Assignment 2 Intro to C++
Value: 5pts
Due at the beginning of the next session.

This assignment is to use the aspects of C programs that become the primary features of C++: multiple source files, structures (or classes), and pointers. Hand in the files you write, and console output to demonstrate correct operation of your program.

Presume for the sake of the assignment you are working at a coin-op vending company. They have just bought a new coin changer which can report how many coins of each denomination is currently in its inventory, you can dispense any number of available coins of any denomination.

Problem: Write a program to compute change, given the number of coins on hand in each of the following denominations: 100, 25, 10, 5, 1 and the value of the change required.

The program must be capable of determining if enough coins are on hand to make it possible to make change in the amount required before actually giving the change. If inadequate coins are on hand, report an error message. Otherwise print out how many dollar coins, quarters, dimes, nickels and pennies to dispense.

The console might look like:

```
Enter the number of:
Dollars quarters dimes nickels pennies
1 2 3 4 5
That makes:
$2.05 on hand
How much change in pennies should be dispensed?
198
The change is:
1 dollar
2 quarter
3 dimes
3 nickels
3 pennies
$0.07 in change remains on hand
```

Given the previous example would be noted as 1 2 3 4 5, 198 -> 1 2 3 3 3

Test your program with a minimum of the following cases:

```
1 2 3 4 5, 198 -> 1 2 3 3 3
0 3 3 3 3, 79 -> not enough change on hand
0 3 3 3 3, 78 -> 0 3 0 0 3
1 1 1 1 1, 70 -> not enough change on hand
1 0 1 6 9, 38 -> 0 0 1 5 3... etc
```

Background:

Be sure to break your program into multiple source files where the contents of each file are related by the kind of work done. Use structures, pointers, and functions to full effect to make this a trivial program to write.

This assignment has many correct solutions, **one possible approach:** is to design some functions to handle the various sub-problems. The functions might be located in a file named "coins.cpp" and have prototypes such as:

```
Coin *FindCoin(int Value_pennies);  // locate the data about a coin

void GetCoinCount(int Value_pennies); // how many nickels? or whatever

int CurrentInventoryValue(); // in pennies

void PrintCurrentCoinInventory(); // as n dollars m quarters, etc

void PrintCurrentInventoryValue(); // as $x.xx

bool ComputeChange(int ChangeValue_pennies); // true = possible
```

In this case, coins.cpp would probably define a table of coins holding the state of the current coin inventory. There would be one entry for each denomination of coin. The program will be easier to write if the denominations are entered in order, although it could be made to work if the entries were in random order. The table could be an array of structures or classes.

There should be a corresponding "coins.h" file that defines the structure or class that defines a coin, as well as the prototypes for the functions in coins.cpp. The structure should have fields for the current inventory of a coin (the number on hand), the name of the coin denomination, and the value of the coin in pennies. Working in pennies makes it easier.

The function ComputeChange returns true if enough coins are available to make the required change. It must handle using nickels to substitute for dimes or quarters for instance, when there are enough nickels but not enough dimes or quarters, etc. If change can be made, print the required output. If inadequate change is on hand, the function would return false.

"main.cpp" will contain only the code necessary to get input, call the coin functions and print messages. Don't do the work in "main.cpp".