CSE102 – Homework 5

Submission Deadline: November 20, 2016,11:30pm

1. (70 Pts)

There are several ways of representing real numbers in C. Although built in types such as "double" are useful for representing a real number, they may not be exact, especially for rational numbers. In this homework, you are asked to develop a simple exact arithmetic system that works on fractional numbers without any loss of accuracy.

A rational number is represented by a fraction $\frac{a}{b}$ where a and b are integers. For example, 1/3, 2/1 and -3/5 are all fractions.

Assume that we will use strings to represent a fraction. Examples include:

- "1/3" represents +1/3
- "2" represents +2/1
- "0" represnets 0
- "-12/97" represents -12/97
- "123456789/1234567890" represents +123456789/1234567890

Implement a library (a set of functions) providing the following operations on fractions represented as strings:

- **sfrac_simplify(char * n):** simplifies the fractional number such that there is no common divisor of the two parts of the fraction.
- **char * sfrac_add(char * n1, char * n2):** add two numbers n1 and n2 and return the resulting fractional number (make sure that the number is simplified).
- **char * sfrac_sub(char * n1, char * n2):** subtract n2 from n1 and return the resulting fractional number (make sure that the number is simplified).
- **char * sfrac_negate(char * n):** negates the given number.
- **char * sfrac_mult(char * n1, char * n2):** multiplies two numbers n1 and n2 and return the resulting fraction (with simplification if needed).
- **char * sfrac_div(char * n1, char * n2):** divides the number n1 by n2 and return the resulting fraction (with simplification if needed).
- **char * sfrac_fromdouble(double x):** constructs a fractional number from the given double number.
- **double sfrac_todouble(char * x):** converts a fractional number to a double number. If there is overflow, return "NaN".
- **void sfrac_print(char *a1, char *n1, char *a2, char *n2, char *a3, char *n3, char *a4):** print the three numbers (n1,n2 and n3) with the strings a1, a2, a3 and a4 in between and around them.

You should make sure that the input arguments (the strings) are not corrupted during the call to these functions when one or more of them are used as outputs. For example in the the following code segment

```
char n1[100] = "1/3";
char n2[100] = "2/5";
```

```
n1 = sfrac_add(n1,n2);
```

n1 should have the correct answer “11/15”.

Your print function should work like the following:

- sfrac_print(“Result = “, “7/33”, “ + “, “1/133”, “ ?=? “, “2/3”, “.”);
  ```
            7      1      2
  Result = -- + --- ?=? -.
           33   133      3
  ```
- sfrac_print(“x = “, “7/33”, 0,0,0,0, 0);
  ```
   7
  --
  33
  ```

Place your code in a file named “sfrac.c” and the function declaretions in “sfrac.h”.

Write a driver program “sfrac_test.c” such that it reads a fractional operation from the user performs the operation. For example (the red colored text indicate the user input):

-
  ```
  $ ./sfrac_test
  Input your operation: 1/7+2/7
  Result is: 3/7
  ```

-
  ```
  $ ./sfrac_test
  Input your operation: 1/2*2/3
  Result is: 2/6
  ```

-
  ```
  $ ./sfrac_test
  Input your operation: -1/7+-2/7
  Result is: -3/7
  ```

-
  ```
  $ ./sfrac_test
  Input your operation: 1+2/7
  Result is: 9/7
  ```

-
  ```
  $ ./sfrac_test
  Input your operation: 1/2:2/3
  Result is: 3/4
  ```

Assume that addition, subtraction, multiplication and division (“:”) operations are allowed from the user.

2. (30 Pts)

Using the library provided in the first part, implement a program that finds the fist 1000 prime numbers. Handin your code in file “prime_with_fractions.c”.

Rules:

1. Obey honor code principles.
2. **Read your homework carefully** and follow the directives about the I/O format (data file names, file formats, etc.) and submission format **strictly**. Violating any of these directives will be penalized.
3. Obey coding convention.
4. Do not forget to put the required **tags** in the main function.
5. Your submission should include the following file **and NOTHING MORE** (no data files, object files, etc):

   HW05_<StudentName>_<Sirname>_<student number>_ sfrac.c
   HW05_<StudentName>_<Sirname>_<student number>_ sfrac.h
   HW05_<StudentName>_<Sirname>_<student number>_ sfrac_test.c
   HW05_<StudentName>_<Sirname>_<student number>_ prime_with_fractions.c

   Do **NOT** compress the files you submit.
6. Do not use non-English characters in any part of your homework (in body, **file name**, etc.).
7. Deliver the printout of your work **until the last submission date**.