ASSIGNMENT 7

GENE-121-2017-3F-ASSN-07-Description

Fall 2017

CONTENTS

INTRODUCTION

This assignment is to be completed in pairs.

The problem is this assignment is an exercise to:

- explore different ways to storing the same data using arrays
- pass arrays and files between functions
- coordinate multiple functions between you and your programming partner

QUESTION 1 – PARKING SERVICES

You and your partner have been hired to write software to help UW Parking Services keep their records updated for their special MME parking lot. This special lot has 50 parking spaces, numbered from 1 to 50. They have recently introduced a rule that students getting a new parking space are only permitted in spaces 26 through 50. Staff can be given a parking space in any of the 50 spaces. Note that any existing students that have a parking spot from 1 to 25 are grandfathered and should not be moved to a higher-numbered space.

Parking Services has tracked information for parking space assignments in three files:

- parking_current.txt: This file contains the parking space assignments.
- parking_remove.txt: This file contains a list of people that no longer need a parking space.
- parking_add.txt: This file contains a list of people that need a parking space and don't yet have one.

Your job is to start with the current list of parking space assignments, free up unneeded spaces, move staff to preferred spaces, and finally add new parking space assignments. Output your updated list to the file list parking_updated.txt.

PARKING_CURRENT.TXT

Each line of this file contains the details of one assigned space. The data fields for each line are:

- Student/staff: 0 indicates a student and 1 indicates staff.
- Full name: last and first name joined by underscore.
- Parking space number

The following shows the first four lines of the file:

```
1 Hulls_Carol 11
0 Rochlitz_Alexandra 5
0 Barakat_Abdullah 26
1 Bedi_Sanjeev 12
```

PARKING_REMOVE.TXT

Each line of this file contains:

- Student/staff: 0 indicates a student and 1 indicates staff.
- Full name: last and first name joined by underscore

PARKING_ADD.TXT

Each line of this file contains:

- Student/staff: 0 indicates a student and 1 indicates staff.
- Full name: last and first name joined by underscore

DIVISION OF LABOR

Certain portions of this exercise are designated for one group member (Partner A) and some other portions of the exercise are designed for the other group member (Partner B). You need to decide which one of you is Partner A and who is Partner B.

PART (A) – ARRAY DESIGN [BOTH PARTNERS]

One of the critical design decisions for this problem is how you will store the data in the program. In this program, you will have to store at least some of the data in arrays. How you choose to structure your arrays will either make the rest of your code relatively straightforward, or it will make your coding exercise very difficult.

In this part, you will need to decide with your partner how you will store the data from the three files. You will need some understanding of the overall problem before making these design decisions, so make sure you read over the entire problem description before having this discussion.

Some key questions to consider include:

- How would you keep track of this information if you were doing this on paper (for the Parking Services office)?
- How many arrays will you use?
- What are the data types of each array?
- What are the sizes of each array?
- What are the names of each array?
- Once you have stored information in your arrays, how can someone (or your program) tell if a particular parking space is available or not?

You should come up with at least two different ways you could store the data and pick the one which you feel makes the most sense for this application.

Once you have decided upon your array design, sketch it out on paper. Fill in your sketches with the information from the first three lines of each file. Submit your work from this part in handwriting with the rest of your assignment package.

You are your partner must agree on function prototypes and array structure (data type, dimensions, names) before proceeding.

PART (B) – FUNCTION TO READ CURRENT PARKING FILE [PARTNER A]

Write a function that receives the already opened parking_current.txt file and reads the data into the appropriate parking data array(s) as described in part (a).

PART (C) – FUNCTION TO READ REMOVE AND ADD FILES [PARTNER B]

Write a single function that receives an already opened input file of people to be added or removed (parking_remove.txt or parking_add.txt). The function then reads the data from the file into appropriate arrays as described from part (a). This function will then return the number of people in the file.

Note that this is a single function that can read either the parking_remove.txt file or the parking_add.txt, as long as it is passed the correct opened file and the appropriate array(s).

PART (D) – FUNCTION TO FREE UP A SPACE [PARTNER B]

Write a function that receives the array(s) of parking data, one person's name, and (if needed) their student/staff status. The function will modify the parking data so that the parking space for this one person is freed up.

PART (E) – FUNCTION TO FIND THE FIRST FREE SPACE [PARTNER A]

Write a function that receives the array(s) of parking data and a student/staff status. The function will return an integer indicating the available parking space with the lowest number. Note that the function also needs to return a value if the parking lot is full and there are no more available parking spaces.

PART (F) – FUNCTION TO ASSIGN A SPACE [PARTNER B]

Write a function that receives the array(s) of parking data, one person's name, and their student/staff status. The function should assign this person the best (lower number) parking space. If the person is successfully assigned a space, this function returns true. Otherwise, the function returns false.

This function should use (call) the function from part (e).

PART (G) – OPTIMIZATION FUNCTION [PARTNER A]

Write a function that receives the array(s) of parking data. Starting at space 26, this function will check if there are any staff members in parking spaces 26-50. If so, this function will check to see if the staff member can be moved to a better (lower-numbered) space. If such a space exists, the staff member is to be moved to that location. Note that staff members should not be moved to a worse (higher-numbered) space.

This function should use (call) the functions from parts (d) and (f).

PART (H) – FUNCTION TO OUTPUT DATA [PARTNER A]

Write a function that receives an already opened output file and the array(s) of parking data. The function outputs the parking list to the file.

PART (I) – MAIN FUNCTION [PARTNER B]

Complete the main() function so that the program meets the full specification, calling the above functions as appropriate.

SUBMISSION

Submit your array design, program code, and a copy of the output file in your assignment package.