Sabanci University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Summer 2017-2018

Homework 2 – Alan's Psychedelic Breakfast
Due: 13 July 2018 11.55pm (Sharp Deadline)

DISCLAIMER:

Your program should be a robust one such that you have to consider all relevant user mistakes and extreme cases; you are expected to take actions accordingly!

Only checking the sample run cases might not be sufficient as your solution will be checked against a variety of samples different than the provided samples; however checking these cases are highly encouraged and recommended.

You can NOT collaborate with your friends and discuss your solutions with each other. You have to write down the code on your own. Plagiarism will not be tolerated AND cooperation is not an excuse!

Introduction

The aim of this homework is to practice on linked lists structures. In this homework, you will implement a dynamic shopping list for Alan's psychedelic breakfast by using a **circular doubly linked list**. Several operations will be available to the user through a recurring menu.

Input to Your Program

All inputs to your program will be done through a option menu which looks like below:

```
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit
```

If the user selects option 1, the name and the quantity of the item will be asked to the user. Also, the estimated price of the item may be asked as well (see the next sections for clarification). Or, if the user selects option 2, the name of the item will also be asked to the user.

Format of the Inputs

In the main menu, a valid input can either be "1", "2", "3", "4", "5" or "6". Trying to read this input as a string may save you a lot of headaches. Be reminded to check if the user enters anything other than valid option numbers, examples of which exist in the sample runs.

For the options "1" and "2", the name of the item will be asked. This item name might consist of multiple words and there can be multiple spaces or tabs between the words. Your program should handle these cases and the final item name should have single spaces between the words (and no space in the end :)). A good way to handle this might be reading the console input by *getline* and split this line into words by using a string stream. You can use getline for reading standard input like this:

```
string line;
cin.clear();
getline(cin, line);
```

And the string *line* will hold the whole line of the input from the console.

Moreover, Alan is a little obsessed and likes everything written in uppercase. So, all letters of the item names should be capitalized, you can use the *strutils* library for this purpose. The *cpp* and *header* of the abovementioned library are given to you with the homework bundle. Besides, as mentioned above, for each item, there will be quantity and estimated-price-per-quantity data. These two fields should be considered as integers. You are free to assume that the user enters a positive integer for those fields all the time.

Extra information for curious folks, skip if not interested: Having quantity and price of the items as double makes a lot more sense in real life scenarios. However, checking the equality of fractional numbers are a little tricky for computers and it may affect the grading process. This is why we accept them as integers for this homework.

For more information regarding this situation, you can check this page:
http://wiki.c2.com/?RealNumbersAreNotEqual

Data Structure to be Used

You **must** use a **circular doubly linked list** for the implementation of the shopping list. Apart from the automated grading, we will also inspect your code in detail. Faking to use a linked list will not help you and you will end up with a grade of 0 directly. So, please do not try to get

away with a vector or array implementation.

You can encapsulate the name, quantity and estimated price information regarding one item in a struct, as we did for all the linked list examples in this course. Then, you can basically form a circular doubly linked list of these struct elements. Remember that you should also have some kind of pointer**s** in these structs for connecting them to each other. You are also allowed to keep extra fields in the struct if you feel like so, but these three data fields (and some pointer**s**) should be enough minimally.

Details of the Tasks and Outputs of Your Program

In the first task, addition of an item to the shopping list may occur in two different cases: (i) Either this item is a new item and it will be added to the shopping list, or (ii) it already exists in the list and the quantity of the item will be updated (increased by the new quantity). In the first case, you should also ask about the estimated price of the item whereas in the second case, you should not be asking about it, as the value which is already known will be kept using.

The second task is deletion of an item from the shopping list. Again, two case might occur for this: The requested item may exist in the list or not. If it exists, the node regarding that item will be removed from the list, and if not, an error message will be printed.

For the first two tasks, there are two points to be clarified. First of all, after getting the names of the items, you should consider their uppercase versions, otherwise you might have duplicate entries in your list or your deletion operations might fail unexpectedly. Always try to convert names to uppercase as soon as you obtain them. Second thing is that Alan is a little keen on his shopping list and he wants things organized according to some measures. For this, you should always have the list sorted with respect to the total estimated prices (i.e., quantity * e.price) of the items. If two items have equal estimated prices, they should be sorted with respect to their names alphabetically (greater total estimated price comes first; if equal, smaller name comes first).

The third and fourth tasks are basic linked lists traversals. In order for your program to find the correct results in these steps, your list should be sorted (descending wrt estimated price and ascending wrt name if necessary) after each action taken in first and second tasks. For the format which you will show items in the console, you can see the sample runs below.

