CSE 5A  
Introduction To Programming (C/C++)

Homework 6

Read Chapter 8

Due: Friday, November 16 by 6:00pm

Create a new folder named HW6 (note uppercase HW6) in your cs5f@ieng9 Home Directory. With Dev-C++, create a new source file - file type: C source (*.c) - named hw6.c (note lowercase hw6.c) saved in the new HW6 folder. Feel free to ask the tutors on duty in the lab for help.

hw6.c will be a program using several functions, two arrays of integers representing how many bills and coins of various denominations are required for change, and some simple calculations and casts. The program will prompt the user to enter the number of items to be purchased, the cost of each item, display an invoice based on the values input by the user, prompt the user for the payment amount, calculate the change owed the user, and display the change as the number of various dollar bill denominations and various coin denominations. Some error checking will be required as outlined below. Your output is expected to be the same as the example hw6.exe executable provided in the public/Examples directory.

Let us start with an example execution of the program (user input is in bold):

Welcome to the Freud Store!

How many plush stuffed Pavlov's Dogs would you like to buy? 4

How much do you think your mother would pay for each one?
Dollars: 12
Cents: 55

Your Invoice Statement:
  Subtotal: $ 50.20
  Tax: $ 3.51
  ----------
  Total: $ 53.71

How much money are you giving me to pay for this invoice?
Dollars: 60
Cents: 75

Your change is $7.04:
1 $5 bill(s)
2 $1 bill(s)
4 Penny coin(s)

The first input must be an integer greater than or equal to 0.
The second set of inputs will be read as two ints (Dollars and Cents) that must be greater than or equal to 0.
The third set of inputs will be read as two ints that must be greater than or equal to the Invoice Total dollars and cents (see below for details).

The dollar bill denominations will be: $100, $50, $20, $10, $5, $1
The coin denominations will be: Half Dollar (50¢), Quarter (25¢), Dime (10¢), Nickel (5¢), Penny (1¢)
The number of each denomination will be maintained in the arrays bills[] and coins[], respectively.
General outline of your program
(See the output above and the sample executable for exact wording of prompts and other output/format.)

0) START EARLY!!!    I Really Mean It!!!

1) Display the store greeting. Prompt the user to enter the number of plush Pavlov's Dogs to purchase. Call the function `readInteger()` to get the user's input. The minimum value expected is 0. See below for the details of `readInteger()`.

2) Prompt the user for the unit cost of these puppies, first the Dollars then the Cents. Call the function `readInteger()` to get the user's Dollars input and again to get the user's Cents input. The minimum value expected for both Dollars and Cents is 0. See below for the details of `readInteger()`. Keep track of the cost as total number of cents (dollars * 100 + cents). One dollar = 100 cents.

3) Calculate the subtotal (number of items * cost per item in cents), tax (subtotal * tax rate [use 7% with no rounding – #define TAX_RATE_PERCENT 7]) or `subtotal * TAX_RATE_PERCENT / 100`), and total (subtotal + tax). All values should be in terms of cents. Display a formatted invoice statement calling the function `displayInvoice()` passing the calculated subtotal, tax, and total as parameters. See below for the details of `displayInvoice()`.

4) Prompt the user for the amount of the payment for this invoice. Call the function `readInteger()` to get the user's Dollar input and again to get the user's Cents input. The minimum value expected is the invoice total such that:
   
   Minimum dollars expected = total / 100
   
   if ( dollars entered > dollars owed )
   
   Minimum cents expected = 0
   
   else
   
   Minimum cents expected = (total - dollars entered)
   
Remember, you are keeping track of all the values as cents. 100 cents = 1 dollar.

5) Calculate the change (in cents) to be returned to the user based on the payment amount and the invoice total (payment in cents – invoice total in cents). From this change amount, calculate the number of `dollars` (change in cents / 100) and calculate the number of `cents` (change % 100).

6) Call the function `makeChangeInBills()` to determine how many of the various dollar bill denominations and call the function `makeChangeInCoins()` to determine how many of the various coin denominations are needed for the change. See below for the details of `makeChangeInBills()` and `makeChangeInCoins()`.

7) Output the change as the two integers change in dollars and change in cents [%d.%02d]. Call `displayChangeInBills()` to display the number of various dollar bill denominations and call `displayChangeInCoins()` to display the number of various coin denominations needed for the change. See below for the details of `displayChangeInBills()` and `displayChangeInCoins()`.

