12.05 Assignment Instructions

Instructions: Write a program to recursively calculate the *n*th Fibonacci number.

1. Create a new project called 12.05 Fibonacci in the Mod12 Assignments folder.
2. The program will follow an object-oriented design and include a client and implementation class. You may write the code in one file or two. Create classes as needed and use appropriate names of your choosing.
3. Review the **Background Information** section below to learn more about Fibonacci number sequences.
4. The program should allow the user to enter an integer value, and the project should calculate the Fibonacci number for that value. For example, if n = 4, the Fibonacci number should be **3**.
5. The program should allow the user to continue entering numbers to request another Fibonacci number until they choose to quit.

Through experimentation, determine the range of Fibonacci numbers your program can calculate. Be sure to have the Debugger open so you can terminate the program as needed during your testing. Once the upper limit of the range is determined, make sure the prompt instructs the user to enter numbers within the range. Provide an error trap to catch user input that is out of bounds in order to prevent the program from crashing.

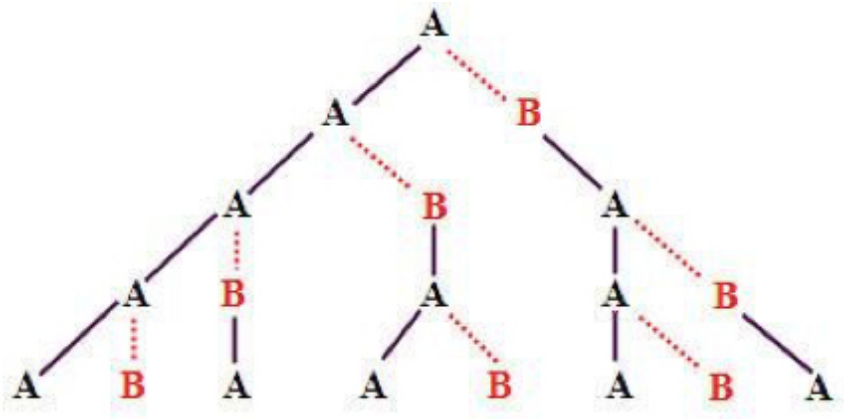
Background Information: Fibonacci numbers

In *Liber Abaci* (1202), Leonardo of Pisa described the growth of a hypothetical rabbit population with the following assumptions:

- The first month, there is just one pair of rabbits.
- Newborn pairs become fertile after one month.
- Each month, every fertile pair produces a new pair (one male and one female).
- Rabbits never die.

The monthly increase in pairs of rabbits each generation is depicted in the following table. After starting with two numbers, each number that follows is the sum of the two preceding numbers.

Month	Pairs of Rabbits	Pairs of Adults	Pairs of Babies	Total Pairs
01/01		1	0	1
02/01		1	1	2
03/01		2	1	3

Month	Pairs of Rabbits	Pairs of Adults	Pairs of Babies	Total Pairs
04/01		3	2	5
05/01		5	3	8
06/01		8	5	13
07/01		13	8	21
08/01		21	13	34
09/01		34	21	55
10/01		55	34	89
11/01		89	55	144
12/01		144	89	233

This series of numbers was known much earlier in India, but subsequently became associated with Fibonacci.

The Fibonacci series can also be expressed as a Piecewise function as shown below:

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f(n-1) + f(n-2) & \text{if } n > 1 \end{cases}$$

Notice that there are two base cases for the Fibonacci series.

The first few numbers of the Fibonacci series are given below:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, …

Before attempting to write the program, you should try and solve the Piecewise function using the S-S-S Strategy for several values of ***n***. When you write the program, remember to do the following:

1. Know when to stop. This is the destination (the base case).
2. Decide how to take one step to solve the problem (the recursive call).
3. Break the task down into one step plus the rest of the problem.

Use your recursive method template and the Steps to Writing Recursive Methods:

1. Give the method a name.
2. Complete the parameter list in the method header.
3. Decide whether the method will be void or return something.
4. Write the base case condition.
5. Write the action to be taken if the base case is true.
6. Write the recursive call. The parameter list must match the method header.

Do not let the two base cases or the two functions in the recursive call stump you. Use your recursive instinct and follow the previously-established patterns.

Print