CSC205 - Program #2
Due Thursday, September 10
Population Growth with the Logistic Model

Application:

Write a program to predict population growth. For example, let's assume we have a bunny farm and we want to calculate how fast our herd of bunnies is growing. First, ask the user for the number of bunnies that the farm can comfortably support. (This is the 'carrying capacity' of the farm; above that number, bunnies start dying from overcrowding.)

Then your program will be able answer both of the following kinds of questions:

(1) How many bunnies will I have after this many months?
(2) How many months will I have to wait to get this many bunnies?

Let the user choose which one to answer. Then ask the user for the information necessary to answer the question. After answering each question, ask the user whether they want to ask another question for that farm (with the same carrying capacity). If they want to ask another question, then ask for all the information again (except for the carrying capacity). When they are done asking questions about that farm, ask the user if they want to try another farm. If they do, then ask them for the carrying capacity and repeat the whole process for any number of farms. Otherwise, stop the program.

Calculations:

Use the following formula, called the Logistic Model, to calculate how many bunnies you have at the end of each month:

$$B_{new} = B_{old} + g\, B_{old} (1 - B_{old})$$

where:

P_{old} = the actual number of bunnies at the beginning of the month
B_{old} = the relative population of bunnies at the beginning of the month
where $B_{old} = P_{old} / N$
B_{new} = the relative population of bunnies at the end of the month
P_{new} = the actual number of bunnies at the end of the month
[notice that $P_{new} = B_{new} \bullet N$]
N = the carrying capacity; the number of bunnies that the local habitat will support
g = the growth rate (bunnies per bunny per month)

Notice that the 'P'-numbers are actual populations, so they will be integers (no partial bunnies). In contrast, the 'B'-numbers are ratios of the actual population over the carrying capacity, so they must be real numbers. The 'B'-numbers represent a percentage of the carrying capacity. For example, If the local habitat will support 200 bunnies, and the actual population (P-number) is 50, then the relative population (B-number) is...

50/200 = 0.25 (which is the same as 25 percent)

Notice that the formula for the new population uses relative population (B-numbers), but the numbers you ask the user to enter and the numbers displayed will be actual populations (P-numbers), so you will have to convert back and forth between the two.

The growth rate represents how fast the bunnies are multiplying, as a percentage of the current population. The growth rate is typically computed as follows:

$g = b - d$

where:

b = birth rate (number of new baby bunnies per bunny per month)
d = death rate (number of dead bunnies per bunny per month),

Notice that the growth rate is a percentage of each day's population, so the number of new babies each month increases as the population grows.

To compute how fast the bunny population is growing, repeat these calculations for however many months are necessary. You MUST use the formula above and repeat the calculations with a loop. You must use double precision variables to do the calculations, but the current population (the number of bunnies) at the end of each month must be rounded to the nearest integer. (No fractional bunnies; too messy.) (Relative population must be a double, and actual population must be an integer.)

One warning: the actual population can exceed the carrying capacity - it is NOT a maximum. However, at this point, note that one of the terms in the equation becomes negative and the population will shrink.

Input and Output:

The program needs the following information from the user: the initial population (how many bunnies we start with), the birth rate, the death rate, and the carrying capacity. If the user requests the first option, the program also needs to know the length of time (the number of months), and then computes the final size of the population (the number of bunnies). For the second option, the program does the opposite: it needs to know the desired size of the population and computes the required length of time.

Check every user input for errors: initial population, carrying capacity, and number of months must be positive numbers. The goal population must be greater than or equal to the initial population and less than the carrying capacity. The birth rate must be non-negative (positive or zero). The death rate must be a number between 0 and 1 inclusive (1 and 0 are okay too). Also check every option for a valid entry.

For outputs, in addition to displaying the answer to the user's question, your program must also give the user the option to have the monthly growth displayed in a table. This table should have at least two columns: the month and the current number of bunnies. (Use tabs to make the columns line up.)

The table must be optional so the user can choose to turn it on or off. If I want to know how many bunnies I have after 1000 months, I don't want to wait while the program displays 1000 lines, but for shorter problems I might like to see how the herd is doing month to month.

Make your program easy to use. Start with a brief message identifying what the program does. Afterwards, include enough information in each input prompt so that the user will know what to enter without reading your code. Label the results thoroughly as well. For example, include the units (if appropriate) in all input and output. To make your messages easier to read and understand, print occasional blank lines, especially between the input prompts and the results.

Implementation:

Use the Scanner class for all input from the keyboard, and "System.out.println" or "....print" for all output to the screen. This will be a console-oriented application. Create only one class, containing the main program and the subprograms. This assignment will not involve defining new classes of objects.

You must have at least two subprograms, one for each option. These subprograms must match the following specifications exactly. For the first subprogram, compute (and return) the number of bunnies after a given length of time. For the second subprogram, compute (and return) the length of time to achieve a given number of bunnies. Both subprograms will also be given the initial population, the growth rate, and the carrying capacity, as well as a boolean parameter indicating whether or not the user wants the table of monthly results. There will be no input in either of these subprograms, and the only output is the table of intermediate results, if the user chooses to get it.

You also must have one subprogram that asks the user to repeat an entry until it is positive. The subprogram will be given the text of the input prompt, and return the final (positive) entry.

Program:

Follow the guidelines listed in your handout on software engineering standards. Start early and get the program working one piece at a time. For example, start by working on just the first question (how many bunnies?) for just one year. Then compute the result for multiple years. Then add the table of daily population growth. Then, finally, move on to the next option.

Turn in your program file, called program2.java, by email to simms@mesacc.edu

```
                        Logistic Model Example

Carrying Capacity: 100
Initial Population: 10
Birth Rate: 60%
Death Rate: 10%

Growth Rate = Birth - Death = 60% - 10% = 50% = 0.5
Pold = Initial Population = 10

Month 1:
      Bold = Pold / Carrying Capacity = 10/100 = 0.1
      Bnew = Bold + g X Bold X ( 1 - Bold )
            = 0.1 + 0.5 X 0.1 X ( 1 - 0.1 )
            = 0.1 + 0.5 X 0.1 X 0.9
            = 0.1 + 0.045
            = 0.145
      Pnew = Round( Bnew X Carrying Capacity )
            = Round( 0.145 X 100 )
            = Round( 14.5 )
            = 15

Month 2:

      Pold = Pnew = 15
      Bold = Pold / Carrying Capacity = 15 / 100 = 0.15
      Bnew = Bold + g X Bold X ( 1 - Bold )
            = 0.15 + 0.5 X 0.15 X ( 1 - 0.15 )
            = 0.15 + 0.5 X 0.15 X 0.85
            = 0.15 + 0.06375
            = 0.21375
      Pnew = Round( Bnew X Carrying Capacity )
            = Round( 0.21375 X 100 )
            = Round( 21.375 )
            = 21

Month Population
  0   10
  1   15
  2   21
  3   29
  4   39
  5   51
  6   63
  7   75
```