CSC 101 Lab Week 2
Mileage Reimbursement Calculator

ISSUED:	Monday, 2 April 2012
DUE:	Monday, 9 April 2012, by the end of lab
POINTS POSSIBLE:	1
WEIGHT:	1% of total class grade

Overview

For this lab, you are writing a program that calculates mileage reimbursement at a specified rate. Your program should interact with the user in the following manner:

```
MILEAGE REIMBURSEMENT CALCULATOR

Enter beginning odometer reading: 13505.2
Enter ending odometer reading: 13810.6
Enter reimbursement rate: .35

You traveled 305.4 miles.  At $0.35 per mile,
    your reimbursement is $106.89.
```

Your program must produce output in precisely this format, including the blank lines, indentation, and number of decimal places in the numeric output. Specifically, the miles traveled is output with one decimal place and the reimbursement is output with two decimal places. Section 2.6 of the book discusses the details of formatting numbers in program output.

Exercise 1: Implementing the Program

Write the program described in the overview and put your source code in a file named `mileage.c`. Compile your program with the following command

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

Note the `gcc` argument "`-o mileage`". This tells the compiler to store the program in an executable file named "`mileage`", instead of "`a.out`". This means that you run the program using the name "`mileage`", instead of "`a.out`". Try it out.

Exercise 2: Using a Simple Makefile

You can save yourself tedious typing at the terminal using UNIX *Makefiles*. A Makefile is a set of commands that you execute simply by typing "`make`".

For this lab, a sample makefile is provided

```
http://users.csc.calpoly.edu/˜gfisher/classes/101/labs/2/Makefile
```

Copy this file into the directory where you have `mileage.c` and run the UNIX command:

```
make
```

This will compile your program in the same way as the `gcc` command did in Exercise 1.

Exercise 3: Using Input/Output Redirection

In this lab, you should also test your program with *input and output redirection*. Start by creating an input file named `sample_input`. This file should contain the following three lines:

```
13505.2
13810.6
.35
```

Input redirection allows you to send input from a file instead of typing it on the terminal. You can test your program with the following command:

```
mileage < sample_input
```

Notice that when using input redirection, the input from the file is not shown to the screen.

To save the output of your program to a file, use output redirection:

```
mileage < sample_input > sample_output
```

This command reads input from the file `sample_input` and writes the output to the file `sample_output`. You can look at the contents of `sample_output` with the following command:

```
more sample_output
```

Exercise 4: Using Sample Input/Output Files

The Lab 2 web page has two pairs of sample input/output files:

```
test_input1, test_output1
```

and

```
test_input2, test_output2
```

Run your `mileage` program using the input files, and your output should be the same as the corresponding output files.

Exercise 5: Using UNIX diff

UNIX has another handy utility named `diff`. It compares two files and reports the differences, if any. For example, the following commands will put the output from your `mileage` program in the file named `my_output1`, and compare that file with the sample `test_output1`:

```
mileage < test_input1 > my_output1
diff my_output1 test_output1
```

If the 2 files are exactly the same, diff will print out nothing. If the files are different, diff will print out the differences. See the UNIX man page for the `diff` command for details on its inputs, and the brief tutorial in http://www.unixtutorial.org/2008/02/compare-text-files-using-diff for an explanation of its output.

If you're feeling adventurous, you can try the Emacs `ediff` command, which is located on the `Tools` menu.

Submitting Your Program

After your program has been checked in person during lab, submit it on unix1 using the command:

```
handin gfisher 101_lab2 mileage.c
```