CS121 - Computer Science II
Lab Assignment #3
Fall 2011

The purpose of this exercise is to give you experience with using dynamic memory and linked lists. The exercise should further reinforce what you know about files.

You did such a good job with the first part of computerizing the inventory of *Joe's Auto Parts* store that he would like you to now work on phase two. This time, you are to input the parts inventory information, apply the day's transactions to that database, and then generate some output reports - the current inventory list (after the transactions have been applied), a list of backordered items, and a reorder report. Since Joe expects to expand his inventory quickly, he doesn't want any limit on the number of parts the program can handle (**HINT:** This will require the use of a linked list to hold the part information.) All of the reports this time should be output in order of the part number (**HINT:** Perhaps the easiest way to do this is to create the list of part data in sorted order.)

This program will use **two** input files. The first is the previous inventory file:

Part No.	Part Description	No. in Stock	Value Each	Reorder Level	Reorder Qty

The second file is a "transaction" file. Each line has the following form:

Part No.	Transaction type	Number involved in transaction

The possible transaction types are:

- **A** - add. Add the number specified to the number in stock.
- **S** - sale. Subtract the number specified from the number in stock.
- **P** - price change. Change the price of the item in inventory.

The output from the program should consist of three different parts -

- A current inventory report, which shows the *updated* inventory status (i.e., after all transactions have been applied). In this report, the quantity in stock should **not** show less than zero.
- A reorder report, which lists all the items which should be reordered (i.e., those items whose current stock falls below the reorder level, as in your first program), and the number to reorder. The number to should be a multiple of the reorder quantity. Enough should be reordered to bring the current stock to a level that is above the reorder level, after any current sales and backordered amounts have been satisfied.

- A backorder report, which lists all those items for which the current stock cannot satisfy current sales (i.e., those items which would have a *negative* quantity in stock if all the transactions had been applied.) in stock).

Sample output: (NOTE: This output does not necessarily show the correct program results!)

```
                    Joe's Auto Store - Inventory Status

Current Inventory Report:

Part No.  Description       No. in Stock   Value Each   Total Value
--------  -----------       ------------   ----------   -----------

2027      Wing Nut               10           1.27          12.70
2030      Rocker Panel            5         137.85         689.25
2035      Wheel Rim               0          14.64          29.28
  .           .                   .            .             .
  .           .                   .            .             .
  .           .                   .            .             .
2974      Headlight - LoB         .            .             .
                                                           -------
                                 Total Inventory Value    $ xxxx.xx

Reorder list:

Part No.  Description       Reorder Qty.  Reorder Cost
--------  -----------       ------------  ------------

2035      Wheel Rim               6           87.84
  .           .                   .            .
  .           .                   .            .
                                                --------
                       Total Reorder Cost    $ xxxx.xx

Backorder list:

Part No.  Description     Backorder Qty.
--------  -----------     --------------

2035      Wheel Rim             3
  .           .                 .
  .           .                 .
```