them in separate files).

4. Write a function \( \text{deleteSecond} :: (\text{Eq a}) \Rightarrow a \to ([a] \to [a]) \) 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 \( \text{associated} :: (\text{Eq a}) \Rightarrow a \to ([a,b]) \to [b] \) such that \( \text{associated} \ x \) pairs is the list, in order, of the second elements of pairs in pairs, whose rst element is equal to the argument \( x \). For example:
associated 3 [(3,4), (5,7), (3,6), (9,3)] = [4, 6]
associated 2 [(1,a), (2,b), (4,d)] = [b]
associated c (zip [c, c ..] [1, 2 ..]) = [1, 2 ..]