CISC 3110 - Assignment #4 - Creating a Mini-Database Using Arrays

Overview:

What is this assignment about?

> You have recently started a small business that sells smart phones. In the first version of this assignment, you used arrays in C++ to implement some very basic functionality such as: add, update, display, print.

In the second part of this assignment, you will:

1. **Add a menu item allowing the user to delete a phone.**
2. **Allow lookup queries by different criteria.**
3. **Change the productID to be a 4 character C-string. The first 2 should be alphabetical characters (A-Z) and the last 2 should be digits (0-9). You will use C-string functions to compare two productID's. The array of productID's will be a 2D array (viewed as a 1d array of C-strings).**
4. **Use dynamic memory allocation to allocate the arrays, and then double the size of the array if it gets filled. (Note: You can use dynamic memory allocation for only one array and you will still get full credit. E.C. to use for all arrays.)**
5. **Add another array of type C++ string to store the names of the phones.**

Necessary Functions

Here are some of the new functions that should be written:

- delete_phone: Given a product number, remove this phone from the database. This function should call find_phone to locate the location of the phone in the array. It should then remove the phone from the array if the quantity on hand is 0. This can be implemented in one of several ways (some choices are: (a)you can shift over everything else in the array (b) you can place a special value in the productnum array at that location indicating that it is empty.)

  Some things to think about: you most likely will have to maintain the number of phones in the database. Be sure to update this when both adding and deleting. If you use option b, you will have to change your add_phone function to find an empty entry in the array.

- query_database: This function should have a submenu allowing the user to query the database by product number, name, or price. If the user chooses product number or name, you will call find_phone and then display_phone.If the user chooses price, allow them to enter a range of prices.

- grow_array: This function will enlarge the size of the array using dynamic memory allocation. (Hint: we wrote a similar function in class.)

Error Checking

Some new things to check for: if the user tries to delete a phone that is not in your database, or if the user enters an invalid product number.