Seneca Groceries

You have been asked to write a program for a grocery store that will:

- Start by allowing the entry of the current stock in the store, which it will save and update during the day.
- Switch to sales mode where it will act like a cash register and allow customers to purchase things. It will update the inventory as items are sold and will produce an itemized receipt [CG1] for the customer.
- At the end of the day, print a summary of the sales, the inventory remaining and the top sellers in each category.

Products for sale are identified by name, and are placed in a category and the category is entered by its number:

1. produce
2. bakery
3. meat
4. dairy
5. baking
6. house wares
7. miscellaneous

Each item has a price and an indication of whether it is sold per unit or by weight which is entered as 0 if sold by the number purchased or 1 if sold by weight. It also indicates how much we have of this item either in units or by weight rounded down to the nearest kilogram.

When we sell items, they are sold by weight in fractional Kg or by integral unit counts. When the final amounts in inventory are calculated, they are rounded down to the nearest Kg for items sold by weight. If a customer tries to purchase more of something than we have in stock, we sell only the amount we have in stock. Most groceries are tax free except items in the house wares and miscellaneous categories which are taxed at 13%.

The list of items being purchased is terminated by an item with the ID of 0. A sale with no items added to it marks the end of sales which should be followed by the summary of the day's sales.

Input data should be checked for correct category range (1-7), sold by weight range (0-1). As the stock is entered, each item is assigned a numeric identifier from 1 upwards in the order the items are entered. For stock entry, you can assume that each item is only entered once.

When purchasing, the customer will identify the product to purchase by the identifier assigned to it when it was entered in the stock and listed in the initial stock report in the ID column. These product ID number should be checked to ensure that they are valid.

Below this section you will find:

- main.c which is the main for the program. You must use this main.c for your program without changes. This means that you will need to implement the data structures it uses and the functions it uses in the files stock.h and stock.c. You SHOULD create additional functions, other than the ones called from main, to create a highly modular, well-designed program. Your code will be tested against this main.c and your code must work and produce the expected output.
- A sample execution of the program to which your program's output should look very similar.
- Sample data which can be cut and pasted into the program for final testing. This is the data which generated the sample output.

You are required to:

- Place a comment at the top of both stock.h and stock.c listing the names of every member of the group.

- BONUS: if you read the existing stock from a file containing the test data with the errors fixed. You will need to correct the errors in the data and place it in a file. Then, you can read it from the file without the error checking that is done with the manually typed version. You can change the value of the constant STOCK_FROM_STDIN declared in main to allow you to read from a file. Note that the sales will continue to be read from stdin, it is only the initial stock which will be read from the file. This will change the output to show the initial prompt for stock followed immediately the summary of the opening stock. The typing of the stock will vanish as it is being read from a file.

main.c

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include "stock.h"

#define MAX_STOCK_ENTRIES 100
/*************************************************************/
/* Students who are implementing the file reading version for  */
/* bonus marks are allowed to change STOCK_FROM_STDIN to 0     */
/* to enable the program to read from a file.                  */
/*************************************************************/
#define STOCK_FROM_STDIN 1

int main(void)
{
      struct StockRecord storeStock[MAX_STOCK_ENTRIES] = { {{0, 0, 0.0, 0}, 0, 0.0} };
      struct SalesRecord saleItems[MAX_ITEMS_IN_SALE];
      struct SalesRecord topSellers[5];

      int  numStockItems = 0, numSales = 0, numSaleItems, totalSaleItems = 0, cat;
      double sale, totalSales = 0.0;

      // Read existing stock from stdin
      printf("Enter current stock in format amount, category, price, byWeight, name (0 amount to
end):\n");

      numStockItems = readStockItems(storeStock, MAX_STOCK_ENTRIES, STOCK_FROM_STDIN);

      printf("\n");
      centreText(70, '*', " Seneca Groceries - Opening Stock ");
      printf("\n");
      centreText(70, '=', "");
      printf("\n");
      printStockReport(storeStock, numStockItems);

      printf("\n");
      centreText(70, '*', " Now in Sales Mode ");
      printf("\n");
```

