

Assignment #1

Topic(s): I/O & Functions in C
Date assigned: Wednesday, August 31, 2011
Date due: Wednesday, September 7, 2011
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. 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.

You are to rewrite the sample solution below using functions.

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:

```
/*
*****
File name: 01ChangeMaker.c
Author: Doug Ryan
Date: 9/1/10
Edited: Chadd Williams
Edited Date:7/22/11
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
{
    printf("\nEnter the amount received (in cents): ");
    scanf("%d", &amountReceived);
} while (amountReceived <= 0);

change = amountReceived - purchasePrice;

printf("\n*****");
printf("\nYour purchase price is:\n");
printf("\nDollars = %d", purchasePrice / DOLLARS);
purchasePrice %= DOLLARS;
printf("\nHalf-dollars = %d", purchasePrice / HALF_DOLLARS);
purchasePrice %= HALF_DOLLARS;
printf("\nQuarters = %d", purchasePrice / QUARTERS);
purchasePrice %= QUARTERS;
printf("\nDimes = %d", purchasePrice / DIMES);
purchasePrice %= DIMES;
printf("\nNickels = %d", purchasePrice / NICKELS);
purchasePrice %= NICKELS;
printf("\nPennies = %d", purchasePrice);

printf("\n*****");
printf("\nYour payment is:\n");
printf("\nDollars = %d", amountReceived / DOLLARS);
amountReceived %= DOLLARS;
printf("\nHalf-dollars = %d", amountReceived / HALF_DOLLARS);
amountReceived %= HALF_DOLLARS;
printf("\nQuarters = %d", amountReceived / QUARTERS);
amountReceived %= QUARTERS;
printf("\nDimes = %d", amountReceived / DIMES);
amountReceived %= DIMES;
printf("\nNickels = %d", amountReceived / NICKELS);
amountReceived %= NICKELS;
printf("\nPennies = %d", amountReceived);

printf("\n*****");
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*****");
printf("%c", '\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. To get started, make a directory named `cs300_1_PUNetID` and place your `01ChangeMaker.c` file in this directory. These names are required.
4. 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.
5. Function documentation can be found in the coding standards document. Make sure you follow the documentation exactly.
6. You must provide a hard copy (color, double-sided, stapled) as well as an electronic copy by 9:15am September 7, 2011.

For testing on Zeus:

How to compile your project from the command line:

```
chadd@zeus:~> gcc -Wall -g -o 01ChangeMaker 01ChangeMaker.c
```

How to run your project from the command line:

```
chadd@zeus:~> ./01ChangeMaker
```