CS160 Computer Science I
Project 2
Spring 2022

Assignment

This project will store the daily high and low temperatures in Grand Forks from a month. To test this project I some found monthly data after a basic search. I'm guessing the data is correct, I found it on the Internet, so it must be right. 😊 The actual site I used for the test data is https://www.wunderground.com/history/monthly/us/nd/grand-forks/KGFK/date/2022-3.

This project will have you read in the data, store the temperature data in two separate lists (one for the high temps and one for the low temps), and then do some basic calculations. Some of the required functions will require both lists, some will only require a single list and will work with either list.

Write each of the required functions. The function header must be implemented **exactly** as specified. You can change the name of the parameters, but not the order or functionality of any parameters.

If there are built-in functions that can perform some or part of assigned tasks, you cannot use them. You can write any additional functions beyond the require functions for use within the program.

The data file will contain one line that contains the name of the month and year, and then one line for each day of the month. This line will be the first line of the data file. You do not need to do anything with the first line of the data file other than read the data and then display it in the output.

Following the first line there could be between 28 and 31 additional lines of data, depending upon the month. The lines after the initial line will contain the day, the high temperature for that day, and the low temperature for that day, all separated by commas.

A sample of the data file is:

```
March 2022
1,18,7
2,17,-8
3,12,-11
4,23,9
...
30,32,22
31,37,22
```

Required Functions

`readTempData (fileName)` – creates two lists, one for the high daily temps and one for the low daily temps. It then read the data file, filling the two lists with the data from the file. This function returns a tuple of the initial line (the month and year information), and the two list (first the high temps list followed by the low temps list).

`def dailyHigh (highTemps, date)` – if `date` is a valid day this function returns the high temperature for that date. If `date` is invalid, this function returns `None.`

`def dailyLow (lowTemps, date)` – if `date` is a valid day this function returns the high temperature for that date. If `date` is invalid, this function returns `None.`

`def biggestDailyDifference (highTemps, lowTemps)` – returns the largest difference between the daily high and low temps. Given the data above this would return 25.

`def dayOfBiggestDailyDifference (highTemps, lowTemps)` – returns **the day** of largest difference between the daily high and low temps. Given the data above this would return 2.

`def smallestDailyDifference (highTemps, lowTemps)` – returns the smallest difference between the daily high and low temps. Given the data above this would return 10.

`def dayOfSmallestDailyDifference (highTemps, lowTemps)` – returns **the day** of smallest difference between the daily high and low temps. Given the data above this would return 30.

`def monthlyAverage (temps)` – returns the average of the monthly temperatures. This function should work with either list. DO NOT use any built-in functions are achieve this functionality. Return `None` if unable to calculate an average.

`def biggestDifferenceBetweenDays (temps)` – returns the first day of the pair of days with the largest difference between days. This function should work with either list. Keep in mind the difference can be positive negative between successive days. Given the very limited sample data above this function would return 3 if passed the list with high temperatures, as days 3 and 4 had the largest difference between their high temps.

`def printTemps (monthYearInfo, highTemps, LowTemps)` – this function DOES create output on the display. Print out month and year information, followed by the day, high temp, and low temp to a table. Ensure that the days are left justified and the high and low temps are right justified. Print an appropriate label above each column. Do not return anything from the function.

`def main()` - Tests each of your functions.