Here are the function prototypes you must use (besides your main() function) for the functions you write:

```c
int readInteger( int minValue );
void displayInvoice( int subTotal, int tax, int total );
void makeChangeInBills( int bills[], int dollars );
void makeChangeInCoins( int coins[], int cents );
void displayChangeInBills( int bills[] );
void displayChangeInCoins( int coins[] );
```
NOTE: Please follow the instructions on how to write this program. This will make grading much easier. The main purpose of this assignment is to learn more about using arrays and functions.

Please also go to a Discussion Section to learn more about how to complete this programming assignment.

**Function Details**

**int readInteger( int minValue );**
This function will read an int from the standard input with scanf(). You can assume only valid integers will be entered. You must check to see if the integer value entered by the user is at least as large as the minValue passed as the parameter. If it is not, then in a loop output an indented (tabbed) error message and read an int again. Keep looping until the user enters a value that is >= minValue.

For example, calling readInteger() passing the value 0 as the parameter in Step 1) above:

Welcome to the Freud Store!
How many plush stuffed Pavlov's Dogs would you like to buy?
-3

    You must enter a value >= 0
    Try again: -2

    You must enter a value >= 0
    Try again: -1

    You must enter a value >= 0
    Try again: 0

... Rest of program ...

NOTE: The value of minValue is used as the value to display in the line "You must enter a value >= XXX". This value will be different depending on how readInteger() is called – especially when reading in the payment dollars and cents.

**void displayInvoice( int subTotal, int tax, int total );**
This function outputs the subtotal, tax, and total cost in a nicely formatted invoice-like manner. Use a tab for the initial indenting, additional appropriate spaces, and $%6d.%02d for each of the monetary format specifiers.

**void makeChangeInBills( int bills[], int dollars );**
This function calculates how many of the various dollar bill denominations are needed for the dollar part of the change to be given back to the user. bills[0] represents $100 bills, bills[1] represents $50 bills, bills[2] represents $20 bills, bills[3] represents $10 bills, bills[4] represents $5 bills, and bills[5] represents $1 bills. There are many ways to calculate this. You may use a sequence of integer divides (quotient) and mods (remainder). For example, 357 / 100 = 3 (we need 3 $100 bills – assign this value into bills[0]). 357 % 100 = 57 (there is $57 we still need to determine what lower denomination bills to give as change). 57 / 50 = 1 (we need 1 $50 bill – assign this value into bills[1]). 57 % 50 = 7 (there is $7 we still need to determine what lower denomination bills to give as change). And so on.
void makeChangeInCoins( int coins[], int cents );
This function is similar to makeChangeInBills() except the array coins represents coin denominations vs. dollar
denominations. coins[0] represents 50 cent coins (half dollars), coins[1] represents 25 cent coins (quarters),
coins[2] represents 10 cent coins (dimes), coins[3] represents 5 cent coins (nickels), and coins[4] represents 1
cent coins (pennies). Similar integer divides and mods can be used to set the values in the coins array.

void displayChangeInBills( int bills[] );
This function will display how many of the various dollar bill denominations are part of the change to be
returned to the user. There are many ways to go about doing this. You are free to use sequence, loops, switch
statements. Do not display dollar bill denominations with a zero count. Use a tab (vs. spaces) in the output.

void displayChangeInCoins( int coins[] );
This function will display how many of the various coin denominations are part of the change to be returned to
the user. There are many ways to go about doing this. You are free to use sequence, loops, switch statements.
Do not display coin denominations with a zero count. Use a tab (vs. spaces) in the output.

We will automatically collect your HW6/hw6.c files from your cs5f course specific home directories on ieng9
at 6pm on Friday, Nov. 16. Make sure your HW6 project is stored in your cs5f home directory folder. If you
have any questions/problems with this, see one of the tutors or the instructor.

Follow the commenting style outlined in the example full listing attached to the Functions notes. In particular, a
file header comment is required, each function requires a header comment, and each logical block of code
requires a block comment.

Follow proper indentation and coding style as outlined in class. Use judicious use of #define statements instead
of hardcoded.

You will use a good part of this project for the next HW assignment, so get it right now and HW7 will be
pretty easy!

I will place a sample executable in the Class Resources / cs5f Public directory under Examples folder

hw6.exe

START EARLY!!!
And have fun!

Early Turnin Extra Credit: 5% Extra Credit for being finished with this assignment by

6:00pm Thursday, Nov. 15.