```
	centreText(70, '=', "");
	printf("\n");
	do
	{
		numSaleItems = readSale(storeStock, numStockItems, saleItems);
		if (numSaleItems > 0)
		{
			sale = printSalesReport(storeStock, saleItems, numSaleItems);
			totalSales += sale;
			totalSaleItems += numSaleItems;
			numSales++;
		}
	} while (numSaleItems > 0);

	printf("\n");
	centreText(70, '*', " End of Day Summary ");
	printf("\n");
	centreText(70, '=', "");
	printf("\n");
	printf("%35s%8.2lf\n", "Cost of items sold before taxes", totalSales);
	printf("%35s%8d\n", "Number of Sales", numSales);
	printf("%35s%8.2lf\n\n", "Average items per sale", totalSaleItems / (double)numSales);

	printf("\n");
	centreText(70, '*', " Closing Stock ");
	printf("\n");
	centreText(70, '=', "");
	printf("\n");
	printStockReport(storeStock, numStockItems);
	printf("\n");
	for (cat = 0; cat < 5; cat++)
	{
		getTopSellers(storeStock, numStockItems, topSellers, 3, cat);
		printTopSellers(storeStock, topSellers, 3, cat);
	}
	for (cat = 5; cat < NUM_CATS; cat++)
	{
		getTopSellers(storeStock, numStockItems, topSellers, 2, cat);
		printTopSellers(storeStock, topSellers, 2, cat);
	}
	return 0;
}
```

Sample Output

Sample output from the program is shown below. Note that the input values are separated by commas and that you should do the same in your program.

```
Enter current stock in format amount, category, price, byWeight, name (0 amount to end):
21,9,1.99,0,broccoli
Invalid Category - Enter a number between 1 and 7: 1,1.99,2,broccoli
Invalid soldByWeight - Enter a number between 0 and 1: 0,broccoli
21,1,1.5,1,potato
3, 1,1.5,0,carrots
10,3,12.99,1,steak
2,3,9.5,1,Italian sausage
2,4,4.99,0,2% milk
2,4,3.99,0,coffee cream
2,2, 3.99,0, white bread
5,2,4.99,0,chocolate cup cakes
10,5,4.99,0,all purpose flour
8,5,5.99,0,whole wheat flour
5,5,2.99,0,raisins
2,6,0.99,0,scrub pads
4,7,13.95,0,laundry detergent
0

****************** Seneca Groceries - Opening Stock ******************
========================================================================
  ID                         Product        Category   Price Quantity
   1                        broccoli         produce    1.99       21
   2                          potato         produce    1.50       21
   3                         carrots         produce    1.50        3
   4                           steak            meat   12.99       10
   5                 Italian sausage            meat    9.50        2
   6                          2% milk           dairy    4.99        2
   7                     coffee cream           dairy    3.99        2
   8                      white bread          bakery    3.99        2
   9              chocolate cup cakes          bakery    4.99        5
  10                all purpose flour          baking    4.99       10
  11                whole wheat flour          baking    5.99        8
  12                          raisins          baking    2.99        5
  13                        scrub pads     house wares    0.99        2
  14                laundry detergent   miscellaneous   13.95        4

************************ Now in Sales Mode *************************
========================================================================
Enter a product ID to purchase, and the quantity (0 to stop): 1,2
Enter a product ID to purchase, and the quantity (0 to stop): 20,3
Invalid Product - Enter a number between 0 and 14: 2, 200
Invalid quantity - Enter a number between 0.10 and 100.00: 2,2
Enter a product ID to purchase, and the quantity (0 to stop): 3,2
Enter a product ID to purchase, and the quantity (0 to stop): 4,1
Enter a product ID to purchase, and the quantity (0 to stop): 6,3
```

```
Enter a product ID to purchase, and the quantity (0 to stop): 8,1
Enter a product ID to purchase, and the quantity (0 to stop): 0

********** Seneca Groceries **********
============================================================
                     broccoli     1.99    3.98
                       potato     1.50    3.00
                      carrots     1.50    3.00
                        steak    12.99   12.99
                      2% milk     4.99    9.98
                   white bread     3.99    3.99
Purchase Total                             36.94
Tax                                         0.00
Total                                      36.94
Thank you for shopping at Seneca!

