Sonoma State University
Computer Science Department
CS 460 – Fall 2023 – Watts

Exercise 2

(Due date: Monday, 2 October 2023)

For **Exercise 2 Part A,** you are to complete and fully test the implementation of the class “money”.

The set of starter files for this exercise are available in a folder called Exercise2 in the course pickup directory. You should carefully review the contents of these files before you start any coding.

You should place all of your function implementations for the class in the file money.cpp.

You should also write one or more programs to test your class. The name of each program should be *lastname*E2α.cpp where α is a letter between A and Z. (Please use them consecutively.) Each program should be accompanied by an input file called inputE2α and an expected output file called expectedE2α. You may use ChatGPT or other AI agent to generate test programs and test data for this class. If you do so, you need to indicated that in your README file.

For **Exercise 2 Part B,** you will be rewriting the program you wrote for Exercise 1 Part B to use your money class. There will be a few additional modifications to the output.

The name of the program should be Exercise2.cpp. The make file can be used to compile this program.

Program Input: The first argument on the command line (following the name of the executable) will be the name of the input file. Each line of the input file contains two monetary amounts: the amount of the purchase and the amount tendered. All monetary amounts will be in legal monetary input formats.

Program Output: The second argument on the command line (following the name of the executable) will be the name of the output file.

To turn in: Create a folder called *lastname*E2 to contain all of your source files from Part A and Part B. You should also include your input and expected output files from Part A, your makefile, README, and any additional test files you created for part B. Tar and zip your folder to a file called *lastname*E2.tgz and copy it to the course dropbox.

Date Due: Monday, 2 October 2023, 6:59 am.

Problem 2: Change Back

The Problem

Modern grocery stores now often have a ``U-Scan" checkout lane - allowing the customer to scan and check out their own groceries, without the need of a human checker. These lanes require that change be provided automatically, after the customer enters his/her cash. You are to write a program that computes the bills and coins to be dispensed, minimizing the total number of bills and coins. (That is, for change totaling $5.50, you should not dispense 5 ones and 50 pennies, but a $5 bill and a 50-cent piece.) The bills and coins available for you to dispense are as follows: $100 bill, $50 bill, $20 bill, $10 bill, $5 bill, $1 bill, 50-cent coin, 25-cent coin, 10-cent coin, 5-cent coin, 1-cent coin.

Input

The input file will consist of two numbers per line, each constituting a transaction. The first number is the amount of the purchase, and the second one is the amount tendered by the customer. The format of each value will be a legal money input format. Input is terminated by the end of the file. The name of the input file will be provided as a command line argument when the program is executed.

Output

Output will be written to a file. Output for each transaction will be a single line showing the amount of the purchase, the amount tendered, the amount of change returned and details of the number of bills and coins that will be dispensed as change, in ascending order of monetary amount. If a bill/coin is not needed in the change returned, no output is produced for that bill/coin. The output format is illustrated below. The width of the column labeled “Trans.” is 6; the width of the columns labeled “Purchase”, “Tendered”, and “Change” is 10; and the width of the remaining columns is 5. Transactions without sufficient tender should not be included in the totals. The name of the output file will be provided as a command line argument when the program is executed.

Sample Input

```
42.15 $50
$5.77 5.00
99.99 100.00
142.15 500
2.47 5.00
2.07 $5
99.41 $1000.41
999 1000.00
```

Corresponding Output

```
Trans.  Purchase  Tendered    Change   1c   5c  10c  25c  50c   $1   $5  $10  $20  $50 $100
     1    $42.15    $50.00     $7.85                1    1    1    2    1
     2     $5.77     $5.00    -$0.77   Insufficient amount tendered
     3    $99.99   $100.00     $0.01    1
     4   $142.15   $500.00   $357.85                1    1    1    2    1              1    3
     5     $2.47     $5.00     $2.53    3                        1    2
     6     $2.07     $5.00     $2.93    3    1    1    1    1    2
     7    $99.41  $1000.41   $901.00                                 1                        9
     8   $999.00  $1000.00     $1.00                                 1
Totals  $1387.24  $2660.41  $1273.17    7    1    3    3    4   10    2              1   12
```