CSC 101 Program 1 Test Plan

This is a test plan for programming assignment 1. It's in the format we'll use throughout most of the quarter. The plan consists of *test cases*, each one of which runs the make_change program with different inputs. For example, the first test case runs the program with the sample inputs shown in the Program 1 writeup. These inputs are a purchase amount of of 216 cents and an amount tendered of 500 cents.

For this assignment, the program being tested must be named "make_change". You make this program with the compiler command

```
gcc -ansi -pedantic -Wall -Werror make_change.c -o make_change
```

as described in the writeup.

There is an executable testing script named run.csh. This script implements the test plan defined in the table below. This is the script that will be run on your submitted program to compute your score.

It is strongly recommended that you run this script yourself, before submitting your program. To do so, you need a copy of the script itself, as well as a complete copy of both the inputs and expected-output testing directories. To obtain all of the testing files, run the following UNIX commands from the directory where you've stored your executable make_change program:

```
cp -rp ~gfisher/classes/101/programs/1/testing .
cd testing
./run.csh
```

Note that you only need to run the "cp -rp ..." command one time, not every time you want to test your program. A two-part name is used for the testing files, corresponding to the two program inputs for each test case. For example, the file input/216,500 has the inputs for the first test case. The same file name is used in all of the testing sub-directories:

- files in the input directory have the input values for the program
- files in the expected-output directory have the correct program results for those inputs
- files in the output directory have the actual program results produced by your program
- files in the diffs directory have any differences between the expected output and your actual output

Before your programs runs, both the output and diffs directories will be empty. After your program runs, the output directory will have one file for each test case run through your program.

If your make_change program passes all of the test cases, the only terminal output you will see is the score, which will be "100/100" points. If one or more cases fail, the script will report the failure(s), and print the appropriate score at the end. The difference files in the diffs directory show the details of how your output differs from the expected output. The testing script uses the UNIX diff utility to compare expected and actual output files. An explanation of how to read UNIX diff files is available at http://www.gnu.org/software/diffu-tils/manual/diffutils.html#Comparison

The complete test plan has 25 test cases. Each test case is worth 4 points, for a total of 100 points possible. The following table describes what each test case is for.

Test Case	Input: purchased,tendered	Correct Output	Description of Test Case
1	216,500	expected-output/ 216,500	Sample used in the assignment writeup.
2	0,0	expected-output/ 0,0	Empty case -- nothing purchased, no change.
3	0,100	expected-output/ 0,100	Free case -- $1 purchase, everything back in change.
4	100,100	expected-output/ 100,100	No change case -- $1 purchase, $1 tendered.
5	1,100	expected-output/ 1,100	Max change for $1 purchase.
6	99,100	expected-output/ 99,100	Min change for $1 purchase.
7	1,500	expected-output/ 1,500	Max change for $5 purchase.
8	499,500	expected-output/ 499,500	Min change for $5 purchase.
9	81,500	expected-output/ 81,500	Everything but quarters.
10	69,500	expected-output/ 69,500	Everything but dimes.
11	36,500	expected-output/ 36,500	Everything but nickels.
12	60,500	expected-output/ 60,500	Everything but pennies.
13	57,500	expected-output/ 57,500	Some of each.
14	9,10	expected-output/ 9,10	1 penny in change, 0 for all other denominations, typical case
15	95,100	expected-output/ 95,100	1 nickel in change, 0 for all other denominations, typical case
16	40,50	expected-output/ 40,50	1 dime in change, 0 for all other denominations, typical case
17	75,100	expected-output/ 75,100	1 quarter in change, 0 for all other denominations, typical case
18	400,500	expected-output/ 400,500	1 dollar in change, 0 for all other denominations, typical case
19	1,2	expected-output/ 1,2	1 penny in change, 0 for all other denominations, dumb customer
20	10,15	expected-output/ 10,15	1 nickel in change, 0 for all other denominations, dumb customer
21	10,20	expected-output/ 10,20	1 dime in change, 0 for all other denominations, dumb customer
22	50,75	expected-output/ 50,75	1 quarter in change, 0 for all other denominations, dumb customer
23	100,200	expected-output/ 100,200	1 dollar in change, 0 for all other denominations, dumb customer
24	1,10000000	expected-output/ 1,10000000	Max change from a $100,000 dollar bill
25	9999999,10000000	expected-output/ 9999999,10000000	Min change from a $100,000 dollar bill