**Topic(s):**      I/O & Functions in C
**Date assigned:** Wednesday, September 1, 2010
**Date due:**      Wednesday, September 8, 2010
**Points:**        15

You have now had a full year of C++ and we are now going to move to the world of Data Structures in C. The transition to C is not that difficult but can be frustrating. For this first assignment, I would like to introduce you to the differences of I/O in C and C++ while reinforcing the use of functions when writing code.

**Sample Problem:**

Write a complete C++ (fully documented) program that allows the user the ability to input from the keyboard the amount of a purchase and the amount received in payment (both amounts in cents). Compute the change in dollars, half-dollars, quarters, dimes, nickels, and pennies using the highest amounts of each starting with the dollars. As output, you are to print the number of dollars, half-dollars, quarters, dimes, nickels, and pennies given back as change.

**Sample Solution:**

```c
/*****************************************************************************
 File name:  01ChangeMaker.c
 Author:     Doug Ryan
 Date:       9/1/10
 Class:      CS300
 Assignment: Sample Problem
 Purpose:    This program shows a simple C program that calculates change from
             a purchase price and amount received. The change is in the form
             of dollars, half-dollars, quarters, dimes, nickels, and pennies.
 *****************************************************************************/


#include <stdio.h>
#include <stdlib.h>

const int DOLLARS = 100;
const int HALF_DOLLARS = 50;
const int QUARTERS = 25;
const int DIMES = 10;
const int NICKELS = 5;
const int PENNIES = 1;

int main (void)
{
   int purchasePrice;
   int amountReceived;
   int change;

   printf ("**************************************************");

   do
   {
     printf ("\nEnter the purchase price (in cents): ");
     scanf ("%d", &purchasePrice);
   } while (purchasePrice <= 0);

   do
```

```c
   {
      printf ("\nEnter the amount received (in cents): ");
      scanf ("%d", &amountReceived);
   } while (amountReceived <= 0);

   change = amountReceived - purchasePrice;

   if (change < 0)
   {
      printf ("\nYou owe more money");
   }
   else
   {
      printf ("\nYour change is:\n");

      printf ("\nDollars = %d", change / DOLLARS);
      change %= DOLLARS;

      printf ("\nHalf-dollars = %d", change / HALF_DOLLARS);
      change %= HALF_DOLLARS;

      printf ("\nQuarters = %d", change / QUARTERS);
      change %= QUARTERS;

      printf ("\nDimes = %d", change / DIMES);
      change %= DIMES;

      printf ("\nNickels = %d", change / NICKELS);
      change %= NICKELS;

      printf ("\nPennies = %d", change);
   }
   printf ("\n*************************************************");

   return EXIT_SUCCESS;
}
```

**Your Problem:**

The above solution was written by someone not comfortable with functions. You are to modify the above solution so that the solution is correctly written using functions. Place the function prototypes above int main (void) and place the function definitions below the main function. You must determine what functions to write. Think about reusability and also look for patterns of repetition in the code.

**Notes:**

1. Your solution is to look and work EXACTLY the same as the sample solution above.
2. We will use the coding guidelines coding.C.v6.pdf found on the CS300 home page.
3. Your code is to be written in C using Geany and tested on zeus. Programs written in other environments will not be graded. We will be using a submit script for submitting programs. I will talk about using the submit script in class and we will submit a sample program in class.
4. Function documentation can be found in the coding standards document. Make sure you follow the documentation exactly.

Modified: 9/8/10

To use the submit script the command is:

submit classname filename

EXAMPLE
-------
submit cs300f10 ryan.c.tar.gz which will submit the zipped up tar
file ryan.c.tar.gz

Once you have successfully submitted a file you will get a receipt
which is a file that will end in .receipt. You can make sure your
file was submitted correctly by typing the command:

checkReceipt submittedfile classname receiptfile

EXAMPLE
-------
checkReceipt ryan.c.tar.gz cs300f10 ryan.c.tar.gz.cs300f10.receipt

will produce the result:

HASH>>>y)�� �,=�↙��D�"�,m�
HASH>>>y)�� �,=�↙��D�"�,m
 SUCCESS! Your receipt is valid

DO NOT MOVE OR MODIFY THE SUBMITTED FILE "OR" THE RECEIPT FILE OR THE
CHECK RECEIPT COMMAND WILL NOT BE SUCCESSFUL.