CS2123 Data Structures II Summer 2017

Assignment 1: C refresher

Due 6/16/17 by 11:59pm

1. Student ADT

Suppose we have an ADT specification for the data structure *student*. The values associated with each student are:

- id (integer)
- name (string)
- age (integer)
- gpa (float)

A valid student ADT requires that id must be > 0 and name must be a string of at least 1 character.

There are 3 operators:

- Make Student, a data type definition (i.e., constructor) which takes id and name as inputs. This operator assumes that id > 0 and name has at least 1 character.
- Set Age, which takes student and age as inputs, and if age is > 0, assigns it to the students age member.
- Set GPA, which takes student and gpa as inputs, and if gpa is between 0.0 and 4.0 inclusive then assigns it to the students gpa member.

2. C refresher (14 points)

Following the ADT given above, you will be provided a data file containing a number of student data sets. Each data set in the file has an integer, a string, another integer, and a float. You must dynamically allocate an array of student structs that will store each student data set read from the file. After all student structs are stored, you will compute the total # of students, an average age, an average GPA. You must use pointer arithmetic to traverse your student array, not subscripting. For example, an array of students may be dynamically created (assuming the # of students is stored in variable numStudents) using a statement such as:

```
student * students = (student *) malloc(sizeof(student) * numStudents);
```

Using pointer arithmetic, the n^{th} student may be accessed as:

```
*(students + (n -1))
```

You must define a struct called student that has 4 members: id, name, age and gpa. Use data types that are appropriate to those defined in your ADT. For string, use a statically allocated char array of size 255.

Continued on the back ↪

Your program should first print the following to stdout on its own line:

CS 2123 Assignment 1 written by <your name>.

Your program must read in the data sets, storing the values of each data set into a new student struct variable. If any data violate the ADTs conditions, your program should print an appropriate error message and skip that record in the file. Your error message should look something like:

Cannot create student record from file row <row number>: id is invalid

To build the student record, first create the student struct variable by calling a functional implementation of "Make Student" ADT operator (passing and validating both id and name). Then call a function based on your Set Age operator and lastly call a function based on your Set GPA operator. The final result will be a complete student record that can be saved in your array. Because you are not given the total # of students initially, you can either resize your dynamic array as you create each student, or you can read through the data file once to determine the # of students, and then a second time to read the actual records.

Finally, calculate and print to stdout (each on its own line) the total # of students that were successfully created from the file, average age (as a float or double), and average GPA (as a float or double). You can calculate each in separate traversals or combine them all into one traversal. You must use pointer arithmetic in the array traversals instead of array subscripting (i.e., dont use square brackets to access array elements). Your output should look like the following:

# of students: <your result for # of students>
Average age: <your result for average age>
Average gpa: <your result for average gpa>

To receive full credit, your code must compile and execute. You should use valgrind to ensure that you do not have any memory leaks.

A test data file for you to use is located on Blackboard under Assignment 1.

Deliverables:

Your solution should be submitted as one or more C source code files. You must submit all of the C source that you produce for this assignment problem so that it can be compiled and executed.

Archive (e.g., zip) all of your files and upload them to Blackboard under Assignment 1.

Remember:

The program you submit should be the work of only you. Cheating will be reported to judicial affairs. Both the copier and copiee will be held responsible.