TECNOLÓGICO SUPERIOR DE JALISCO
MARIO MOLINA
ZAPOPAN

Data structures

Recursiveness – continued fractions

Deadline: 09/04/2017.

Objective

To implement a recursive function to compute a continued fraction.

Background

A *continued fraction* is an expression of the form:

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

A continued fraction

If $b_i = 1$ for all i, the expression is called a *simple continued fraction*, or a *generalized continued fraction*, otherwhise. The integers a_i are called the coefficients of terms of the continued fraction.

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

A simple continued fraction

A simple continued fraction can be written $[a_0; a_1, a_2, a_3, \cdots]$. A *finite continued fraction* (or *terminated continued fraction*) is that continued fraction in which the iteration/recursion is terminated after finitely many steps by using an integer a_n in lieu of another continued fraction, i.e. $[a_0; a_1, a_2, \cdots, a_n]$. If $x_n = [a_0; a_1, a_2, \cdots, a_n]$ then $x \approx x_n$, for a value of n large enough.

The Euler's number

The Euler's number, also denoted e is a mathematical constant that is the base of the natural logarithm: the unique number whose natural logarithm is equal to one. It is an *irrational* (is not the ratio of integers) and *trascendental* (it is not a root of any non-zero polynomial with rational coefficients) number. Its value is approximately equal to 2.71828182846:

$$e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n$$

The Euler's number

The Euler's number can be characterized in many different ways. One of them is by a simple continous fraction $[2; \overline{1, 2k, 1}]$ with a growing $k \geq 1$:

$$e = 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{4 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{6 + \cfrac{1}{1 + \cdots}}}}}}}}} \quad (1)$$

The Euler's number as a continued fraction

Development of the practice and example

The student must implement the terminated form of the continued fraction (equation 1) to aproximate the Euler's number e using a recursive function with the following prototype:

```
double euler(int n);
```

where $a_n = 1$. So, the call `euler(n)` calculates e_n which aproximates e. For example, `euler(5)` calculates e_5:

$$
\begin{aligned}
e_5 &= 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{1 + \frac{1}{1}}}}} \\
&= 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{1 + \frac{1}{2}}}} \\
&= 2 + \cfrac{1}{1 + \cfrac{1}{2 + \frac{1}{1.5}}} \\
&= 2 + \cfrac{1}{1 + \frac{1}{2.6667}} \\
&= 2 + \frac{1}{0.3749} \\
&= 2.6674 \approx e
\end{aligned}
$$

Delivering

You must compress your code in a file named `ef.`*<name>*`.zip` (`.rar`, `.7z`), where *<name>* is your name, e.g. `ef.alejandro.aguilar.zip`. Don't add object code or executable code. Send your file to aaguilar.itszapopan@gmail.com with the subject *Euler's fraction*. Your deadline will be Monday, September 4th.