COS301
University of Maine
Spring 2022

Homework Assignment #4
Assignment Due: Friday, April 29, 2022, by 11:59pm

Grading: *To avoid surprises, refer to the syllabus about how assignments are graded and about the limited possibilities to submit corrections.*

Silent Policy: *A silent policy will take effect at 6pm on the due date. This means that no questions about this assignment will be answered after this time before the submission deadline, not matter how or where it is asked. Questions will be answered after the submission deadline.*

Late Policy: A 'grace extension' system is in effect, see the course syllabus for details. If you have a documented medical condition, please inform your instructor **before** the assignment is due.

How to Submit: Submit your solutions electronically on Brightspace by the deadline as a single **PDF file**. Make sure the file is not corrupted. In case we cannot open the file, your assignment only counts as submitted at the time we receive a workable file.

Make sure you show your work and explain/justify all answers as requested.

We encourage the use of LaTeX to produce your assignment solutions. A good tool to use is `overleaf.com`, it allows you to write LaTeX documents and compile them into a PDF without having to install LaTeX locally.

You may alternatively use Google Docs, Microsoft Word or a similar word-processing system but make sure you use correct notation throughout.

For Questions:

Questions concerning the assignment should be directed to the TA, Sepideh Neshatfar, or the instructor either in person (during office hours) or by email (sepideh.neshatfar@maine.edu, torsten.hahmann@maine.edu). Please place "[" + COS301++ "]" and "A4" in your email subject header.

Important corrections (hopefully few or none) and clarifications to the assignment questions will be posted as announcements on Brightspace. You are responsible for monitoring Brightspace.

Policy on Group Work:

While you may discuss questions and approaches with others, all submitted answers must be your own work, written in isolation, without help from others. See the course information sheet for more details of what kind of collaboration is acceptable. If in doubt, talk to the instructor. **Be sure to acknowledge all sources of help!** Refer to the course syllabus for more detailed instructions and the course policy on academic honesty.

General instructions:

- **You may work together with one other student (but no more than that), but everyone must submit a solution**
- Make sure you have the Haskell compiler and cabal installed, follow the instructions here: https://www.haskell.org/downloads/
- Afterwards, install the cassava library for processing CSV files using Cabal by executing the following on your command line: `cabal install --lib cassava`
- Similarly, install and expose other libraries needed: e.g. vector, bytestring
  `cabal install --lib vector`
- Start your Haskell interpreter `ghci`, then call
  `:set -XOverloadedStrings`
  `:set -XRecordWildCards`
- Load the provided Haskell module `weatherstats.hs` inside your Haskell interpreter. You can do this by calling `:l weatherstats.hs`
- Make sure the module works as intended, by calling `jan3Minimum "2069-2021.csv"`
- If it doesn't give you the answer `minimum temperature on January 3 = -7.8`, make sure the csv file is correctly formatted. To test that, you can call `decodeItemsFromFile "2069-2021.csv"` for any parsing errors.
- Stick to the provided template `weatherstats-template.hs`, do not edit/add code above the line: `-- DO NOT CHANGE ANYTHING ABOVE THIS LINE --`

Question 1. (24 points) Basic functional programming in Haskell

(a) (6 points) Use the higher-order function `foldl1` to write a function called `average` that computes the average of a list of Floats. Make sure you include a type declaration for the function.

(b) (6 points) Write a function `maxDiff` that computes the differential between the highest and lowest value in a list of Floats. Again, make sure you include a type declaration.

(c) (12 points) Use the two functions you defined to write another function `daySummary day` that computes the average and maxDiff for the temperature and the windSpeed on a particular day of the year from the data provided by the `readData` function (for simplicity, the day is given as Integer). The output should be: (day, average temperature, max temperature differential, average windspeed, max windspeed differential). Use the following example as inspiration: `let result = minimum (map temp (take 24 (drop 72 obsData)))`. *Note that the function `temp` is provided but you need a similar function to access the windSpeed.*

Question 2. (10 points) List processing in Haskell

(a) (8 points) Write a function `chunkby n l` that takes a list `l` and an integer `n` and produces a list of lists of length n. Make sure you include a type declaration. *Use the provided function* `dailyTemperatureStat` *as inspiration. You will also need to use the ":" function to append things to lists.*

(b) (2 points) Define a partial function `chunkByDays l` that relies on `chunkby` and chunks the list of hourly observations (as provided by the `readData` function) into a list of lists of observations for each day.

Question 3. (20 points) Data processing in Haskell

(a) (6 points) Consider the function `dailyTemperatureStat`. Provide its type declaration (keep it as generic as possible!) and explain in a comment line what it does.

(b) (6 points) Use the function `dailyTemperatureStat filename` to define a new function `allMinimumTemps` that produces a list of tuples consisting of the day of the year and the minimum temperature on that day. *You will need to use `map` and function composition (using the "." operation), but be careful because we're operating on a list of lists.*

(c) (8) Write a function `highDifferentialDays filename` that finds all days with a temperature differential (from Question 1b) of over 15 degrees and that provides their temperature differential and average temperature (from Question 1a). It should output (on the command line) a list of tuples (day, temperature difference, average temperature). Use the function `jan3Minimum` as a template for how the function should be structured.

(d) **Bonus 4 points**: Adjust the function `highDifferentialDays` so that it throws out any days with obvious sensor errors, namely when the temperature is recorded as -99.9 at some time during the day.

Have fun and good luck!