CS 238	**Discrete Mathematics II**
Clearness	
Recurrence	*Exploration*

Goal

The goal of this exploration is for you to explore some of the intricacies of a special type of a clearly relevant recurrence relation. This special type is a "practical" recurrence relation rooted in the world of finance.

Background

In many financial scenarios, you want to find certain numbers given certain others. For example:

- Given:
  - *Principal* or *Present* Value (p)
  - APR – Annual Percentage *Rate* (r)
  - Monthly *Interest* Rate (i) ($= r/12$ as a percent)
  - *Monthly* Payment (m)
- Find: the *Number* of payments (n).

If after n payments there is a *Balance* (or *Future Value*) b remaining, the formula relating the above is

$$p = \frac{m}{i}[1 - (1+i)^{-n}] + b(1+i)^{-n} \tag{1}$$

which if b is zero gives

$$p = \frac{m}{i}[1 - (1+i)^{-n}]$$

which implies

$$\frac{p - \frac{m}{i}}{-\frac{m}{i}} = (1+i)^{-n}$$

which when solved for n gives

$$n = \frac{-log\left(\frac{p-\frac{m}{i}}{-\frac{m}{i}}\right)}{log(1+i)}.$$

For example, if $p = 60000$, $m = 860$ and $i = 0.01$ ($r = 12$) then

$$n = \frac{-log\left(\frac{60000-86000}{-86000}\right)}{log(1.01)} = \frac{-log\left(\frac{26}{86}\right)}{log(1.01)} = 120.$$

Solving for the monthly payment m in terms of the other numbers gives

$$m = p(i/(1-(1+i)^{-n})).$$

If $a =$ the number of payments made, then the balance remaining to be paid is

$$b = m((1-(1+i)^{(a-n)})/i).$$

Question: How was formula (1) derived?

Here's how:

- Initial condition: $p_0 = p$, the original *Present* Value or *Principal*
- Recurrence relation:
$$p_n = p_{n-1}(1+i) - m$$
- Which means (via back substitution):

$$\begin{array}{rcl} p_1 & = & p_0(1+i) - m \\ p_2 & = & p_1(1+i) - m = p_0(1+i)^2 - m[(1+i)+1] \\ p_3 & = & p_2(1+i) - m = p_0(1+i)^3 - m[(1+i)^2 + (1+i) + 1] \\ & \vdots & \\ p_n & = & p_0(1+i)^n - m[\sum_{j=0}^{n-1}(1+i)^j] \end{array}$$

You know that

$$\sum_{j=0}^{n-1}(1+i)^j = \frac{(1+i)^n - 1}{i}.$$

because it's just the sum of a geometric series; therefore

$$p_n = p_0(1+i)^n - m\left[\frac{(1+i)^n - 1}{i}\right].$$

which is *almost* formula (1). So if $p_n = 0$ (meaning a fully **amortized** annuity or loan) then

$$p(1+i)^n = \frac{m}{i}[(1+i)^n - 1]$$

or

$$p = \frac{m}{i}[1-(1+i)^{-n}]$$

which is formula (1) with b equal to zero.

Requirements

Use the supplied sample code as a guide and a starting point. The file `clearnessCLI.cpp` is a C++ program with a Command-Line Interface that takes inputs which are files containing parameter settings, or else these same settings as command-line parameters. Your piece of the program (in a file named `clearness.cpp`) will simply compute p given m, n, and r, or else m given p, n and r, or else n given p, m and r, or else (the hardest one), r given p, m and n.

The `clearness` program supports the additional optional parameters x, s and e, for the extra monthly payment made (x) between the start (s) and end (e) months (inclusive). If supplied, there will be only one such contiguous interval where extra payments are made, where e is also optional (defaulting to the last numbered month).

Make your code calculate using double precision, truncating money amounts to the penny, which is how financial institutions do it. Do all your work in the `clearness.cpp` file, which has an already mostly implemented `Amortize` class definition (you're welcome), and pay close attention to the instructions in the comments. Nothing but your `clearness.cpp` file will be submitted. Nothing you change in (your own copy of) `clearnessCLI.cpp` will be accepted, so don't even think about copying and modifying that file.

If conditions are right, you can build, test and submit your code in the Linux Lab via the command:

```
make it just so
```

Grading Criteria

See the self-assessment!