Objectives:
- Arrays
- Strings
- Pointers
- Functions
- File I/O

REMINDERS and GRADING CRITERIA:
➢ Read these sections from lab1... they haven’t changed.

LAB DESCRIPTION

PART 1. BE AN ENTREPRENUER (50%) Mandatory filename → lab2p1.c

PROBLEM:
You own a company and your goal as a business owner (as always) is to make a profit. You buy products at a wholesale price and sell them at a retail price. Each product is associated with a supply group i.e. a supply type.

For instance, if you sell groceries, your supply order might consist of some fruits, like apples, oranges and mangos; some dairy products like milk, sour cream, cream cheese, and cheddar cheese; and some meat including hamburger, pork loin, chicken, and steak. This example input has 3 different supply groups (fruit, dairy product and meat) with individual products within each group. Along with the supply type and product name, the following is also included:
- The wholesale price of the product
- The quantity of that item bought at the wholesale price by the owner
- The retail price of the product
- The quantity of that item bought at the retail price by the customer

INPUT:
Filename: lab2p1in
One record of information looks like:
- Supply Type = Alphabetic field with a maximum of 15 characters
- Product Name = Alphabetic field with a maximum of 15 characters
- Wholesale Price = Numeric field with 2 decimal points; maximum value 999.99
- Quantity of Wholesale Product bought by owner = an integer from 0 to less than 1000
- Retail Price = Numeric field with 2 decimal points; maximum value 9999.99
- Retail Product Quantity sold to customers = an integer value from 0 to less than 1000 where each field is separated by a blank and each record ends with a newline character.

Sample Input (spacing is not exactly as required above):
<table>
<thead>
<tr>
<th>SupType</th>
<th>ProdName</th>
<th>WholesalePrice</th>
<th>WholQuant</th>
<th>RetailPrice</th>
<th>RetailQuant</th>
</tr>
</thead>
<tbody>
<tr>
<td>SupType1</td>
<td>ProdNameA</td>
<td>0.23</td>
<td>15</td>
<td>0.50</td>
<td>7</td>
</tr>
<tr>
<td>SupType1</td>
<td>ProdNameB</td>
<td>1.25</td>
<td>10</td>
<td>2.25</td>
<td>8</td>
</tr>
<tr>
<td>SupType1</td>
<td>ProdNameC</td>
<td>0.89</td>
<td>20</td>
<td>1.99</td>
<td>18</td>
</tr>
<tr>
<td>SupType2</td>
<td>ProdNameD</td>
<td>2.05</td>
<td>8</td>
<td>4.25</td>
<td>5</td>
</tr>
<tr>
<td>SupType2</td>
<td>ProdNameE</td>
<td>1.75</td>
<td>12</td>
<td>2.99</td>
<td>10</td>
</tr>
</tbody>
</table>
OUTPUT:
Filename: lab2p1out
Calculate the following:

Output per Type:
- The amount of money the owner spent buying each type of wholesale product
- The amount of money spent by the customers buying each type of retail product
- The amount of profit made by the owner (retail costs – whole sale costs) per type

Output once:
- The amount of money the owner spent buying all of the wholesale products
- The amount of money spent by the customers buying all of the retail products
- The total amount of profit made by the owner (retail costs – whole sale costs)

Sample Output:
Type: SupType1
Owner Cost: $33.75
Customer Cost: $57.32
Profit: $23.57

Type: SupType2
Owner Cost: $37.40
Customer Cost: $51.15
Profit: $13.75

TOTAL
Owner Total Cost: $71.15
Customer Total Cost: $108.47
Profit: $37.32

CONSTRAINTS:
- The maximum number of products is 50.
- You can assume that the Supplier Type and Product Name do *NOT* have spaces in them.
- Be sure to use at least two functions in this assignment (your choice) and completely document what each function is doing, describing the necessary arguments/parameters being passed and how they are being passed. Be sure to use at least one parameter as a pointer type of some kind in one of the two functions.
PART 2. SIMPLE CIPHER (50%) Mandatory filename → lab2p2.c

PROBLEM:
- You have a message that you want to encode. The scheme to encode the message is to reverse each “word” in the phrase where words are separated by one or more non-alphabetic characters; and keep each non-alphabetic character in their original position. Thus, the output string and the input string should be the same length.
- You will need to use the underscore character ‘_’ instead of a blank for the input in order to read it in as a single text value. However, when you create your encoded output, be sure to convert the underscore to a blank.
- In addition, output all alphabetic characters as upper case letters.

INPUT/OUTPUT:
- Input file name: lab2p2in
- Output file name: lab2p2out
- Example Input → Output:
  War_eagle → RAW ELGAE
  Reading_records_of_variable_length? → GNIDAER SDROCER FO ELBAIRAV HTGNEL?
  Have_fun._Doing_this_lab_:) → EVAH NUF. GNIOD SIHT BAL :)
  Words%end*with^non-alpha...characters!!! → SDROW%DNE*HTIW^NON-AHLPA...SRETCARAHC!!

CONSTRAINTS:
- The length of the input message to be encoded will not exceed 100 characters, including the new line character (after all, texting has a character limit so why can’t we?)
- Do not use subscripts for this lab; use pointers to manipulate the location of each character in the array(s)

LAB SUBMISSION

Read this section from lab1... it hasn’t changed except for:

The labname for this lab is: lab2
Be sure to submit the following files: lab2p1.c lab2p2.c README