The fifth task is another linked linked list traversal problem, so it does not need much explanation, we assume.

The main menu will be printed after every action, until the user selects the sixth option, “Exit”. Once the user selects to exit, you should print an appropriate acknowledgement message, **clear the circular doubly linked list from the memory**, and exit the program. Not returning dynamic memory back is known to cause “memory leak”, which is a serious issue. If you do not clear your allocations from the memory, it may cause a leak in your grade as well.

Last warning: always be reminded to check if pointers are not *NULL* before you use them :)

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are the standard inputs (cin) taken from the user (i.e., like ***this***). You have to display the required information in the same order and with the same words as here.

Sample Run 1

This program helps Alan with gathering the shopping list for his psychedelic breakfast.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***0***

This is not a valid option!

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***7***

This is not a valid option!

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***Alan***

This is not a valid option!

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***3***

The shopping list is empty.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***4***

The shopping list is empty.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***5***

The total estimated price of 0 kinds of items is: 0

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***6***

Clearing the shopping list...
Exiting the program...

Sample Run 2
This program helps Alan with gathering the shopping list for his psychedelic breakfast.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***1***

Enter name for the item: ***bread***
Enter quantity for the item: ***2***
Enter estimated price for the item: ***1***

The item BREAD of quantity 2 is added to the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***1***

Enter name for the item: ***CoffEE***
Enter quantity for the item: ***3***
Enter estimated price for the item: ***2***

The item COFFEE of quantity 3 is added to the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***3***

Item: COFFEE
Quantity: 3
Est. Price: 2

Item: BREAD
Quantity: 2
Est. Price: 1

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***1***

Enter name for the item: ***sausages***
Enter quantity for the item: ***2***
Enter estimated price for the item: ***21***

The item SAUSAGES of quantity 2 is added to the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***5***

The total estimated price of 3 kinds of items is: 50

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***1***

Enter name for the item: ***BreAD***
Enter quantity for the item: ***4***

The quantity of the item BREAD is increased by 4. The final quantity is 6.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***4***

Item: COFFEE
Quantity: 3
Est. Price: 2

Item: BREAD
Quantity: 6
Est. Price: 1

Item: SAUSAGES
Quantity: 2
Est. Price: 21

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***1***

Enter name for the item: ***marmalade***
Enter quantity for the item: ***2***
Enter estimated price for the item: ***5***

The item MARMALADE of quantity 2 is added to the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***3***

Item: SAUSAGES
Quantity: 2
Est. Price: 21

Item: MARMALADE
Quantity: 2
Est. Price: 5

Item: BREAD
Quantity: 6
Est. Price: 1

Item: COFFEE
Quantity: 3
Est. Price: 2

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***2***

Enter an item name to delete from the list: ***BRead***

The item BREAD is deleted from the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***3***

Item: SAUSAGES
Quantity: 2
Est. Price: 21

Item: MARMALADE
Quantity: 2
Est. Price: 5

Item: COFFEE
Quantity: 3
Est. Price: 2

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***5***

The total estimated price of 3 kinds of items is: 58

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***2***

Enter an item name to delete from the list: ***coffee***

The item COFFEE is deleted from the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***2***

Enter an item name to delete from the list: ***coffee***

The item COFFEE could not be found in the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***4***

Item: MARMALADE
Quantity: 2
Est. Price: 5

Item: SAUSAGES
Quantity: 2
Est. Price: 21

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***2***

Enter an item name to delete from the list: ***sauSAGES***

The item SAUSAGES is deleted from the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list

6. Exit

Enter your choice: ***5***

The total estimated price of 1 kinds of items is: 10

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***3***

Item: MARMALADE
Quantity: 2
Est. Price: 5

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***2***

Enter an item name to delete from the list: ***marmalaDE***

The item MARMALADE is deleted from the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***3***

The shopping list is empty.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***2***

Enter an item name to delete from the list: ***peace***

The item PEACE could not be found in the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***1***

Enter name for the item: ***bACon***
Enter quantity for the item: ***100***
Enter estimated price for the item: ***4***

The item BACON of quantity 100 is added to the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list

6. Exit

Enter your choice: ***1***

Enter name for the item: ***peace***
Enter quantity for the item: ***1***
Enter estimated price for the item: ***100000***

The item PEACE of quantity 1 is added to the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***4***

Item: BACON
Quantity: 100
Est. Price: 4

Item: PEACE
Quantity: 1
Est. Price: 100000

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***5***

The total estimated price of 2 kinds of items is: 100400

MENU

1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***1***

Enter name for the item: ***GooD daY***
Enter quantity for the item: ***1***
Enter estimated price for the item: ***5000***

The item GOOD DAY of quantity 1 is added to the list.

