Programming 2 – Assignment 2

Your program should apply the transactions described in the sales.txt file to the inventory.txt, generating appropriate error messages if a transaction cannot be completed. After the transactions have been performed, your program should determine the answers to the following queries:

* Print a list of the inventory, sorted in order of increasing price. (lowest to highest, console output)
* Which day resulted in the greatest sales volume (give the day, the total number of items sold an the total amount of money spent (in pounds and pence)).
* How many NPN transistors does Chartlins have in stock after processing all successful sales.
* What is the total resistance of all the remaining resistors in stock?

The program should consist of five files, which define the basis of the marking scheme:

* StockItem.h and StockItem.c These files should define a struct that represents a stock item and a set of functions used to access the struct (forming an abstract data type) (10 marks).
* Inventory.h and Inventory.c These files should define a struct representing the inventory of the company as a linked list (or other suitable data structure) and a set of functions used to access the struct (8 marks).
* Sales.h and Sales.c These files should define a struct representing the sales data, as a linked list (or other suitable data structure), where each node in the link contains a reference to the StockItem structure, the date and the number of items sold. The list should only record the successful transactions (8 marks).
* StockProgram.c This file contains the main part of the program that loads and processes the inventory and sales data, and provides answers to the four queries (6 marks).

The remaining 8 marks are awarded for the answers to the four inventory questions listed above.

# Task

* Store inventory in several struct arrays
* Read through sales file
* Apply transactions to inventory
* Generate error message if transaction cannot be completed
* Sort inventory from lowest to highest price
* Output inventory to console
* Determine which date resulted in greatest sales volume
* How many NPN transistors are in stock after processing all sales
* Total resistance of all resistors in stock

## Apply Sales

* Read in line from sales file
* Use “/ ,” delimiters to get day, month, year, stockcode and quantity
* For each line in sales file, loop through inventory file to find a matching stock code
* If not found, success = false and generate error message
* If found, check stock of item is greater than quantity purchased
* If not, success = false and generate error message
* If stock is greater than quantity then set stock to stock – quantity and set success to true
* If success is true, add the sales item to the sales list, with the corresponding stock item

## Best Sales Day

* Sort salesList by date
* Loop through salesList
* If date = previous date then multiply quantity by item cost and increment counter
* If date != previous date then check if counter is greater than max
* If counter is greater than max then set max to counter and set max date to lastdate

## Total resistance