EECS 10: Assignment 6

October 27, 2006

Due on Monday 11/13/2006 12:00 noon. Note: this is a two-week assignment.

1 Calculations with rational numbers [80 points]

The goal of this assignment is to learn to use functions in C programing by doing algebraic operations of rational numbers. Here are the rules for rational number operations:

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$
$$\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$$
$$\frac{a}{b} \times \frac{c}{d} = \frac{ac}{bd}$$
$$\frac{a}{b} \div \frac{c}{d} = \frac{ad}{bc}$$

Your program should be a menu driven program. At the start of your program, the user is prompted to input a rational number by providing its numerator and denomiator. For instance, it may look like:

```
Welcome to my rational number calculator!
Please input a rational number.
Numerator: 5
Denominator: 7
```

Then the program should print out the current result and a menu. In our example, it looks like this:

```
--------------------------------------------
The current result is: 5 / 7
1. Add the current result with another rational number;
2. Subtract the current result from another rational number;
3. Multiply the current result with another rational number;
4. Divide the current result by another rational number;
5. Quit

Please enter a selection: 1
```

Printing Out the Current Result Rational Number

The current result indicates the result of the previous rational number operation. At the beginning, there is no previous operation, so the current result just shows the rational number you input at the start. In our example, it is 5/7.

Your program should always first reduce the current result in its reduced format, then do the print out. For example, 12/28 must be changed to 3/7 before printing. Also, make sure that in reduced form, the denominator of the rational number must always be positive, while the numerator can be either positive or negative. That is, $-3/-7$ should be changed to $3/7$, and $3/-7$ should be changed to $-3/7$.

When you print out your rational number, just print of something like 3/7 is sufficient. However, there are two special cases:

1. a rational number like 0/7 should be printed out as 0
2. a rational number like 3/1 should be printed out as 3

The Option Menu

Your program should let the user select an option, 1 through 5. If the user selects 5, your program simply exits. For other selections, your program should perform the corresponding rational number operations.

When the user selects an operation, first ask the user input another rational number. Once the user input the rational number (numerator followed by denominator), your program should display the current result and the option menu again. In our example, the user selects 1: the add operation. So the program should ask the user to input a rational number to be added to 5/7 like this:

```
Please input a rational number.
Numerator: 2
Denominator: 7
```

Then the program should bring up the current result and menu again. This time the current result should display 1 ($5/7 + 2/7 = 1$).

```
--------------------------------------------
The current result is: 1
1. Add the current result with another rational number;
2. Subtract the current result from another rational number;
3. Multiply the current result with another rational number;
4. Divide the current result by another rational number;
5. Quit

Please enter a selection:
```

Writing Functions

To implement the program, you need to define some functions; each function does a specific job, such as addition, subtraction, and division etc. Here are the functions you need to define:

```
void rAdd(int, int, int, int);      /*do rational number addition*/
void rSubtract(int, int, int, int); /*do rational number subtraction*/
void rMultiply(int, int, int, int); /*do rational number multiplication*/
void rDivide(int, int, int, int);   /*do rational number division*/
void reduceResultRational();        /*change the global result rational
                                      number to its reduced format*/
void printRational(int, int);       /*print a rational number*/
```

You need to implement these functions outside the main function, and call them in the main function. To define a function, you need to give the name of the function, the input parameters for the funciton, and the return type of the function. For example, the function declaration *void rAdd(int, int, int, int)* means that the function's name is *rAdd*; it requires 4 input parameters, which are of type integer, and the return type of the function is *void*. Functions *rAdd*, *rSubtract*, *rMultiply*, and *rDivide* require four integers as their input parameters. Those four integers are the numerator, the denominator, the numerator, and the denominator of the first and second rational numbers respectively; function *reduceMyRational* requires no input parameters; function *printRational* require two integers as their input parameters, which are the numerator and the denominator of a rational number.

You may want to define some other functions as well.

Using Global Variables

You also need to declare at least two global variables: they are for the *current result*. Those variables must be declared outside the main function. See the following example for the declaration of global variables and the calling of user defined functions in the main function.

Note that this is just an example, NOT a part of your rational number program!

```
#include <stdio.h>

int g; /*global variable*/

void myMultiply( int n1, int n2 ) /*user defined function*/
{
    g = n1 * n2;
}

void myAddition( int n1, int n2 ) /*user defined function*/
{
    g = n1 + n2;
}

int main()
{
    int a, b;
    a = 5;
    b = 10;

    myMultiply(a, b); /*call the user defined function*/
    printf("The multiplication of %d and %d is: %d \n", a, b, g);

    myAddition(a, b); /*call the user defined function*/
    printf("The addition of %d and %d is: %d \n", a, b, g);

    return 0;
}
```

You need to save your program as *rational.c*. To demonstrate that your program works correctly, perform the following steps to generate a script file:

1. Compile and run your program
2. Input the rational number 5/7
3. Add the current result with 2/7
4. Subtract 1/5 from the current result
5. Multiply the current result by -9/14
6. Divide the current result by 8/9

7. Add the current result with 81/140
8. Add the current result with 9/0 /*error handling, only if you did extra credit*/
9. Divide the current result by 0/7 /*error handling, only if you did extra credit*/
10. Exit your program

For your reference, the result obtained after step 7 should be 0.

Name the script file to *rational.script*. If you forgot how to create a script file, check the instructions for homework 1.

You need to create a *rational.txt* file, in which briefly explain your program (if you did the extra credit, you also need to include comments on your error handlings).

2 Extra credit: error handlings [10 points]

A robust program is able to handle all situations, and reports to the user if an error occurs. To get the extra credit, you program should be able to handle the following errors:

- Notify the user if the denominator of a rational number is zero
- Notify the user if the denominator in the division operation is zero

In case of input error, the program should keep on asking the user until a proper number is entered.

Your program should show an error message if an error occurs; for example, if the denominator in a division operation is zero, print a message such as:

```
Error! divides a number by zero!
```

3 Submission

The submission is similar to the previous homeworks. You need to create a directory called *hw6*; put all the files *rational.c*, *rational.script*, and *rational.txt* in the directory and run the command */ecelib/bin/turnin* to submit your homework.