Food Bank Program

By Preston Haynes

The parts of the program:

The FoodItem class stores the name, food group, price, weight and volume of a food item. It also has the quantity available to buy and the amount that will be bought. Variables of ratios of how many food items can be purchase for list sorting purposed are included, too.

The FoodGroup class stores the name and the count of food items bought in that food group.

The FoodPantry class stores the list of FoodItems and the list of FoodGroups along with the budget, weight and volume limits. Food items can be “bought” within the FoodPantry, where the count of the food item is added, and its price is taken away from the budget. When the item is bought the weight and volume is subtracted from the weight limit and volume limit.

The TextFileReader class is used to read the input.txt file, parse it to the appropriate variables they belong and return a FoodPantry object.

The method of solving the tasks:

Buying the most items:

The algorithm I used for task 1,2 and 4, use the same methods but with different parameters. The three tasks were to buy as many items with varying budget, weight and volume constraints. For these tasks I wrote an algorithm that sorts the list (array) of food items before procedurally “buying” each item. The list of food items are rearranged in an order how many items can be bought within the constraints given. Food Items that are cheaper, weigh less and take up less volume are at the top of the list. The items are then “bought” one by one until the budget is depleted, or the weight or volume limit is reached.

Creating balance between food groups:

Task 3 was unique because it was the only one with the problem of finding balance between the food groups. This has a different set of methods from the other three tasks. The task was to find a balance of 25% ± 5% of the 4 food groups. That means that the lowest counted food group is at least 20% of the total count. The other food groups can be more, but only up to 30% of the total count. So algorithm I wrote uses two different “stages” . The first stage buys one item per food group as long as it is possible given the budget, weight or volume constraints. The next stage buys individual food items until the budget, weight, or volume is depleted or until the food group of the item has a count that exceeds 30% of the total count. The program can also accommodate more or less than 4 food groups, creating a balance that is within the margin of ± 5%.