ENGR 0012 – Spring 2019
Project 2

Acceptable behaviors for this assignment include:	Unacceptable behaviors for this assignment include:
• Asking your team members • Asking your professor or TA • Discussion with other teams/students is acceptable so long as no code is directly shared.	• Copying the solution(s) from a solution manual, book, other written material, or from other students • Providing the solutions to a classmate, student in other section, student in future section, or online solution banks

Introduction:

You, Gomy, are a secret undercover operative working at the CIA's Cyber-Security Division. Recently, the CIA has acquired valuable information about increasingly suspicious behavior going in Pittsburgh. As you are highly skilled in programming, your mission is to design a decoy e-shopping portal to gather information about this suspicious behavior. This e-shopping portal will be selling various electronic components. Any customer who tries to buy some components with certain quantities should be reported back to the CIA. More information will be provided later.

Overall Instructions

- You will create several functions, including functions that receive and manipulate arrays!
- You are provided with a file with the following structure (file name: inventory.txt):

```
Resistors       8       2.29    1.99
Capacitors      12      1.19    0.99
Inductors       20      4.99    3.99
Timers  11      2.99    1.99
LEDs    29      1.29    0.89
ICs     5       10.79   9.99
```

- The file contains four columns: the first column contains the name of the electronic component, the second column the available quantity of the electronic component, the third column contains the selling price, and the fourth column contains the discounted sale price.
- Each column is separated by a space. There are exactly 6 rows or 6 electronic components available to purchase this season. Assume all files presented will maintain this structure.
- Include a comment with your team number and team member names at the top.
- Include comments, indentation, and whitespace so that your program is neat and understandable to anyone who reads it.

Main function

1. Include this integrity statement as a comment in your code: "We in team (team number) certify that we have completed this assignment in an honest manner."
2. Print the following header once:
   Welcome to CyberShop, the one stop for all your electronic components.
   Customer Satisfaction is our priority!
3. Ask the user for the file name. Check if the file exists and repeat infinite times until the correct file is entered. Once the correct file is selected, print: "File Selected: FileName"
4. Read the contents of the file using EOF to find the end of file and store them into arrays, one for each column.
   a. To store an array of strings, create a two-dimensional character array. First dimension is the number of strings, select a large number, say 10. Second dimension is the number of characters per string, again select a large number, say 20. Example: char comp[10][20];
   b. To scan a string into the first index position, simply use comp[0] as the receiving variable in the scanf function. Similarly, to display the first string, use conversion character %s and corresponding value as comp[0]. %s stops at the \0 character, which is automatically added to the end of a string.
   c. To send a two-dimensional array to a function, the receiving variable should be defined with the same dimensions as initialized earlier. Example: void myfunk(char comp[10][20]){ function block }

5. Send number of components, array of component names, array of quantities, array of selling price, and array of discounted sale price to function **printmenu**. Details about this function can be found under the function header below.
6. Send number of components to function **item_choice**, which returns the item number chosen by the user to the main function. Details about this function can be found under the function header below.
7. Send the corresponding item quantity to function **quant_choice**, which returns the quantity of that item the user wants to purchase back to the main function. Details about this function can be found under the function header below.
8. Send the item number, selected quantity, and the array containing all components quantities, to a function named **update**. This function updates all quantities to reflect what's available after purchase. Details about this function can be found under the function header below.
9. In the main function, update the customer's basket to reflect which components and quantities they have selected so far.
10. Repeat steps 4-8 if requested by the user.
    a. Conduct the choice check by invoking function **uchoice**, which receives no value from the main function but returns y or n (only lower case) back. Details about this function can be found under the function header below.
    b. Error check, so that customer can only enter Y, y, N, or n. For all other characters, repeat the prompt indefinitely.
11. When the customer quits, display the receipt as shown in the screen shot below.
    a. Hint: Use a loop to run through the customer's basket. Skip components with 0 quantity purchased and print those that have some quantity.
    b. For each component that was bought, calculate and display the total amount spent on that component (use the discounted sale price) and the total quantity bought per component.
    c. Then display overall (total) amount spent on all components.
12. Ask the user for a credit card number, then pass it to a function called **checkout,** which returns 1 if the checkout was successful, otherwise it returns 0. Details about this function can found under the function header below.
    a. This number should read as a string (i.e. use %s with scanf to read the credit card number).
    b. Use a loop when calling this function. You should keep calling it until the checkout is successful.
    c. Print "Invalid number", each time the function returns 0.
13. Finally, A suspect customer should be reported back to the CIA if he/she bought the following items with at least the mentioned quantities:
    a. 3 Resistors.
    b. 2 Capacitors.
    c. 1 Timer.
    d. 1 IC.
14. Create a new file named **Suspects.txt** that contains the credit card number of the suspect along with the items and quantities bought.

printmenu function

- Function Input: 5
- Function Output: 0
- Accepts number of components, array of component names, array of quantities, array of selling price, and array of discounted sale price from main().
- Displays this as a menu properly formatted.
- Items with zero quantity should NOT be printed out to the user.

item_choice function

- Function Input: 1
- Function Output: 1
- Accepts number of components from main().
- Prompts the user to enter item # and error checks item # is available in menu indefinitely.
- Returns item # chosen.

quant_choice function

- Function Input: 1
- Function Output: 1
- Accepts available quantities for item# selected from main()
- Prompts and accepts a quantity from user, error checks until correctly entered.
- Returns quantity selected back.

update function

- Function Input: 3
- Function Output: 0
- Accepts item number selected by user, quantity entered by user, and array of all component quantities from main()
- Updates appropriate item quantity in function.

uchoice function

- Function Input: 0
- Function Output: 1
- Prompts the user to enter y or n.
- Error checks indefinitely.
- Returns lower case char back to main()

checkout function

- Function Input: 1
- Function Output: 1
- Accepts the credit card number as an array of characters from main().
- Check the input to be exactly 16 numeric digits and do the checksum test (see the Appendix).
  If any of these tests fails return 0, otherwise return 1.

