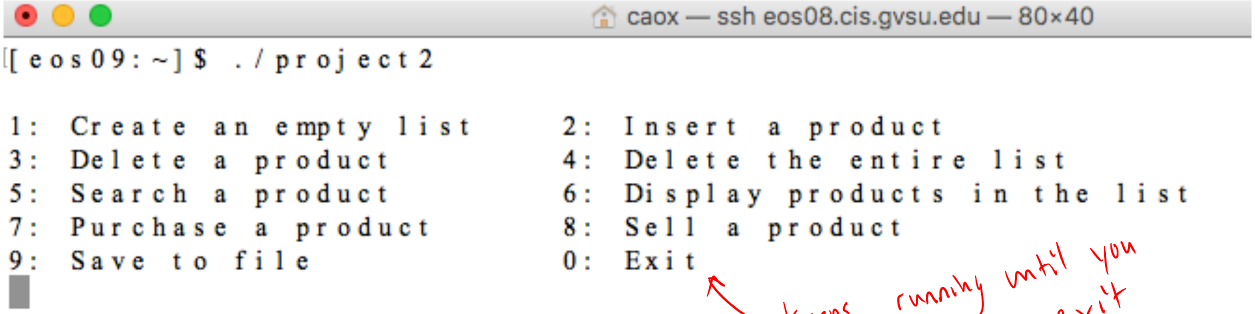
CIS241 Winter 2018 - Project 2: Product Management System
(Due 10:00PM on Thursday, March 22, 2018)
(You may work on this project individually, or as a group of two people)
Maximal Points: 100 (9% of the Final Grade)

In this project, you will simulate a simple product management system using Linked Lists in C. You need to create and maintain a Linked List that manages the current products' information. Each product's information is stored as a node in this Linked List, including product's name, unit, price and quantity.

Product's name: The name of a product (e.g., water, milk, beef). It is a ***string***.
Product's unit: The unit of a product (e.g., bottle, gallon, pound). It is a ***string***.
Product's price: The price of a product (e.g., 3). It is of type ***int***.
Product's quantity: The quantity of a product (e.g., 100). It is of type ***int***.

First, your program should have a menu for choices as follows.



```
[eos09:~]$ ./project2

1: Create an empty list       2: Insert a product
3: Delete a product           4: Delete the entire list
5: Search a product           6: Display products in the list
7: Purchase a product         8: Sell a product
9: Save to file               0: Exit
```

keeps running until you exit

Also, it should support the following operations.

"new list"

- Create an empty list: When '1' is pressed, an empty list will be created. (For the implementation, you can create a Linked List with only one node as the head.) This is the starting point of the program, before operations 2 ~ 9 can be executed.
- Insert a product: When '2' is pressed, the system should take user's keyboard input for a new product's information (e.g., name, unit, price, quantity), create a new node, and add it into the Linked List.

search

- Delete a product: When '3' is pressed, the program should take user's keyboard input for the product's name, and delete this product (the node) from the Linked List.
- Delete the entire list: When '4' is pressed, the program should delete each product's information by removing all the nodes in the Linked List, including the head node that was created when you pressed '1'.
- Search a product: When '5' is pressed, the program should take user's keyboard input for the product's name, search this product in the Linked List, and return the search result (Found, or Not Found).

similar to Lab3

- Display products in the list: When '6' is pressed, every product in the Linked List will be displayed in the screen, including its name, unit, price and quantity.
- Purchase a product: When '7' is pressed, the program should take user's keyboard input for the product's name, and increase this product's quantity by 1.

- Sell a product: When ‘8’ is pressed, the program should take user’s keyboard input for the product’s name, and decrease this product’s quantity by 1. **If this product’s quantity becomes 0, this product should be removed from the Linked List.**
- Save to File: When ‘9’ is pressed, the current products’ information should be written to a file. Each product’s information takes one line in the file.
- Exit: When ‘0’ is pressed, the program exits.

Assumptions: To simply the problem, we can have the following assumptions.

- Product’s name and unit are of types “string”. We can assume there are no blank spaces in these strings, e.g., a product’s name cannot be “wa ter”.
- Product’s price and quantity are of types “int”. We can assume the inputs from the user for these two variables are integers, rather than other data types.
- If “Save to File” is executed multiple times, the latest information of products will be saved to a file (i.e, File overwrite happens).
- Product’s name is the only identifier of a product and it is case-sensitive. In other words, we simply assume “Water” and “water” are two different products.

Robustness and Error handling: The program must have the following considerations.

- Operations 2 ~ 9 cannot be executed if an empty list has not been created (i.e., Operation 1 should be started first). – create head node
- For “Insert a product” (when ‘2’ is pressed), when a product’s name is provided, the program should search it among the existing products in the Linked List. If the product already exists in the Linked List, then it cannot be inserted into the system. (A message should be printed instead.)
- For operations 3, 7 and 8, the program also needs to first search the product by its name. If the product does not exist in the Linked List, these operations cannot proceed. (Some proper messages should be printed instead.)
- The program should keep running until ‘0’ is pressed, if the user provides the correct input.
- For “Insert a product”, the program should not allow negative numbers or zeros for a product’s price and quantity. (A message should be printed if the user does provide negative numbers and zeros.)
- For “Sell a product”, if a product’s quantity becomes 0 after it is decreased by 1, this product should be removed from the Linked List.

In this project, you must provide a makefile to compile the codes. I will use “make” to compile your program. You must make sure your program can run correctly on EOS machines.

Recommended program structure:

You can create two C source files. One is for main() function, and the other is for implementing all the functions to support operations (0 ~ 9). You can also have a header file including all the function prototypes.

Grading Criteria:

Menu: 5% (The program has a menu to take user's input correctly.)

Basic Operations: 60%

- Operation 1: 5%
- Operation 2: 10%
- Operation 3: 10%
- Operation 4: 5%
- Operation 5: 5%
- Operation 6: 5%
- Operation 7: 5%
- Operation 8: 5%
- Operation 9: 5%
- Operation 0: 5%

Robustness and Error handling: 30% (Each bullet point mentioned in the above "Robustness and Error handling" section accounts for 5%.)

Makefile: 5%. (You must provide a makefile for compilation.)

Submission guideline:

Please submit all your source files, header files and the makefile directly to the BlackBoard **(without compressing them to a ".zip" file)**. Please also name one of your C source files using your last name. If you work with another student, name the file using both of your last names with the '_' as the delimiter, e.g., "smith_johnson.c".

If you work as a group of two, **only one member** needs to submit the files. Please include your group member's name in the "comments" field of the submission page on the BlackBoard. **You must make sure your group member includes your name if your files are submitted by him/her.**