EE231002 Introduction to Programming

Lab05. Continued Fractions

Due: Mar. 25, 2014

In mathematics, any real number, x, can be represented by a continued fraction as the following

$$x = a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \cfrac{b_4}{\cdots}}}} \tag{5.1}$$

where a_i, b_i are integers. As a shorthand representation, we write the above equation as a sequence

$$x = [\ a_0;\ b_1,\ a_1,\ b_2,\ a_2,\ b_3,\ a_3,\ \ldots\ ]. \tag{5.2}$$

The sequence can be finite or infinite. If the sequence is infinite, let

$$x_k = [\ a_0;\ \underbrace{b_1,\ a_1,\ \ldots,\ b_k,\ a_k}_{2k \text{ integers}}\ ], \tag{5.3}$$

Thus,

$$x_0 = a_0, \tag{5.4}$$

$$x_1 = a_0 + \frac{b_1}{a_1}, \tag{5.5}$$

$$x_2 = a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2}}, \tag{5.6}$$

$$x_3 = a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3}}}, \tag{5.7}$$

$$x_4 = a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \cfrac{b_4}{a_4}}}}. \tag{5.8}$$

And,

$$\lim_{k \to \infty} x_k = x. \tag{5.9}$$

Using continued fractions, the square root of any integer, x, can be expressed as

$$\sqrt{x} = 1 + \cfrac{x-1}{2 + \cfrac{x-1}{2 + \cfrac{x-1}{2 + \cdots}}}. \tag{5.10}$$

Or, in sequence notation

$$\sqrt{x} = [\ 1;\ x-1,\ 2,\ x-1,\ 2,\ x-1,\ 2,\ \dots\ ] \tag{5.11}$$

Since this is an infinite sequence, the value can only be approximated by a finite sequence in computer evaluation.

$$\sqrt{x} \approx \text{II}_k(x) = [\ 1;\ \underbrace{x-1,\ 2,\ x-1,\ 2,\ x-1,\ 2,\ \dots,\ x-1,\ 2,}_{2k \text{ integers}}] \tag{5.12}$$

In this case, the error exists and is

$$\varepsilon_k = \text{II}_k(x) - \sqrt{x}. \tag{5.13}$$

For example, $x = 2$, $\sqrt{2}$ can be approximated by the following sequence

$$\begin{aligned}
\text{II}_0(2) &= 1 \\
\varepsilon_0 &= 1 - \sqrt{2} = -0.414214 \\
\text{II}_1(2) &= 1 + \frac{1}{2} = 1.5 \\
\varepsilon_1 &= 1.5 - \sqrt{2} = 0.0857864 \\
\text{II}_2(2) &= 1 + \cfrac{1}{2 + \cfrac{1}{2}} = 1.4 \\
\varepsilon_2 &= 1.4 - \sqrt{2} = -0.0142136 \\
\text{II}_3(2) &= 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2}}} = 1.41667 \\
\varepsilon_3 &= 1.41667 - \sqrt{2} = 0.0024531
\end{aligned}$$

The following table shows the approximations up to $k = 10$.

Table 1. Continued fraction approximation for $\sqrt{2}$.

k	II_k	ε_k
0	1	-0.414214
1	1.5	0.0857864
2	1.4	-0.0142136
3	1.41667	0.0024531
4	1.41379	-0.000420459
5	1.41429	7.21519e-05
6	1.4142	-1.23789e-05
7	1.41422	2.1239e-06
8	1.41421	-3.64404e-07
9	1.41421	6.25218e-08
10	1.41421	-1.0727e-08

In this assignment, you need to write a `C` program to find the approximate values of $\sqrt{2}$, $\sqrt{11}$, $\sqrt{121}$, $\sqrt{1221}$, $\sqrt{12321}$, $\sqrt{123321}$, $\sqrt{1234321}$, and $\sqrt{12344321}$, using continued fraction (Equation 5.10) with small errors. To ensure a small error is obtained, your program should check for

$$|\ \text{II}_k\ (x) - \text{II}_{k-1}(x)| \le 10^{-9}. \tag{5.14}$$

assuming $\text{II}_k(x)$ is the approximate value of $\sqrt{x}$.

Example output of the program is

```
$ ./a.out
       x    sqrt(x)
       2    1.414213562
      11    xxxxxxxxxxx
     121   11.000000000
    1221   xxxxxxxxxxxx
   12321  111.000000000
  123321  xxxxxxxxxxxxx
 1234321 1111.000000000
12344321 xxxxxxxxxxxxxx
```

Notes.

1. Create a directory **`lab05`** and use it as the working directory.
2. Name your program source file as **`lab05.c`**.
3. The first few lines of your program should be comments as the following.

   ```
   /* EE231002 Lab05. Continued Fractions
      ID, Name
      Date:
   */
   ```

4. After finishing editing your source file, you can execute the following command to compile it,

   ```
   $ gcc lab05.c
   ```

   If no compilation errors, the executable file, **`a.out`**, should be generated, and you can execute it by typing

   ```
   $ ./a.out
   ```

5. After you finish verifying your program, you can submit your source code by

   ```
   $ ~ee231002/bin/submit lab05 lab05.c
   ```

   If you see a "submitted successfully" message, then you are done. In case you want to check which file and at what time you submitted your labs, you can type in the following command:

   ```
   $ ~ee231002/bin/subrec
   ```

   It will show the last few submission records.

6. You should try to write the program as efficient as possible. The format of your program should be compact and easy to understand. These are part of the grading criteria.