Enter a product ID to purchase, and the quantity (0 to stop): 5,2
Enter a product ID to purchase, and the quantity (0 to stop): 7,1
Enter a product ID to purchase, and the quantity (0 to stop): 10,2
Enter a product ID to purchase, and the quantity (0 to stop): 14,1
Enter a product ID to purchase, and the quantity (0 to stop): 0

********** Seneca Groceries **********
============================================================
              Italian sausage     9.50   19.00
                 coffee cream     3.99    3.99
            all purpose flour     4.99    9.98
            laundry detergent    13.95   13.95
Purchase Total                             46.92
Tax                                         1.81
Total                                      48.73
Thank you for shopping at Seneca!

Enter a product ID to purchase, and the quantity (0 to stop): 0

********** End of Day Summary **********
============================================================
    Cost of items sold before taxes   83.86
                  Number of Sales       2
            Average items per sale    5.00


********** Closing Stock **********
============================================================
  ID                        Product        Category   Price Quantity
   1                       broccoli         produce    1.99       19
```

2	potato	produce	1.50	19
3	carrots	produce	1.50	1
4	steak	meat	12.99	9
5	Italian sausage	meat	9.50	0
6	2% milk	dairy	4.99	0
7	coffee cream	dairy	3.99	1
8	white bread	bakery	3.99	1
9	chocolate cup cakes	bakery	4.99	5
10	all purpose flour	baking	4.99	8
11	whole wheat flour	baking	5.99	8
12	raisins	baking	2.99	5
13	scrub pads	house wares	0.99	2
14	laundry detergent	miscellaneous	13.95	3

---------- Top 3 sellers in produce ----------

Rank	Product	Sales
1	broccoli	2.00
2	potato	2.00
3	carrots	2.00

---------- Top 3 sellers in bakery ----------

Rank	Product	Sales
1	white bread	1.00
2	<none>	0.00
3	<none>	0.00

---------- Top 3 sellers in meat ----------

Rank	Product	Sales
1	Italian sausage	2.00
2	steak	1.00
3	<none>	0.00

---------- Top 3 sellers in dairy ----------

Rank	Product	Sales
1	2% milk	2.00
2	coffee cream	1.00
3	<none>	0.00

---------- Top 3 sellers in baking ----------

Rank	Product	Sales
1	all purpose flour	2.00
2	<none>	0.00
3	<none>	0.00

---------- Top 2 sellers in house wares ----------

Rank	Product	Sales
1	<none>	0.00
2	<none>	0.00

--------- Top 2 sellers in miscellaneous ---------

Rank	Product	Sales
1	laundry detergent	1.00

```
   2                        <none>     0.00
```

Sample Data

```
21,9,1.99,0,broccoli
1,1.99,2,broccoli
0,broccoli
21,1,1.5,1,potato
3, 1,1.5,0,carrots
10,3,12.99,1,steak
2,3,9.5,1,Italian sausage
2,4,4.99,0,2% milk
2,4,3.99,0,coffee cream
2,2, 3.99,0, white bread
5,2,4.99,0,chocolate cup cakes
10,5,4.99,0,all purpose flour
8,5,5.99,0,whole wheat flour
5,5,2.99,0,raisins
2,6,0.99,0,scrub pads
4,7,13.95,0,laundry detergent
0
1,2
20,3
2, 200
2,2
3,2
4,1
6,3
8,1
0
5,2
7,1
10,2
14,1
0
```