

CptS 122 - Data Structures



Programming Assignment 1: Analyzing Fitbit Data

Assigned: Friday, August 24, 2018

Due: Wednesday, September 5, 2018, by midnight

I. Learner Objectives:

At the conclusion of this programming assignment, participants should be able to:

- Analyze a basic set of requirements for a problem
- Apply basic techniques for data cleansing or data cleaning
- Analyze a medium set of data and compute results
- Filter a dataset
- Implement deduping methods
- Compose C language programs
- Create and compile a program using Microsoft Visual Studio 2015
- Execute a program in Microsoft Visual Studio 2015
- Create basic test cases for a program
- Apply arrays, strings, and pointers
- Define and apply structs
- Parse strings from files
- Convert strings to numerical values
- Summarize differences between array notation and pointer notation
- Apply basic string handling library functions

II. Prerequisites:

Before starting this programming assignment, participants should be able to:

- Access Microsoft Visual Studio 2015 Integrated Development Environment (IDE)
- Design and develop a small or medium sized program in any language
- Apply sequential, conditional, and iterative constructs
- Design and implement user-defined functions

III. Overview & Requirements:

The intent of this assignment is to review concepts from your prior “CS 1” course and to challenge and enhance those concepts.

Fitbit is a company that builds wearable technology devices that track various activities. The devices have sensors that measure number of steps and distance walked, heart rate, sleep quality, floors climbed, and calories burned. In this assignment, you will analyze data that was generated from a Fitbit devices. The data is stored in a comma-separated values (.csv) file that you will find at: <http://eecs.wsu.edu/~aofallon/cpts122/progassignments/FitbitData.csv>. The entries in this file were merged from two different devices. You will need to *filter* any data that is not related to the target patient. The first data entry in the file contains the target. You will also need to *dedupe* any entries that appear multiple times and perform *data cleansing* any entries that have missing fields. A .csv file stores data as plaintext in tabular form. Each row in the file is considered a *record*. Each record consists of *fields* separated by commas.

In particular, you will analyze 24 hours of data. Each record in the “FitbitData.csv” represents one minute of data and consists of eight fields. These include the following:

1. Patient ID
2. Minute
3. Calories
4. Distance (in miles)
5. Floors
6. Heartrate
7. Steps

8. Sleep level

What data structures are required?

In this assignment, you must define a C struct to store a subset of the Fitbit data fields as follows:

```
typedef struct fitbit
{
    char patient[10];
    char minute[9];
    double calories;
    double distance;
    unsigned int floors;
    unsigned int heartRate;
    unsigned int steps;
    Sleep sleepLevel;
} FitbitData;
```

The type Sleep is enumerated and must be defined as follows:

```
typedef enum sleep
{
    NONE = 0, ASLEEP = 1, AWAKE = 2, REALLYAWAKE = 3
} Sleep;
```

You must also define an array of `FitbitData` that can store 24 hours of minute data. Hence, you must declare an array of size 1440. You have the freedom to decide on other data structures and variables that you need for the assignment.

What are the other requirements?

This program does not require any user input! However, you will need to display some results to the screen!

- You must open "FitbitData.csv" for mode read; check for success
- You must read each record in the file as a *string*, one line at a time; if the record does not belong to the *target* patient, then it should be discarded
- You must parse each record into the corresponding fields, and store into the `FitbitData` array; note: not all fields have values, some are "empty" or null; if some of the fields are "empty" or null, then you must perform *data cleansing* and insert values to construct a record that is consistent with the others; the data inserted should not represent *valid* values
- You must compute the total calories burned, distance walked in miles, floors walked, and steps taken
- You must compute average heartrate over the 24 hour period
- You must report the max steps taken in one minute over the 24 hour period; if there are multiple minutes throughout the day where the max is discovered, then report the one that is the latest in the 24 hour period
- You must report the longest consecutive range of poor sleep; a range is defined as one or more consecutive minutes where the `sleepLevel > 1`; the poorest sleep is not based on the length of the range, but the sum of the sleep levels in the range; the max sum of the ranges is considered the poorest sleep (report the starting and ending minutes of range)
- You must open "Results.csv" for mode write; this will either create a new .csv or overwrite an existing one
- You must output two lines to "Results.csv" and to the screen in the following format:
 - o Line 1: Total Calories,Total Distance,Total Floors,Total Steps,Avg Heartrate,Max Steps,Sleep
 - o Line 2:
 - valueCalories,valueDistance,valueFloors,valueSteps,valueHeartrate,valueMax,valueSleepStart:valueSleepEnd
 - o Lines 3 - N: the filtered, deduped, and cleansed data in the original FitbitData.csv format
- You must close "FitbitData.csv" and "Results.csv"

IV. Submitting Assignments:

1. Using the OSBLE+ MS VS plugin, please submit your solution. Please visit <https://github.com/WSU-HELPLAB/OSBLE/wiki/Submitting-an-Assignment> for more information about submitting using OSBLE+.
2. Your project must contain at least one header file (a .h file), two C source files (which must be .c files), and a local copy of the .csv file.
3. Your projects must build properly. The most points an assignment can receive if it does not build properly is 65 out of 100.

V. Grading Guidelines:

This assignment is worth 100 points. Your assignment will be evaluated based on a successful compilation and adherence to the program requirements. We will grade according to the following criteria:

- 🐾 5 pts for correctly defining struct `FitbitData`
- 🐾 3 pts for correctly defining enum `Sleep`
- 🐾 3 pts for correctly opening “FitbitData.csv” for mode read
- 🐾 2 pts for correctly checking the success of the file opening operation
- 🐾 10 pts for correctly reading each record in the file as a *string*, one line at a time, and *filtering* and *deduping* any lines that do not belong to the *target* patient or are duplicates
- 🐾 15 pts for correctly *cleaning* the data and parsing each record into the corresponding fields
- 🐾 5 pts for correctly converting the strings to the field types
- 🐾 5 pts for correctly storing the fields into the `FitbitData` array
- 🐾 3 pts for correctly computing the total calories burned
- 🐾 3 pts for correctly computing the total distance walked in miles
- 🐾 3 pts for correctly computing the total floors walked
- 🐾 3 pts for correctly computing the total steps taken
- 🐾 4 pts for correctly computing the average heartrate over the 24 hour period
- 🐾 8 pts for correctly determining the max steps taken in one minute over the 24 hour period
- 🐾 8 pts for correctly determining the longest consecutive range of poor sleep
- 🐾 3 pts for correctly opening “Results.csv” for mode write
- 🐾 8 pts for correctly formatting and outputting the data, without any duplicate entries, to “Results.csv” and to the screen
- 🐾 2 pts for correctly closing the “FitbitData.csv”
- 🐾 2 pts for correctly closing the “Results.csv”
- 🐾 5 pts for appropriate top-down design of functions and good style