CS19003 : Programming and Data Structures Laboratory

Assignment 3 : Loop Statements in C

Date: 20-April-2021

Problem Statement:

After studying the ongoing murder mystery case for some period, Detective Byomkesh Bakshi and Ajit now have several clues in hand. However, so many of these clues are gradually getting entangled harder making the wayout more difficult. Ajit pointed Mr. Bakshi to formulate the entangled path of clues into a mathematical 'knot theory' problem so that they can apply formal/mathematical techniques to untie them easily. Continued Fractions are an well-studied topic in Algebra (Mathematics) and widely used to solve many types of such knot theory problems and often help to formulate the unfolding of deep rooted knots. So, they might need your help to proceed with the implementation of this concept in order to apply it in the case.

Formally, the finite continued fractions over the rational numbers (such as $\frac{x}{y}$, where $x, y > 0$) are expressed as:

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots \; \frac{1}{a_n}}}}} \quad \equiv \quad [\ a_0\ a_1\ a_2\ a_3\ \ldots\ a_n\ ] \quad \leftarrow (\texttt{notation})$$

Any rational number can be expressed in a finite continued fraction form following the basic division (with quotient and remainder) principle, as shown below:

[Note: $\frac{x}{y}, \frac{y}{r_0}, \frac{r_0}{r_1}, \cdots$ produces the quotients, $a_0, a_1, a_2, \ldots$, and the remainders, $r_0, r_1, r_2, \ldots$, respectively.]

$$\frac{x}{y} = a_0 + \frac{r_0}{y} = a_0 + \frac{1}{\frac{y}{r_0}} = a_0 + \frac{1}{a_1 + \frac{r_1}{r_0}} = a_0 + \frac{1}{a_1 + \frac{1}{\frac{r_0}{r_1}}} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{r_2}{r_1}}} = \cdots = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots \; \frac{1}{a_n}}}}$$

However, the generalized (and infinite) form for the continued fractions are expressed as: $a_0 + \cfrac{a_1}{a_2 + \cfrac{a_3}{a_4 + \cfrac{a_5}{\ddots}}}$

There are interesting algebraic manipulations possible with this. In particular, you can determine square root of any positive number using this concept. For example, to determine $x = \sqrt{3}$, we solve the equation $x^2 = 3$ as:

$$x^2 - 1 = 2 \quad \Longrightarrow (x+1)(x-1) = 2 \quad \Longrightarrow x - 1 = \frac{2}{1+x} \quad \Longrightarrow x = 1 + \frac{2}{1+x}$$

Now, the x in the RHS denominator can be further decomposed iteratively as continued fractions with the existing x from LHS itself – thereby stepwise expanding to,

$$x = \sqrt{3} = 1 + \frac{2}{1+x} = 1 + \frac{2}{2 + \frac{2}{1+x}} = \cdots = 1 + \cfrac{2}{2 + \cfrac{2}{2 + \cfrac{2}{\ddots}}}. \qquad \textit{Similarly,} \quad \sqrt{9} = 1 + \cfrac{8}{2 + \cfrac{8}{2 + \cfrac{8}{\ddots}}}$$

So, in general, the real quadratic roots can also be realized using such methods (as you might already sensed):

$$x^2 + bx + c = 0 \implies x = -b - \frac{c}{x} = -b - \frac{c}{-b - \frac{c}{x}} = \cdots = -b - \cfrac{c}{-b - \cfrac{c}{-b - \cfrac{c}{\ddots}}}$$

However, this value of x will give you one root; the other root can then be found out by known formula, $\frac{c}{x}$. Please note that, you need to provide $b^2 \geq 4c$ so that the continued iterations converge into a root as result (that is, only non-complex/real roots of the quadratic equation can be determined!). *Check what happens to your implementation when you give $b^2 < 4c$ – why do you get infinite loop/iterations? Think!*

Can you write a C-program to automate this process of representing rational and real numbers as well as calculating square root of numbers and roots of quadratic equations using iterations/loops? Note that, *you are* **not allowed** *to use arrays or functions.* In particular, your program will do the following:

- Take from user a choice from the menu containing five options: (1) represent rational fraction, (2) represent reals, (3) calculate square root, (4) calculate roots of quadratic equation and (0) exit from the program.

- When user selects choice as 1, the following is to be carried out:
  - Take from user two positive integers, x and y.
  - Normalize the fraction $\frac{x}{y}$, by dividing numerator (x) and denominator (y) both by their `GCD` value.
  - Represent $\frac{x}{y}$ using finite continued fraction representation.
  - Print the continued fraction representation in proper format, that is $[\ a_0\ a_1\ \ldots\ a_n\ ]$.
  - After finishing, automatically loop back to show the menu again.
- When user selects choice as 2, the following is to be carried out:
  - Take from user a positive real number, `real`, having a maximum of 6 digits after the decimal point.
  - Represent the finite `real` using finite continued fraction representation.
  - Print the continued fraction representation in proper format, that is $[\ a_0\ a_1\ \ldots\ a_n\ ]$.
  - After finishing, automatically loop back to show the menu again.
- When user selects choice as 3, the following is to be carried out:
  - Take from user a positive (*why?*) integer whose square root is to be computed.
  - Calculate the square-root of the given number by iteratively unfolding the continued fractions until the difference of previous iteration and current iteration results falls below 0.000001 *(convergence)*.
  - Print the root values in each iteration that are calculated gradually.
  - Finally, print the last-step root calculated before stopping (due to convergence) and also report the root value calculated by direct function `sqrt` from `math.h` library. *Are they almost same?*
  - After finishing, automatically loop back to show the menu again.
- When user selects choice as 4, the following is to be carried out:
  - Take from user two real values (b, c with $b^2 \geq 4c$) for the quadratic equation of the form $x^2+bx+c=0$.
  - Print the quadratic equation in $x^2+bx+c=0$ format.
  - Calculate one of the roots (say, r_1) by iteratively unfolding the continued fractions (as illustrated above) until the difference of previous iteration and current iteration results falls below 0.000001.
  - Print the root (r_1) values in each iteration that are calculated gradually.
  - Finally, print the last-step r_1 calculated before stopping (due to convergence) and also print the other root from $r_2 = \frac{c}{r_1}$ formula.
  - Also report the two root values calculated by direct formula of computing quadratic roots, that is $r_1 = \frac{-b+\sqrt{b^2-4c}}{2}$ and $r_2 = \frac{-b-\sqrt{b^2-4c}}{2}$. *Are they almost same?*
  - After finishing, automatically loop back to show the menu again.
- When user selects choice as 0, exit from the program.
- If user enters a choice other than $(1, 2, 3, 4, 0)$, prompt a wrong choice message and go back to the menu.

