COMP1711 - week7

Files

This repository contains several files: - blood_iron.c - utilities.h

These two files are the code - blood_iron.c is the sourcecode and contains the main() function which will run when you run the code. utilities.h is a headerfile containing some functions. You will be adding more functions into this section.

Both of these files can be edited as much as you like, but it is recommended that you do not edit the struct as it holds the data in useful formats.

There are additionally: - data.txt : A file containing some blood iron data - bad_data.txt : A file which has a missing piece of data

You are welcome to create additional files, for example introduce some different errors to a file.

How to compile

You have 2 files to compile, so you can use the command:

```
gcc utilities.h blood_iron.c -o ironCalc
```

And then run with:

```
./ironCalc
```

Tasks

You will not finish all of these - pick 2 or 3 to focus on!

Before you start these additional tasks, you should: - check that the code can read in a file to an array - try and add some validation so that it rejects 'bad' data such as bad_data.tx where some data is missing

BRONZE tasks

Do not reopen the file in any of these tasks - you should be using the data from the array.

If you don't know how to do this, please ask - this is a skill you will need for your assignment.

A: View all

After the user has entered a valid file and it has been read in, write the data out from the array in a neat format.

B: Mean average

After the user has entered a valid file and it has been read in, calculate the mean average of the blood iron readings. You should print this out to a suitable number of decimal places.

SILVER

C: Highest

Find the reading with the highest blood iron level, and print it out for the user. Hint: a loop will be required.

D: Lowest

Find the reading with the lowest blood iron level, and print it out for the user.

This will probably be quite similar code to finding the highest!

E: View specific month

You may want to use the `<string.h>` function `strstr()` for this. You can use the `man` command in the terminal to see more for this.

You should: - ask the user for the month (in the same format as in the data file) - check each date string to see if there is a match - print out the record if there is.

GOLD Tasks

If you want to try the GOLD challenges, you can find the specifications here.

F: See some additional statistics about your iron levels

This should show the user some different metrics: - The range of their iron levels. - The standard deviation of their iron level. - Their median iron level. - The mean iron level for each month in the data file.

These should still be relatively simple, but you may need to use the math library which is already imported.

To compile with the math library you need to include `-lm` at the end of your compilation line!

```
gcc utilities.h code.c -o ironCalc -lm
```

G: Generate a graph of your iron levels

This is a tougher challenge and might take some time!

You have several options here:

Basic text Files

you could create a basic graph using a text file with symbols representing the axis and data. This is probably more time consuming and difficult than you expect it to be :).

HTML/JavaScript

You could use C to generate you some JavaScript code to make your graphs. You can find a W3Schools guide to one of the many different JS plotting libraries here, or you can find more guides online if you search for 'HTML graphs'. You will need to use fprintf() to write code into a HTML file.

Matplotlib

If you are familiar with Python, you could also create your plots using the matplotlib library - again, you would need to use your C program to analyse the data and then produce the python code which could be run to produce the graph (i.e. don't just re-do the exercise in Python :)).