Title: Grocery Store Database

Due Date: Wednesday, 16 December , 2020 (no later than 15:00 MYS time)

START EARLY - LATE PENALITIES SHALL APPLY!!!

Marks: 25%

Objectives: To write a menu-driven program to process structures, arrays, and text files.

Problem Description:
In this assignment you are to write a program that will maintain a small grocery database (i.e., list of structures). The data for each grocery item consists of the item universal product identifier (integer), item name (string), the unit cost (double), and a character indicating the product category (i.e., 'M' for meat, 'P' for produce, 'D' for dairy, 'C' for canned goods, and 'N' for nonfoods).

You **must use** the following structure for each item:

```
struct item {
     int id;
     char name [20];
     double cost;
     char category;
     struct item *next;
};
```

Your program should first ask the user the name of the text file that contains the data for the database. Then, it should open the file and read its contents into a list of item structures. After the file is opened and the contents are read into the linked list (, close the file and display the following menu:

1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit

Each menu choice should invoke an appropriate function to perform the selected menu item/task. Your program must display the menu after each task and wait for the user to select another option until the user chooses to quit the program.

Menu Option 1: Your program should prompt the user for the item's id number, name, cost and category. The new item must be appended to the end of the database (in memory). Do not allow duplicate items (names nor id) in the database. An item is a duplicate if it's id number matches that of another product already in the database, or if its name is identical to the name of another product in the database. If the user attempts to add a duplicate item, do not add it and print out an error message stating that the item was not added since it was already in stock.

Menu Option 2: Your program should prompt the user for the item's id number. If the id number is found in the database, your program should delete that element from the list (of structures) by replacing it with the last record in the database. If the id number is not found, your program should notify the user.

Menu Option 3: Your program should prompt the user to enter an existing id number. Once the user enters the id number, your program should get the new value for the item's cost, and update the database accordingly.

Menu Option 4: Your program should prompt the user to enter an item name. If the item name is found in the database, your program should display the contents of that item. Otherwise, it will notify the user that the item is not found in the database.

Menu Option 5: Your program should display the contents of the entire database preferably in tabular/spreadsheet format. The user has the option to have the entire database **sorted** (either ascending or descending) based on either the item universal product identifier, item name, unit cost or product category.

Your program should initialize a "**save flag**" to false (i.e., 0) right after the database has been loaded from the input file. If any item is added, deleted, or edited, your program should change the flag to true (i.e., 1). When the "Quit" menu option is selected, your program should check the save flag and determine if changes have been made to the database. If changes have been made (i.e., save flag is true), your program should prompt the user to determine if these new changes should be saved. If the user wishes to save the changes, your program should prompt the user for the name of the text file to which the contents of the database will be written. Copy the contents of the database (linked list of structures) into this new file so that the new file as the exact same format specified above for the input file and terminate the program after closing the new file.

- You must use a structured programming approach for this assignment. That is to say, you must have at least **six (6) functions,** including the "main" function, with appropriate return types, whenever is suitable. All user-defined functions must have function prototypes and function definitions.
- Your program must be menu-driven. In the beginning, after reading the input file name and the contents of the file into the database (linked list of structures), the menu with 6 options should appear. The user should be allowed to continue making menu choices until they choose to quit. If the user makes an illegal menu choice (e.g., 8), you must display a message indicating that the choice was illegal and display the menu again.
- You are free to format your output in an appropriate and organized manner, but a sample will be provided for you below.

Input/Output File Specifications:
For testing your program, you should create a data file of input. The data file should be a text file (i.e., you can create it using notepad). The text file should include one record on each line, with spaces between each field. On each line, the universal product identifier is followed by the item name, the price of the item and the product category.

Here are the contents of a sample file, food.txt:

```
400  Corn    0.89    C
100  Salmon  7.99  M
200  Bananas   1.58   P
500  Dishes  3.45  N
300  Milk  1.89   D
```

Of course, your program should continue reading records (line by line) until it detects the EOF. Once the data are read into the database (linked list of structures) in memory, all changes should be done in the linked list. When the user chooses to quit the program, if the "**save flag**" is true, your program should give the user the option of saving the changes. If the user wishes to save the changes, your program should prompt the user to give a new file name. Your program must not accept the same file name as the original data file. The changes must be written to a new/different file.

References (for Concepts):
Structures, Pointers, Data Structures

Restrictions
Name the file you create and turn in *grocery.c*. Although you may use other compilers, your program must compile and run using gcc. Your program should include a header comment with the following information: your name, assignment title, and date. *You should also include comments throughout your code, when appropriate*.

Deliverables:
- □ A single source file named *grocery.c* submitted via Moodle

Additional Notes:
- □ As a precautionary measure and for documentation, you should email a copy of your source code to yourself.
- □ **NO global pointer is allowed.**
- □ You are welcome to use any compiler you wish. ***However*, before submitting the program for a grade, please verify that your program executes correctly using the gcc compiler.**
- □ Cheating will not be tolerated on this assignment (nor in this class). Those persons who are caught cheating will be handled as outlined in the University's Regulations.

Sample Output (Note: your output does not have to follow this exactly.)

```
Welcome to the grocery store!
Please input the file you'd like to load into stock.
food.txt
food.txt has been properly loaded into the database!

Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
1

What is the ID number of the item to add?
800
What is the name of the item to be added?
Chicken
What is the cost of the item to be added?
5.99
What is the product category of the item to be added?
M
Chicken has been added into the database.

Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
2

What is the ID number of the product to delete?
200
Bananas have been deleted from the database.

Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
3
What is the ID number of the item in question?
400
What is the new price for Corn?
0.99
Corn now costs 0.99.
```

```
Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
4

Which item would you like to search for?
Milk
Here is the product information:
300  Milk 1.89 D

Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
4

Which item would you like to search for?
Cereal
Sorry, we do not have Cereal in the store.

Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
2

What is the ID number of the product to delete?
199
Sorry, there is no item in the database with that ID number.

Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
1

What is the ID number of the item to add?
300
What is the name of the item to be added?
Yogurt
```

```
What is the cost of the item to be added?
0.50
What is the product category of the item to be added?
D
Sorry, an item with the ID number 300 is already in the database.

Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
5

Here is a listing of all the products in stock:<- without sorting
ID    Product     Price Category
400   Corn        0.89        C
100   Salmon      7.99        M
800   Chicken     5.99        M
500   Dishes      3.45        N
300   Milk        1.89        D

Please make one of the following selections:
1. Add new item
2. Delete item
3. Change the cost of an item
4. Search for item
5. Display inventory details
6. Quit
6

Would you like to save the changes made to the database?
yes
What file would you like to save the new database?
newfood.txt
The file has been saved. Thanks for shopping!
```

(Note: It's not necessary for your chart for menu item 5 to print out perfectly like the one in the sample above.)