Appendix:

How to check the credit card number and make sure it's valid:

1. Verifying a 16-digit card number starts by taking the first 15 digits, which are the institution code and the individual account identifier. For example, in the card number 4578 4230 1376 9219, those digits would be:

   4-5-7-8-4-2-3-0-1-3-7-6-9-2-1

2. Starting with the first digit, multiply every second digit by 2:

   8-5-14-8-8-2-6-0-2-3-14-6-18-2-2

3. Every time you have a two-digit number, just add those digits together for a one-digit result (see the Appendix):

   8-5-5-8-8-2-6-0-2-3-5-6-9-2-2

4. Finally, add all the numbers together:

   8 + 5 + 5 + 8 + 8 + 2 + 6 + 0 + 2 + 3 + 5 + 6 + 9 + 2 + 2 = 71

5. When this number is added to the check digit, then the result must be an even multiple of 10. In this case:

   71 + 9 = 80

   The number is therefore valid. If the algorithm doesn't produce a multiple of 10, then the card number cannot be valid.

Hint 1:

Characters in C are encoded using ASCII code. The ASCII codes representing characters '0'- '9' are 48 – 57.
To convert a digit character to integer, you need to subtract the character '0' from it.
Code Example:

```
void main(void){
        char c = '7';
        int x;

        printf("the character is: %c\n", c);

        x = c - '0';

        printf("the corresponding digit is: %d\n", x);
}
```

Hint 2:

To sum the two digits of a 2-digit number, try to use integer division and modulus.

Program output should look something like this:

```
Welcome to CyberShop, the one stop for all your electronic components.
Customer Satisfaction is our priority!
Please enter file name: invenory.txt
Please enter file name: inventory.txt


File selected: "inventory.txt"
*****************************
Item       Component        Quantity          Price          Sale
   0       Resistors               8    $      2.29     $      1.99
   1      Capacitors              12    $      1.19     $      0.99
   2       Inductors              20    $      4.99     $      3.99
   3          Timers              11    $      2.99     $      1.99
   4            LEDs              29    $      1.29     $      0.89
   5             ICs               5    $     10.79     $      9.99
*****************************
Please enter item to purchase: -1
Please enter item to purchase: 19
Please enter item to purchase: 0
Please enter quantity [8 available]: 30
Please enter quantity [8 available]: -1
Please enter quantity [8 available]: 8
Buy more components? (y/n): H
Buy more components? (y/n): Y
*****************************
Item       Component        Quantity          Price          Sale
   1      Capacitors              12    $      1.19     $      0.99
   2       Inductors              20    $      4.99     $      3.99
   3          Timers              11    $      2.99     $      1.99
   4            LEDs              29    $      1.29     $      0.89
   5             ICs               5    $     10.79     $      9.99
*****************************
Please enter item to purchase: 1
Please enter quantity [12 available]: 3
Buy more components? (y/n): y
*****************************
Item       Component        Quantity          Price          Sale
   1      Capacitors               9    $      1.19     $      0.99
   2       Inductors              20    $      4.99     $      3.99
   3          Timers              11    $      2.99     $      1.99
   4            LEDs              29    $      1.29     $      0.89
   5             ICs               5    $     10.79     $      9.99
*****************************
Please enter item to purchase: 2
Please enter quantity [20 available]: 6
Buy more components? (y/n): N
Thank you for Shopping


Receipt
Purchased 8 Resistors for a total of    $15.92
Purchased 3 Capacitors for a total of   $2.97
Purchased 6 Inductors for a total of    $23.94
                Total = $42.83

Please enter your credit card number: 123

Invalid number!
Please enter your credit card number: 12345678915935728965

Invalid number!
Please enter your credit card number: 1234567891593572

Invalid number!
Please enter your credit card number: 4578423013769219

Checkout Complete
```

Sample of the Suspects.txt file

```
Credit Card number: 4578423013769219
Components bought:
Resistors 3
Capacitors 3
Inductors 3
Timers 1
ICs 2
```

Graded on (partial grades available):

- Code formatting and comments
- Functionality (does the code run)
- Match to screen shot (does display formatting match)

- Accuracy (is displayed data correct)
- Code features (does the code follow directions, including use of functions when required)

SUBMISSION INSTRUCTIONS:

By submitting code, you are stating that:

1. The code was generated wholly by your team.
2. No part of the code was plagiarized, copied from a source, or discussed on a forum or online discussion board.
3. You did not help or receive help in writing code from anyone outside your team, TAs, and instructors.
4. Peer discussion of the project did not include sharing code snippets.

Integrity violations will be dealt with following First-Year Program and SSOE regulations.

Name your C file: InstructorName_Time_Project2_teamNo.c
e.g. Mahmoud_4PM_Project2_TeamL01.c

Submit your copies using the Assignment Submission link found on desktop computers in classrooms. Select Homework link under your class instructor link.

N.B.: This file will self-destruct in 5 mins after being displayed. Good Luck, Gomy!