Programming Assignment #2

Electronics Store Inventory

CS 2308.001 and 002 Fall 2016
Instructor: Jill Seaman

Due: Tuesday, 9/27/2016: upload electronic copy by 12:00 midnight.

Problem:

Write a C++ program that will allow a user to manage the inventory of a small store that sells electronics like mobile phones, video games, and streaming devices.

The inventory for the store will contain the following information for each product:

product name	(i.e. "Apple iPhone 3GS 8GB", may contain spaces in it)
sku	(stock keeping unit code, an integer)
quantity	(how many of this product in stock)
price	(in dollars and cents)

Note: Your program should be able to store up to 100 different products. You may assume that the skus will be unique (you do not need to check for this).

The program should offer the user a menu with the following options:

1. Add a new product to the inventory (prompt user for input values).
2. Remove a product from the inventory (by sku).
3. Display the inventory sorted by sku.
4. Lookup a product by sku.
5. Quit

The program should perform the selected operation and then re-display the menu.

Do not change the menu numbers associated with the operations.

For the **Add** operation, give one prompt for the user to enter the data in the following order: sku, quantity, price, and then the product name on a separate line.

For the **Remove** operation, the program should indicate whether the operation was successful or not (if the product was not found).

For the Add and Remove operations, you are not changing the quantity of a product. You are adding (or removing) the information about a product from the inventory (you are adding or removing an element from an array).

For the **Display** operation, display the information for each product on a separate line. The values should line up in columns (use setw). Headers for the table are optional.

For the **Lookup** operation, label the output values (i.e. Name: Apple iPhone 3GS 8GB, etc.). If the product is not found, display an appropriate message.

NOTES:

- This program must be done in a **Linux or Unix** environment, using a command line compiler like g++. Do not use codeblocks, eclipse, or Xcode to compile.
- Use a **partially filled array** of structures to store the inventory**:** Use a counter variable initialized to 0 to track the number of products, and keep all the products at the front of the array, in locations 0..count-1. If you have empty slots in your array, the provided search and sort code will NOT work!
- You MUST use **binary search** for Lookup by sku.
- The program must be **modular**, with significant work done by **functions**. Each function should perform a single, well-defined task (Hint: each menu choice). Also note that some arguments will need to be passed by reference!
- To input the product name (which may contain spaces) use this each time:

  ```
  cin >> ws;       // skips whitespace (newline) after prev input
  getline(cin, productName); // where productName is a string var
  ```

- OR you may require that the productName has no spaces and use cin >> productName; (for a small point deduction).
- You may use (and modify) the code from the book. See the Resources tool in TRACS.
- I will put sample output on the class website in a separate file (output2.txt).
- I recommend implementing the operations in this order: Add, Display, Lookup, Remove. Test to ensure each is correct before implementing the next one.
- Your program **must compile** and run, otherwise you will receive a score of 0.
- Your program must pass **Test Case 0** or you will receive a score of 30 or less with no credit for the other grading categories (correctness/constraints/style). The input values and expected output are in a file called **TC0.txt** on the class website. This test case Adds 3 products, with no spaces in the product names. It then selects the option to display the inventory in a table (it does **not** need to be sorted to pass TC0).

Logistics:

Name your file **assign2_xxxxx.cpp** where xxxxx is your TX State NetID (your txstate.edu email id). The file name should look something like this: assign2_js236.cpp

There are two steps to the turn-in process:

1. Submit an electronic copy using the Assignments tool on the TRACS website for this class.

2. Submit a printout of the source file at the beginning of class, the day the assignment is due. Please **print your name on top of the front page**, and staple if there is more than one page.

See the assignment turn-in policy on the course website (cs.txstate.edu/~js236/cs2308) for more details.