Forritun
Háskólinn í Reykjavík Haustönn 2018

Skilaverkefni 6 / Assignment 6

Background
The Centers for Disease Control (CDC) keeps data on what it calls "Winnable Battle Risk Factors and Health Indicators". These are negative behaviors or incidents that could be avoided by changes in lifestyle. In this project, you will examine some CDC data to discover the US states with the best and worst records in regard to a few of these risk factors and health indicators.

Project Specifications
The file "riskFactors.csv" lists data on 20 different risk factors and health indicators for each state. The data is in "comma separated value" format (csv), which means that each entry is separated from the others by a comma. Examining the file in a text editor or a spreadsheet program or both should help you understand the format.

Since analyzing 20 different indicators can be a bit confusing, we will only look at five: *Heart Disease Death Rate*, *Motor Vehicle Death Rate*, *Teen Birth Rate*, *Adult Smoking*, and *Adult Obesity*. Your program will read in the data from the csv file and find the states with the best and worst record for each of these indicators (largest and smallest values). It will then print the states that have the highest and lowest value for each of the indicators, along with their values.

Your program must be general enough to work with a similar file with states as rows and with the same column headers. That is, if the CDC puts out a new file with different values in the cells (e.g. maybe new research changed some values), your program will work correctly.

Additional Requirements

1. You are only allowed to use concepts/constructs that are covered in the first 8 chapters in the textbook.

2. Prompt for the input file. Your program must check if the file exists. If it does not, your program should output "Cannot find file X" (where X is the name of the file), and only print the header (the first two lines) in the output below.

3. Your program must format the output into columns (see below). You have to match our formatting exactly, columns should line up and it should be readable. Use string formatting. The header consists of the line starting with "Indicator …" and the line with the dashes. The length of each line is 87 characters. There are 6 columns in the output:

   a. The first one ("Indicator") is of size 33, left justified.
   b. The second one (the "Min" state) is of size 21, left justified.
   c. The third one (the "Min" value) is of size 6, right justified.
   d. The forth one (empty space) is of size 6.
   e. The fifth one (the "Max" state) is of size 15, left justified.
   f. The sixth one (the "Max" value) is of size 6, right justified.

Output:

```
Indicator                        Min                              Max
---------------------------------------------------------------------------------------
Heart Disease Death Rate (2007): Minnesota             129.8      Mississippi     266.5
Motor Vehicle Death Rate (2009): District of Columbia    4.8      Wyoming          24.6
Teen Birth Rate (2009):          New Hampshire          16.4      Mississippi      64.2
Adult Smoking (2010):            Utah                    9.1      West Virginia    26.8
Adult Obesity (2010):            Colorado               21.4      Mississippi      34.5
```

Notes

1. The first line in the input file contains a header describing each column.
2. You need to convert strings to numbers where appropriate.
3. Watch out for the fact that some data has a % sign.
4. Depending on how you design your program you may find the following list functions useful: `min,max`.
5. There exists a module for reading csv files, but you are not allowed to use it in this project.
6. It should be clear that you need to apply divide-and-conquer in this project! Make sure that all your functions are short, including the main program.

Forritun
Háskólinn í Reykjavík Haustönn 2018

Project 4

Write a program that first asks the user to input numbers of shares and then price information (price of each share) as whole-dollar and a fraction (numerator and denominator). For example, the price string “29 7 8” stands for the price 29 7/8 or 29,875. User input needs to be error checked, i.e. making sure it only contains digits. Use *try-except* constructs for handling input errors.

The market price should then be output with two digits after the decimal point. The user should be able to repeat this as often as he/she wants.

In your solution, you should only use methods/material discussed in the first six chapters in the textbook.

Decompose the project into subtasks. With the exception of the loop, your main program should just be a series of function calls. You need to figure out yourself which functions are needed. Make sure that your implementation does not contain repeated code or at least that repeated code is very minimal.

Example input/output:

```
Enter number of shares: 100
Enter price (dollars, numerator, denominator): 29 a b
Invalid price!
Enter price (dollars, numerator, denominator): 29 7 8
100 shares with market price 29 7/8 have value $2987.50
Continue: y
Enter number of shares: 1a
Invalid number!
Enter number of shares: 30
Enter price (dollars, numerator, denominator): x 1 2
Invalid price!
Enter price (dollars, numerator, denominator): 89 1 2
30 shares with market price 89 1/2 have value $2685.00
Continue: n
```