CIS 22A: Beginning Programming Methodologies in C++

Final Exam

170 points

You have two hours to do the final test. It's open book. If two finals are similar, both will get a zero.

Write a program to create a list of products by storing their information in the following arrays:

- itemNum: an array of long integers to hold the product numbers.
- itemName: an array of string to hold the product names.
- quantity: an array of integers to hold each product's quantity.
- unitPrice: an array of double to hold each product's unit price.
- totalPrice: an array of double to hold each product's total price (unit price x quantity).

The program should relate the data in each array through the subscripts. For example, the number in element 0 of the itemName array should be the name of the product whose number is stored in element 0 of the itemNum array. Similarly with the other arrays. Create a constant for the array size to be used by all the arrays.

When the program is run, it first calls a function to display a menu as follows:

```
=====================================================

                List of Items

1. Add Item
2. Update Item
3. Retrieve Item
4. List all
5. Quit the Program

Enter your choice:
```

There are 6 functions:

- showMenu: display the menu above.
- addItem: when 1 is selected, this function asks the user to enter the product number, name, quantity, and unit price. They are stored in their respective arrays. Keep track of the total number of products entered (how many elements have values) to be used by the listItem function. After each input, ask the user if he/she wants to add more. If yes, make sure that the total number of inputs does not exceed the array size before proceeding to ask the user to enter again. If no, return to display the menu.
- updateItem: when 2 is selected, this function asks the user to enter the product number. It searches for the number in the itemNum array. If found, display the number and name, and ask the user to enter the quantity and unit price.
- retrieveItem: when 3 is selected, this function asks the user to enter the product number. It searches for the number in the itemNum array. If found, display the number, name, quantity, unit price, and total price.
- listItem: when 4 is selected, this function traverses the arrays and displays each product information (number, name, quantity, unit price, and total price). Make sure to display only the elements that have values. Do not use the whole array size if the arrays are not filled up. Make sure to format the output as shown below.
- searchList: this function performs a linear search on an integer array. It is called by updateItem and retrieveItem functions.

Input Validation: Accept only value between 1 and 5 for the menu choice. The quantity must be positive. The unit price must be at least 0.10.

Do not use global variables. You can use a global constant if you want. That's all. Learn to pass by reference if necessary. Do not pass reference everything because you like to. Pass by reference only when the variale is modified. Do not use break, continue, exit and return from an if (the only return is from a function. So only one return and not twice or several in a function).

The display should look like the following (make sure to format it correctly):

```
==================================================

                List of Items

1. Add Item
2. Update Item
3. Retrieve Item
4. List all
5. Quit the Program

Enter your choice: 1

Enter the requested information for each Item.

Enter Item #: 1234567
Enter name: Gatorade
Enter quantity: 10
Enter unit price: $0.77
Do you want to add more (Y or y): y
Enter Item #: 3456789
Enter name: Ritz Crackers
Enter quantity: 15
Enter unit price: $1.88
Do you want to add more (Y or y): n
==================================================

                List of Items

1. Add Item
2. Update Item
3. Retrieve Item
4. List all
5. Quit the Program

Enter your choice: 4
-------------------------
List of all the Items
-------------------------

Item #1234567
Name:          Gatorade
Quantity:      10
Unit price:  $  0.77
Total price: $  7.70


Item #3456789
Name:          Ritz Crackers
Quantity:      15
Unit price:  $  1.88
Total price: $ 28.20


==================================================
                List of Items

1. Add Item
2. Update Item
3. Retrieve Item
4. List all
5. Quit the Program

Enter your choice: 5



==================================================

                List of Items

1. Add Item
2. Update Item
3. Retrieve Item
4. List all
5. Quit the Program

Enter your choice: 2
Enter the Item #: 1234
-------------------------

You look for item #1234.
Sorry we cannot find the Item.Please try again.

==================================================

                List of Items

1. Add Item
2. Update Item
3. Retrieve Item
4. List all
5. Quit the Program

Enter your choice: 2
Enter the Item #: 1234567
Please provide the new data for:
Item #1234567
Name: Gatorade

Enter new quantity: 20
Enter the new price: $0.65
==================================================

                List of Items

1. Add Item
2. Update Item
3. Retrieve Item
4. List all
5. Quit the Program

Enter your choice: 3
Enter the Item #: 1234567
-----------------------------
Retrieve Item information
-----------------------------
```

```
Item #1234567
Name:          Gatorade
Quantity:      30
Unit price:  $  0.65
Total price: $ 19.50


==================================================

                List of Items

1. Add Item
2. Update Item
3. Retrieve Item
4. List all
5. Quit the Program

Enter your choice: 5
```

Include in your program the following comments:

- Name of program
- Programmer's name
- Brief description of the program
- Comments for all the statements

- Optional (I believe you should do it, but it's OK if you don't do it): Create an IPO chart. List the input, processing, and output items.
- Optional: Plan the algorithm using pseudocode.
- Save the program in a filename LastFirst_final.cpp
- Compile and run the program.
- Take a screen shot of the output. Save it in a filename LastFirst_final.doc (e.g. DoeJoe_final.doc)
- Submit LastFirst_final.cpp and LastFirst_final.doc by uploading them in Catalyst.

```
There are 6 functions including main.
main: 15
showMenu: 10
addItem: 40
updateItem: 30
retrieveItem: 30
listItem: 30
searchList: 15
total: 170 points.
```