CENG322 Operating Systems
2022-2023 Spring

Assignment #2

Due date: 13.04.2023, 11:55 pm

You will write a C code for a cafeteria menu system in this assignment. You are expected to write a system that holds the menu details and the total number of sales to calculate monthly income of the cafeteria.

You should define a global structure called “cafeteria” that consists of 3 different arrays, including three other types of struct and the name of the month for the given menu.

Cafeteria:

- Name of the month (initially unallocated and NULL)
- Pointer for the normal menu array (initially unallocated and NULL)
- Pointer for the vegan menu array (initially unallocated and NULL)
- Pointer for the vegetarian menu array (initially unallocated and NULL)

Each array will hold a different type of struct, and their total number will equal the number of days food has been served in a month. The 3 types of struct inside the cafeteria struct should be “normal menu”, “vegan menu”, and “vegetarian menu”. For example, if the food is served 20 days within a month, the normal menu array should contain 20 normal menu structs inside it. The same logic applies to other menu types as well.

Each menu structs has similar contents. For “normal menu”, it should consist of the following attributes:

- **Date:** The day and the month need to be held as strings.
- **Soup:** The name of the soup.
- **Main dish:** The name of the main dish.
- **Side dish:** The name of the side dish.
- **Extra:** Name any extra contents in the menu such as “salad”, “fruit”, etc.
- **Sale count array:** An array with three elements that looks like [‘student_count’, ‘academic_count’, ‘administrative_count’]. The first element represents the number of students that order the normal menu on the given date. The second element represents the academician count, and the last represents the administrative staff count.

The other types of menus also include the same attributes according to the logic above.

Please define all arrays and strings mentioned previously with dynamic allocation while properly using your knowledge of pointers.

Initially, you need to initialize the global cafeteria struct, including the menu structs as well. Then, you must record the three menus for each day within a month by using the provided "cafeteria_march_menu.csv" file in a function. The month's name should be read from the name of the CSV file. All counts inside count arrays for each day should be initialized as 0. You needed to arrange the global struct such that, you can reach all the elements by using its pointer.

After the cafeteria struct is initialized, for each menu of each day, a random count should be generated for each type of people that uses the cafeteria. The count arrays for each type should be updated according to these random counts. The range of the random numbers should be from 0 to 50.

Then, the total sales calculation for that month should be performed. The prices according to the type of customers follow:

- Student: 6 TL
- Academic personal: 16 TL
- Administrative personal: 12 TL

The type of menu does not affect the price.

Please write functions for the following operations:

1. Fulfill the described global structure with the given CSV file.
2. Randomly generate customer counts for the global cafeteria structure.
3. Calculate and show the total income in three different ways shown below:
    a. The distribution of the total gained money on the type of menus. For example, "normal menu: 1526 TL, vegan menu: 218 TL, vegetarian menu: 120 TL".
    b. Overall sales for a month using the customer counts within each menu. For example, "students: 922 TL, Academic personal: 538 TL, Administrative personal: 404 TL".
    c. Total sales income of the whole month i.e., summation of the values above. For example, "Total sale income: 1864 TL".

Afterward, you simply **need to call these functions with the cafeteria struct as input**. Finally, **print out the sales results** at the end of the month using these functions.

The naming scheme for the necessary functions are provided below. Other than these functions, you can freely create and name other utility functions as you deem necessary.

- void initialize_menus(Cafeteria *cafeteria, char *csv_file_name) {…}
- void record_customer_counts(Cafeteria *cafeteria) {…}
- void calc_and_show_income(Cafeteria *cafeteria) {…}

Submission Rules:

Important Note: Violation of any rules given below will end up with a point reduction.

- Your **code must be working**! If not, your work will be considered a **zero grade.**
- **Late submissions** will not be allowed.
- Include your **names and student numbers as a comment block** at the top of the code.
- **Cheating or collaborative work** with other teams will not be tolerated and will be considered a **zero grade**.

- Please **utilize structs and functions** as much as possible. Try to keep the contents of the main function as minimal as possible. The codes that do not properly use structs and functions will be penalized.
- It is enough for just one member of each group to upload the assignment **through Teams until the due date.**
- The uploaded file should be a **zip file** and its name should contain the student IDs of both group members i.e., "<studentID1>_<studentID2>.zip".

Sample Execution:

Below, you can find the console output of a sample execution. You don't have to print out the exact same results as the scenario below. You simply need to show the contents of your structures and the results of your functions. The example dates like "10-Mar" can be any date of your choosing.

~ ~ ~ ~ Normal menu for the first and last days of the month:
date--> "1-Mar"
soup--> "Tarhana Corba"
main_dish--> "Patlican Kebabi"
side_dish--> "Pirinc Pilavi"
extra--> "Ayran"

~ ~ ~ ~ Normal menu for the first and last days of the month:
date--> "31-Mar"
soup--> "Mercimek Corba"
main_dish--> "Arnavut Cigeri"
side_dish--> "Pirinc Pilavi"
extra--> "Ayran"

~ ~ ~ ~ Vegan menu for the first and last days of the month:
date--> "1-Mar"
soup--> "Tarhana Corba"
main_dish--> "Patlican Yemegi"
side_dish--> "Pirinc Pilavi"
extra--> "Salata"

~ ~ ~ ~ Vegan menu for the first and last days of the month:
date--> "31-Mar"
soup--> "Mercimek Corba"
main_dish--> "Kereviz"
side_dish--> "Pirinc Pilavi"
extra--> "Meyve"

~ ~ ~ ~ Vegetarian menu for the first and last days of the month:
date--> "1-Mar"
soup--> "Tarhana Corba"
main_dish--> "Patlican Yemegi"
side_dish--> "Pirinc Pilavi"
extra--> "Yogurt"

~ ~ ~ ~ Vegetarian menu for the first and last days of the month:
date--> "31-Mar"
soup--> "Mercimek Corba"
main_dish--> "Kereviz"
side_dish--> "Pirinc Pilavi"
extra--> "Ayran"

Example output for normal menu ====> "3-Mar" | "Terbiyeli Tel Sehriye Corba" | "Etli Kuru Fasulye" | "Pirinc Pilavi" | "Salata"

Example output for vegan menu ====> "10-Mar" | "Ezogelin Corba" | "Enginar" | "Piyaz" | "Sogus"

Example output for vegetarian menu ====> "20-Mar" | "Kremali Domates Corba" | "Kereviz" | "Pirinc Pilavi" | "Ayran"

Counts for the example normal menu output of March 3rd ====> Student:50, Academic:16, Administrative:23

Counts for the example vegan menu output of March 10th ====> Student:19, Academic:33, Administrative:11

Counts for the example vegetarian menu output of March 20th ====> Student:14, Academic:28, Administrative:40

********** The Sales Results **********

Normal menu sales: 20618 TL, Vegan menu sales: 19198 TL, Vegetarian menu sales: 20616 TL
Student sales: 11064 TL, Academic personal sales: 27552 TL, Administrative personal sales: 21816 TL
Total sale income: 60432 TL