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list
4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***3***

Item: PEACE
Quantity: 1
Est. Price: 100000

Item: GOOD DAY
Quantity: 1
Est. Price: 5000

Item: BACON
Quantity: 100
Est. Price: 4

MENU
1. Add item to the shopping list
2. Delete item from the shopping list
3. Print the shopping list

4. Print the shopping list in reverse order
5. Show total estimated cost of the list
6. Exit

Enter your choice: ***6***

Clearing the shopping list...
Exiting the program...

Some Important Rules

Although some of the information is given below, please also read the homework submission and grading policies from the lecture notes of the first week. In order to get a full credit, your program must be efficient, modular (with the use of functions), well commented and indented. Besides, you also have to use understandable identifier names. Presence of any redundant computation, bad indentation, meaningless identifiers or missing/irrelevant comments may decrease your grade in case that we detect them.

When we grade your homeworks, we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we are going to run your programs in Release mode and **we may test your programs with very large test cases**. Hence, take into consideration the efficiency of your algorithms other than correctness.

How to get help?

You may ask your questions to TAs or to the instructor. Information regarding the office hours of the TAs and the instructor are available at SUCourse.

YOU CAN USE GRADE CHECKER FOR THIS HOMEWORK!

You can use GradeChecker (http://sky.sabanciuniv.edu:8080/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

Make sure you upload the *strutils.h* and *strutils.cpp* files, too.

Grade Checker and the automated grading system use a different compiler than MS Visual Studio does. Hence, you should check the "Common Errors" page to see some extra situations to consider while doing your homework. If you do not consider these situations, you may get a lower score (even zero) even your program works correctly with MS Visual Studio.

Common Errors Page: http://sky.sabanciuniv.edu:8080/GradeChecker/commonerrors.jsp

Grade Checker can be pretty busy and unresponsive during the last day of the submission. Due to this fact, leaving the homework for the last day generally is not a good idea. You may wait for hours to test your homework or make an untested submission, sorrily..

Grade Checker and Sample Runs together give a good estimate of how correct your implementation is, however we may test your programs with different test cases and **your final grade may conflict with what you have seen on Grade Checker.** We will also **manually** check your code, indentations and so on, hence do not object to your grade based on the Grade Checker results, but rather, consider every detail on this documentation. **So please make sure that you have read this documentation carefully and covered all possible cases, even some other cases you may not have seen on Grade Checker or Sample Runs**. The cases that you *do not need* to consider are also given throughout this documentation.

Submit via SUCourse ONLY! **Grade Checker is not considered as a submission**. Paper, e-mail or any other methods are not acceptable, neither.

The internal clock of SUCourse might be a couple of minutes skewed, so make sure you do not leave the submission to the last minute. In the case of failing to submit your homework on time:

"No successful submission on SUCourse on time = A grade of 0 directly."

What and where to submit (PLEASE READ, IMPORTANT)

You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the abovementioned platform while testing your homework.

It'd be a good idea to write your name and lastname in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts). If your full name is "Duygu Karaoğlan Altop", and if you want to write it as comment; then you must type it as follows:

```
// Duygu Karaoglan Altop
```

Submission guidelines are below. Since the grading process will be automatic, you are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be zero. The lack of even one space character in the output will result in your grade being zero, so please test your programs yourself and with the Grade Checker tool explained above.

- Name your cpp file that contains your program as follows:

"SUCourseUserName_hw2.cpp"

Your SUCourse user name is actually your SUNet username which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SU e-mail address is **atam@sabanciuniv.edu**, then the file name must be: **"atam_hw2.cpp"**

- Please make sure that this file is the latest version of your homework program.
- You should upload *strutils.h* and *strutils.cpp* to SUCourse as well.
- Do not zip any of the documents but upload them as separate files only.
- Submit your work **through SUCourse only**! You can use the Grade Checker only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse.

You may visit the office hours if you have any questions regarding submissions.

Plagiarism

Plagiarism is checked by automated tools and we are very capable of detecting such cases. Be careful with that...

Exchange of abstract ideas are totally okay but once you start sharing the code with each other, it is very probable to get caught by plagiarism. So, do NOT send any part of your code to your friends by any means or you might be charged as well, although you have done your homework by yourself. Homeworks are to be done personally and you have to submit your own work. **Cooperation will NOT be counted as an excuse.**

In case of plagiarism, the rules on the Syllabus apply.

Good Luck!
Tolga Atam, Duygu K. Altop