CPE 100 – Introduction to Computer Programming
Sections C and D August 2019
Laboratory Exercise 5

Objective

This lab is intended to give you practice using simple arrays and command line arguments, as well as reviewing loops.

Instructions

Write a program called ***shopping.c*** which acts like a supermarket cash register.

This program should do the following:

1. Ask the user how many items they have in their shopping basket. If they have more than 20 items, print a message saying that the program cannot handle this many items, and exit the program.

2. For each item that they have, ask them to enter the price of that item.

3. Read the price from the terminal. Convert it to a double. Then store it in an array of prices (*Hint: this array should be declared to be an array of doubles with 20 elements)*.

4. When the user has finished entering prices, print out a table (using ***printf***) of the prices and the corresponding VAT. This table should have three columns: Price, VAT and Total (price plus VAT). You should print one row for each item in the user's shopping cart. See the example below.

> The rules for calculating the VAT are as follows: If the price is greater than or equal to 100 baht, the VAT is 7% (price multiplied by 0.07). Otherwise the VAT is zero.

5. After you have printed one row for each item stored in your array, print a row of dashes, and then print a row of totals: the total of all the prices (without VAT), the total of all the VAT amounts, and the grand total. *(Hint: you can calculate these totals while you are printing the rows for the individual items.)*

After you have written the code, compile and test your program. Make sure that you test the boundary cases (number of items is 20, number of items greater than 20). Make sure you try some cases where there are no prices greater than 100 baht and other cases where some prices are greater than 100 baht.

Hints: This program will have two separate loops. The first loop will get the prices from the user and store them in the array. The second loop will process each stored price to compute the VAT and display a row in the table. You only need ***one array*** *to solve this problem.*

Sample Output

```
   Price       VAT    Item Total
---------- ---------- ----------
    230.00      16.10      246.10
     15.00       0.00       15.00
     22.50       0.00       22.50
    310.00      21.70      331.70
     99.00       0.00       99.00
---------- ---------- ----------
    676.50      37.80      714.30
```

Now copy ***shopping.c*** to a new C file ***shopping2.c***. Modify ***shopping2.c*** to use command line arguments to pass two values: the number of items in the basket and the VAT percentage. You will run ***shopping2.c*** as follows:

```
./shopping2 8 5
```

The first number is the item count. The second number is the VAT percent.

Your new program needs some new logic at the beginning to process the command line arguments:

1. Check the value of *argc.* If it is less than 3, give an error message and exit. This means the user did not provide both the item count and the VAT percentage.
2. Transform *argv[1]* (the item count) into an integer. You can use ***sscanf*** for this as shown in our command-line version of ***multiples2.c*** in this week's demos. Store this in the same variable you used for the item count in your original version.
3. Transform *argv[2]* (the VAT percentage) into a double. You will need a new variable for the VAT percentage.
4. Check to make sure that the VAT percentage is between 0 and 10 (0% - no VAT – to 10%). If it is not, give an error message and exit.
5. Check to make sure that the item count is less than or equal to 20. You should already have this code in your program.

Be sure that you remove the code that asks the user for the item count. Also you need to change the calculation of VAT to use your new variable rather than a constant value of 7%. Your program should still only charge VAT on items that cost 100 baht or more.

Compile and test ***shopping2.c.*** Be sure test the error conditions:

- No arguments
- Only one argument
- Item count negative
- Item count greater than 20
- VAT less than 0
- VAT greater than 10

Upload both ***shopping.c*** and ***shopping2.c***.