Example Execution Details:

```
====================================================
++ Determine the Following using Continued Fraction:
  --> 1. Representation of Rational Fractions
  --> 2. Representation of Finite Reals
  --> 3. Square Root of Positive Integer
  --> 4. Roots of Quadratic Equation
  --> 0. Exit
====================================================
++ Enter Your Choice: 1

++ Enter Two Positive Integers (space-separated): 2700 1884

** The Continued Fraction Representation of 2700/1884 = 225/157 : [ 1 2 3 4 5 ]

====================================================
++ Determine the Following using Continued Fraction:
  --> 1. Representation of Rational Fractions
  --> 2. Representation of Finite Reals
  --> 3. Square Root of Positive Integer
  --> 4. Roots of Quadratic Equation
  --> 0. Exit
====================================================
++ Enter Your Choice: 2

++ Enter a Positive Real Number: 3.141593

** The Continued Fraction Representation of 3.141593: [ 3 7 16 983 4 2 ]

====================================================
++ Determine the Following using Continued Fraction:
  --> 1. Representation of Rational Fractions
  --> 2. Representation of Finite Reals
  --> 3. Square Root of Positive Integer
  --> 4. Roots of Quadratic Equation
  --> 0. Exit
====================================================
++ Enter Your Choice: 3
```

```
++ Enter a Positive Integer: 3

** Iteration-wise Square-Root Values:
Iteration ##      1 : SQRT(3) = 1.000000
Iteration ##      2 : SQRT(3) = 2.000000
Iteration ##      3 : SQRT(3) = 1.666667
Iteration ##      4 : SQRT(3) = 1.750000
Iteration ##      5 : SQRT(3) = 1.727273
Iteration ##      6 : SQRT(3) = 1.733333
Iteration ##      7 : SQRT(3) = 1.731707
Iteration ##      8 : SQRT(3) = 1.732143
Iteration ##      9 : SQRT(3) = 1.732026
Iteration ##     10 : SQRT(3) = 1.732057
Iteration ##     11 : SQRT(3) = 1.732049
Iteration ##     12 : SQRT(3) = 1.732051

** The Final Square-Root = 1.732051
** The MATH FORMULA Square-Root = 1.732051

====================================================
++ Determine the Following using Continued Fraction:
  --> 1. Representation of Rational Fractions
  --> 2. Representation of Finite Reals
  --> 3. Square Root of Positive Integer
  --> 4. Roots of Quadratic Equation
  --> 0. Exit
====================================================
++ Enter Your Choice: 4

++ For Quadratic Equation x^2 + Bx + C = 0,
   Enter B and C (space-separated real-values ensuring B^2 >= 4C) : -3 2

** The Quadratic Equation: x^2 + (-3.000000)x + (2.000000) = 0
** Iteration-wise Root Values:
Iteration ##      1 : Root = -3.000000
Iteration ##      2 : Root = 3.666667
Iteration ##      3 : Root = 2.454545
Iteration ##      4 : Root = 2.185185
Iteration ##      5 : Root = 2.084746
Iteration ##      6 : Root = 2.040650
Iteration ##      7 : Root = 2.019920
Iteration ##      8 : Root = 2.009862
Iteration ##      9 : Root = 2.004907
Iteration ##     10 : Root = 2.002447
Iteration ##     11 : Root = 2.001222
Iteration ##     12 : Root = 2.000611
Iteration ##     13 : Root = 2.000305
Iteration ##     14 : Root = 2.000153
Iteration ##     15 : Root = 2.000076
Iteration ##     16 : Root = 2.000038
Iteration ##     17 : Root = 2.000019
Iteration ##     18 : Root = 2.000010
Iteration ##     19 : Root = 2.000005
Iteration ##     20 : Root = 2.000002
Iteration ##     21 : Root = 2.000001

** The Final Root = (2.000001)
   The Other Root = (1.000000)
** The MATH FORMULA Roots = (2.000000) and (1.000000)

====================================================
++ Determine the Following using Continued Fraction:
  --> 1. Representation of Rational Fractions
  --> 2. Representation of Finite Reals
  --> 3. Square Root of Positive Integer
  --> 4. Roots of Quadratic Equation
  --> 0. Exit
====================================================
++ Enter Your Choice: 5

** Wrong Choice Entered! Please Try Again ...

====================================================
++ Determine the Following using Continued Fraction:
  --> 1. Representation of Rational Fractions
  --> 2. Representation of Finite Reals
  --> 3. Square Root of Positive Integer
  --> 4. Roots of Quadratic Equation
  --> 0. Exit
====================================================
++ Enter Your Choice: 0

** Thank You! Bye ...
```

Submit a single C source file. Do not use *arrays*, *functions* or global/static variables.