SML Practice.

Implement each of the following functions as one-line functions using `map`, `foldr`, and/or `foldl`. Note that if the function takes multiple arguments, you should do it as a curried function (instead of using a tuple).

1. `il2rl : turns an int list into a real list (e.g. [1,2,3] → [1.0,2.0,3.0])`
2. `squareList : takes an int list and returns a list of all the squares of the integers (e.g. [1,2,3,4] → [1,4,9,16])`
3. `sqSum : takes an int list and returns the sum of the squares of the integers in the list (e.g. [1,2,3,4] → 30)`
4. `dupList : takes list of any type and returns a new list where all the elements are duplicated (e.g. [1,2,3,4] → [1,1,2,2,3,3,4,4])`
5. `myLength : takes a list and counts the number of elements in the list (without using the built-in length function) (e.g. [1,2,3,4] → 4)`
6. `il2absrl : takes an integer list and returns a list containing the absolute value of each element as a real (e.g. [~1,2,3,~4] → [1.0,2.0,3.0,4.0])`
7. `countTrue : takes a function and a list and counts the number of elements in the list for which the function returns true (e.g. countTrue (fn x => x mod 2 = 0) [1,2,3,4,5,6] → 3)`
8. `max : return the max element in a list of integers (e.g. max [1,2,3,4] → 4)`
9. `member : takes an element and a list and returns true if the element is in the list and false otherwise (e.g. member 4 [1,2,3,4] → true)`
10. `pivot : takes an integer and a list of integers and returns two lists of integers where the first list contains all the elements in the original list that are less than or equal to the integer argument and the second list contains all the elements in the original list that are greater than the integer argument (e.g. pivot 3 [1,2,3,4,5,1,2,3,4,5] → ([1,2,3,1,2,3],[4,5,4,5])) (Note that the order of the elements in each list doesn’t matter as long as they are all there and in the right list.)`