Name: _____

CS380: Modern Functional Programming
Prof. Richard Eisenberg
Spring 2017

Practice Exam 2

This exam has two parts. The first part, attached to this cover sheet, contains several questions (worth a total of 30 points) that are to be completed without the use of a computer. You will hand this handwritten part into the folder provided before starting the computer work. You may not retrieve the handwritten part once you have submitted it. The second part, distributed separately, describes problems to be completed on the computer. Download the `Exam2.hs` file from the course syllabus page and put your answers in that file. Do not change the name of the file. Do not make any other files. When you are done, upload the file to Gradescope.

This entire exam is open-book/open-note. You may use any printed resources you like. You may *not* use any computing devices, such as laptops, phones, or calculators on this part. On the computerized section, you may use any websites you like, but you may *not* communicate with others.

I certify that my responses in this examination are solely the product of my own work and that I have fully abided by the Bryn Mawr College Academic Integrity policy and instructions stated above while taking this exam. (Sign before handing in the first section.)

Signature: _____

Printed Name: _____

1. Write a function *merge* that takes two *Vec*s (which we assume are in sorted order) and produces a sorted *Vec* combining them. The *Vec*s should be able to store any type that allows for comparisons.

$$\begin{aligned}
&\textit{merge}\ [3,7,8]\ [1,5] &&== [1,3,5,7,8]\\
&\textit{merge}\ [\,]\ \ \ \ [2,5,8] &&== [2,5,8]\\
&\textit{merge}\ [5]\ \ \ \ [1,2] &&== [1,2,5]\\
&\textit{merge}\ [7,8]\ \ [\,]\ \ \ \ &&== [7,8]\\
&\textit{merge}\ [\,]\ \ \ \ [\,]\ \ \ \ &&== [\,]
\end{aligned}$$

2. Write a function *runningSum* that computes the partial sums of a *Vec* of *Int*s. That is, each element in the output should be the sum of the numbers that came before that location in the input.

$$\begin{aligned}
&\textit{runningSum}\ [1,2,3,4] &&== [1,3,6,10]\\
&\textit{runningSum}\ [5,5]\ \ \ \ &&== [5,10]\\
&\textit{runningSum}\ [\,]\ \ \ \ &&== [\,]
\end{aligned}$$

3. Write a function *mins* that takes a *VecList* of *Vec*s of *Int*s and returns a *Vec* of the minima of each *vec*. If a *Vec* is empty, the minimum of that *Vec* is 0.

$$\begin{aligned}
&\textit{mins}\ [[3,8],[9,2,7],[\,],[6,1,2]] &&== [3,2,0,1]\\
&\textit{mins}\ [\,] &&== [\,]\\
&\textit{mins}\ [[\,],\ \ \ [\,]] &&== [0,0]\\
&\textit{mins}\ [[7,8],[1,2,3]] &&== [7,1]
\end{aligned}$$

4. Write a function *wither* that takes a *Vec* and removes every second element, starting with the first. The element type can be anything.

$$\begin{aligned}
&\textit{wither}\ [1,2,3,4,5] &&== [2,4]\\
&\textit{wither}\ [1,2,3,4,5,6] &&== [2,4,6]\\
&\textit{wither}\ [1,2] &&== [2]\\
&\textit{wither}\ [1] &&== [\,]\\
&\textit{wither}\ [\,] &&== [\,]
\end{aligned}$$

5. Write a function *repeater* that looks for a repeated element in a *Vec*. The element type can be any type that supports equality comparisons. If it finds an element that occurs in the list twice in a row, the function returns *Just* the index of the first element, as a *Fin*. If there are no repeaters, then *repeater* returns *Nothing*. (That is, the return type is *Maybe* (*Fin* ???), for your choice of ???.)

$$
\begin{aligned}
&\textit{repeater}\ [5,3,1,2,2,8] &&== \textit{Just}\ 3\\
&\textit{repeater}\ [8,2,2,0,6,6,7] &&== \textit{Just}\ 1\\
&\textit{repeater}\ [1,2,3,1,2,3] &&== \textit{Nothing}\\
&\textit{repeater}\ [8,8] &&== \textit{Just}\ 0\\
&\textit{repeater}\ [9] &&== \textit{Nothing}\\
&\textit{repeater}\ [\,] &&== \textit{Nothing}
\end{aligned}
$$