

CSCI-1200 Data Structures — Fall 2009

Homework 4 — Grocery Lists

In this assignment you will write a program to manage the contents of a kitchen as various recipes are prepared. Your program will handle several different operations: adding a recipe, buying ingredients for the kitchen, printing out a recipe, making a recipe (which removes ingredients from the kitchen), and printing out the current contents of the kitchen. *Please carefully read the entire assignment before beginning your implementation.*

The input for the program will come from a file and the output will also go to a file. These file names are specified by command-line arguments. Here's an example of how your program will be called:

```
grocery_list.exe requests.txt results.txt
```

The form of the input is relatively simple. Each request begins with a single character. There are five types of requests, described below. You may assume the input file strictly follows this format (i.e., you don't need to worry about format error-checking).

```
r salad
2 tomatoes
1 lettuce
0
a
3 lettuce
4 tomatoes
0
p salad
m salad
k
```

1. The first request, indicated by the letter 'r', says to add a particular recipe to the list of recipes. The ingredients follow, listed one per line with the quantity and name. The ingredient list ends with a '0' on a line by itself. You should verify that there is not already a recipe with this name in the list of recipes. The output for this request is:

```
Recipe for salad added
```

2. The second request, indicated by the letter 'a', says that various ingredients should be added to the kitchen supplies. Again, the ingredients follow, listed one per line with the quantity & name, and the ingredient list ends with a '0'. When you add an ingredient to the kitchen you need to first check if the ingredient is already present. If it is, simply increase the quantity of that item. The output for this request indicates the number of distinct ingredients (not the quantity) that were added:

```
2 ingredients added to kitchen
```

3. The third request, indicated by the letter 'p', asks that the recipe for **salad** be output. The ingredients should be listed in alphabetical order with their quantities. For example:

```
To make salad, mix together:
  1 unit of lettuce
  2 units of tomatoes
```

Note the user friendly output text distinguishes between singular and plural quantities.

If the program does not have the requested recipe, the following message should be output:

```
No recipe for salad
```

4. The fourth request, indicated by the letter ‘m’, asks the program to attempt to make a particular recipe. First the program will check the kitchen to see if the necessary ingredients are available. If all of the ingredients are available in sufficient quantities, the kitchen will be edited to remove the appropriate amounts of each ingredient and this message is output:

```
Made salad
```

If the ingredients are not available, the output message will list the insufficient ingredients (sorted alphabetically), with the quantities that are missing. In this example, one salad can be made, but if a second salad is requested, the following message will be output:

```
Cannot make salad, need to buy:  
  1 unit of tomatoes
```

If the program does not have the requested recipe, the following message should be output:

```
Don't know how to make salad
```

5. The final request, indicated by the letter ‘k’, asks for the current contents of the kitchen to be output, sorted by quantity first, and then alphabetically for items with equal quantity. In this example:

```
In the kitchen:  
  1 unit of tomatoes  
  3 units of lettuce
```

If a particular ingredient has been “used up”, it should be removed from the list of ingredients.

Sample input and output files are posted on the course web site. Please follow these examples exactly.

Additional Requirements, Hints and Suggestions

- **You may not use vectors or arrays for this assignment.** Use standard library lists instead. You may not use maps, or sets, or things we haven’t seen in class yet.
- You must write at least one new class. We have provided a partial implementation of the main program to get you started. There are member function calls to our versions of the `Recipe` and `Kitchen` classes, so you can deduce how some of the member functions in our solution work. You may use none, a little, or all of this, as you choose, but we strongly urge you to examine it carefully.
- For extra credit you may implement a sixth request ‘w’ which looks through the recipe list and suggests items that can be made from the current kitchen contents. Include your new test cases and a short writeup in your `README.txt` file.
- Do all of your work in a new folder named `hw4` inside of your homeworks directory. Use good coding style when you design and implement your program. Be sure to make up new test cases to fully test your program and don’t forget to comment your code! Use the template `README.txt` to list your collaborators and any notes you want the grader to read. When you are finished please zip up your `hw4` folder exactly as instructed for the previous assignments and submit it through the course webpage.