

Comp160
Lab 9
Spring 2019

For this lab you'll look at three list-based functions each of which represent common list processing idioms.

Lab 9

The overall problem we want to solve is counting the number of strings in a list of strings that are longer than some given length. To do this we'll break the work into three sub-tasks: determine the length of every string in the list, compute a list containing on those strings that are long enough, then counting the length of a string. These subtasks are examples of the *map*, *filter*, and *reduce* idioms respectively. A map is function that computes something about each element in a list and produces a list containing those results. In this case, we're computing the lengths of every string in a list of strings. A filter selects out all the elements in a list that meet a particular criteria. In this case we're filtering out all the numbers that are greater than or equal to a given value. Finally, a reduce effectively reduces a list down to some other piece of information. In this case we're reducing the list down to it's length. A great number of problems can be rethought in terms of some combination of *map*, *filter*, and *reduce*. After this lab we'll talk more about the patterns inherent in each idiom.

Complete the following tasks in the order in which they are presented.

1. Write the data type definition for a list of strings.
2. Write out the template for functions acting on a list of strings.
3. Write out the data type definition for a list of PositiveNumbers.
4. Write out the template for functions acting on a list of PositiveNumbers.
5. Produce the signature, purpose, and header for each of these functions:
 - (a) *count-longer-than* - A function that takes a list of strings and a length and counts the number of strings that are at least as long as the given length.
 - (b) *length-map* - A function that takes a list of strings and computes the list containing the length of every string in the given list of strings.

- (c) *at-least-filter* - A function that takes a list of PositiveNumbers and a PositiveNumber and computes the list of all the numbers in the given list that are greater than or equal to the given PositiveNumber.
 - (d) *list-length-reduce* - A function that takes a list of PositiveNumbers and computes the length of the given list.
6. Write tests for all of the functions listed above.
 7. Complete the definition for the functions listed above. The function *count-longer-than* should first use the map, then the filter, then the reduce functions to solve the problem. It does not use the list template.