**CS 1400 Fundamentals of Programming**

**Programming Project #6**

**Rabbits, Rabbits, Rabbits**

**Version 1.0**

**Objective:**

At the completion of this Project, you will have created a Console application that:

* makes use of a ***UML Activity Diagram*** that describes the steps required to solve this problem,
* uses a loop statement,
* uses decision logic,
* uses arithmetic expressions, assignment, and
* properly formats the output.

**Project:**

This is another problem solving project. Since I want you to focus on the algorithmic part of the problem, you should write this as a Console Program. Once you have figured out the steps required to solve the problem, writing the code should be pretty easy. You should be familiar with accomplishing these steps by now:

1. Write down all the required variables, expressions, input and output that you know about the problem.
2. Draw pictures to help you visualize the problem.
3. Refine the expressions that your program needs to compute.
4. Refine any formulas, equations, or relationships that you might find useful.
5. Create an Activity Diagram. Then use the Activity Diagram as the basis for writing the Pseudo-Code and then use it to generate C# code that you submit. You will need to submit your Activity Diagram and pseudo-code as part of this project.

A scientist is doing some important research work, that requires her to use rabbits in her experiments. She starts out with one ***pair*** of adult rabbits (a male and a female). At the end of each month, a ***pair*** of rabbits produces ***one pair*** of ***offspring*** (a male and a female). However, these new offspring will not be able to reproduce until they are a month old, and won't have babies of their own until the following month. To illustrate this, consider the first two months:

* At the beginning of month one, the scientist just has the original pair of adult rabbits. The table for month one will look something like the following:

Month Adults Babies Total Pairs  
 1       1       0       1

* At the end of month one month, this pair of adults produces one ***pair*** of offspring. Thus, at the beginning of the month two the table will look like this:

Month Adults Babies Total Pairs  
 1       1       0       1  
 2       1       1       2

At the end of month two the adults have another pair of baby rabbits. The first pair of babies, born at the end of last month are not old enough to have babies yet, but we will categorize them as adults. So, at the beginning of month three the table looks like this:

Month Adults Babies Total Pairs  
 1       1       0       1  
 2       1       1       2  
 3       2       1       3

The scientist has ***500*** cages in which to hold her rabbits. Each cage holds one pair of rabbits. Assuming that no rabbits ever die, when will she run out of cages?

Your program must do the following:

1. Display a program header, introducing the program.
2. Display a table that contains the following information for the beginning of each month. Stop displaying the results, when you run out of cages.
3. The number of months that have passed.
4. The number adult rabbits (those over 1 month old).
5. The number of baby rabbits produced this month.
6. The total number of rabbit pairs in the lab.
7. Calculate and display how many months it will take until the number of rabbits exceeds the number of available cages.

Format and document your code in accordance with the course Style Guidelines.  Include a Project Prolog, identifying you as the author, and method Prolog’s.  Also include an electronic version of your Activity Diagram. Diagrams may be done in Word, PowerPoint or some other application and submitted as an image file. Submit your entire Project folder and Activity Diagram to Canvas.

**File(s) to Submit:**

Zip your entire Project folder, add your Activity Diagram to the zip file and name the zip file Proj\_06\_your-initials\_V1.0.zip. For example, I would name my file Proj\_06\_DAF\_V1.0.zip. Submit this assignment as Project #6 on Canvas.

**Hints**

If you need help solving this problem, you can find some hints below.

**Hints for Project Six**

The code for this project is quite simple. However, figuring out the algorithm will take some work. The best approach is to sit down with a paper and pencil and write down how and what is going on each month. As noted in the problem description you begin with ***one pair of rabbits***. At the end of one month that pair produces ***one pair of offspring***, so at the beginning of month two you have ***one adult pair of rabbits*** and ***one pair of baby rabbits***. Your total number of pairs of rabbits is two.

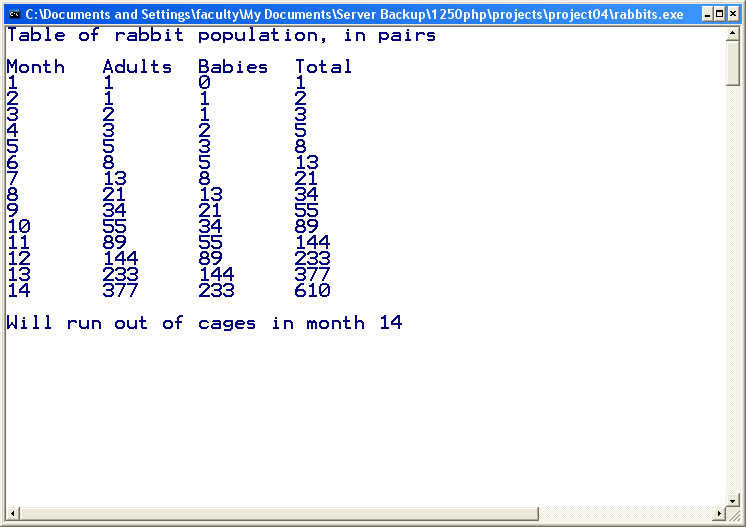
What happens the following month? Well, the initial pair of adult rabbits produces another pair of offspring. However, the pair of babies born at the end of last month isn't old enough to have babies yet. In order to keep track of this, you may want to define a third category of rabbits – rabbits previous adults and previous babies. Now write down what happens each month for several more months. Now, think carefully about what you did to calculate the number of rabbits each month. There is a simple arithmetic relationship between adults, babies, and almost adult rabbits. Write these steps down in an Activity Diagram. Now you are ready to convert your Activity Diagram into Pseudo-Code and finally into C# code.

To produce the table in the output, consider embedding tabs to move between columns. Your output line might look something like

**Console.WriteLine("{0}\t{1}\t{2}\t{3}", i, adultRabbits, babyRabbits, pairs);**

|  |  |  |
| --- | --- | --- |
|  | **Grading Checklist** |  |
| # | Program | C(correct)  X(incorrect) |
| 1 | Meets & works to specifications | 6 points |
| 2 | Error Free, elegant & efficient | 4 points |
| 3 | Pseudo-Code | -3 points |
| 4 | Style Guidelines | -2 points |
| 6 | Source Files(s) & Formatting | -2 points |
| 7 | Project Prolog | -1 points |
| 8 | Function Prologs | -1 points |
| 9 | Zip Filename | -1 points |
| 10 | Lab & Project Names | -1 points |
| 11 | Zip File is invalid or will not unzip | Lab = 0 pts |
|  | Total Points | 10 | 0-9 |

**Sample Output:**



You can get an executable that runs correctly on Canvas.