Exercise4

Attached Files:

- Exercise4.zip (5.275 KB)
- makex4-v2.zip (9.52 MB)

Topic: "Higher"-Order Programming

Instructions:

- Download M**akex4.zip**, uncompress (produces M**akex4** folder, and use your command-line/terminal window inside that folder to issue the command: **stack install**. If that succeeds, then you will have installed an executable called **makex4**.
- Go to the folder (FPCW say) where you put Exercise1. It should still have the **makex.conf** file left over from that exercise. Download and uncompress **Exercise4.zip** here.
- Execute the following in FPCW: **makex4**. You should not need to re-enter any details. It should create file **Exercise4/src/Ex4.hs**.
- Enter Exercise3 and give the command **stack install**, which will install an executable called **ex4**.
- Running **ex4** will result in a ***long*** message, which describes Q3 (see below)
- Your task is to edit both **Ex4.hs** and Main.hs so that it produces the correct output - this is defined in
  - the comments in **Ex4.hs**.
  - the output produced by **ex4** when it runs
- You are reccommended to use **stack ghci src/Ex4.hs** to observe and debug your **Ex4.hs** code. You can re-compile your code using **stack install**, and the re-enter ghci as above
- You can use **stack ghci src/Main.hs** to observe and debug it also.
- Your submission is BOTH the file **Ex4.hs** and **Main.hs** as is (do not compress/tar/package them up in any way)

This exercise supplies you with the tailored partial expression data-type from Exercise3, and asks you to:

- Write Function **mdeval** which is like meval from Exercise 3 but uses monads for error handling
- Solve a "puzzle" based on abstracting out a common pattern of behaviour from list processing functions.
- Modify the mainline program to read numbers from one file, process them and write results to another file

Q&A

Q1: My program says it can't find **input.dat**

A1: You need to run **ex4** in from a folder that contains **input.dat**. The **ex4** program should be runnable from any directory.

Q2: Can I import **System.IO** into my **Main.hs**?

A2: Yes. It's not necessary, but it gives you more options for how you program your mainline. (Keep It Simple though!)

Q3: Where is the file "input.dat" stored in the test environment, from which file path should it be read? And where should the "output.dat" file be stored?

A3: You need to provide your own version of "input.dat" in the folder from which you run **ex4**. It will create/overwrite "output.dat" on the same directory.

Q4: It is an issue if my output file would have the last line being blank?

A4: No, it won't - that is easily handled.

Q5: Where I should put the extra helper functions I created for Exercise 4, Q3?

Assuming both options work fine, is it okay for them to be located in Ex4.hs, or should I move them to Main.hs?

A5: if the code compiles and runs you will be fine. Note that helper functions defined in Main.hs are not visible in Ex4.hs.