Independent Assignment – Recursion

**Instructions:**

1. Read the recursion notes on the course site.
2. Review the Triangle class example discussed in class using both the iteration method and the recursion method. Check that the two programs produce the same results.
3. Complete parts A, B, and C below in **one single java class**.
4. Submit your program and this document **online**

**Part A:**

Factorial is a popular mathematical operation. It is defined as follows:

The factorial of a non-negative integer *“n”,* written “” (and pronounced “n factorial”), is defined as the product:



For example:

 which is equal to 120.

There are two special cases: and.

The following program calculates the factorial of any number using several iterations of a simple loop – meaning running through a loop several times.

|  |
| --- |
| import javax.swing.\*;  public class Factorial  {  public static long factorial (long number)  {  long fact = 1;  for (long count = number ; count >= 1 ; count--)  {  fact = fact \* count;  }  return fact;  }  public static void main (String [] args)  {  String input = JOptionPane.showInputDialog("Enter a non-negative integer: ");  long num = Long.parseLong(input);  JOptionPane.showMessageDialog ( null,num + "! = " + factorial(num),  }  } |

Change the above factorial method above using the new recursive method below.

Notice that given the above formula for factorial it can be expressed as follows:



Also note that the special case for the recursive method is that 0! = 1 and 1! = 1. Therefore, the **pseudo code** for the recursive factorial method can be stated as follows:

|  |
| --- |
| method factorial (large integer number) returns large integer   1. if (number <= 1)    1. returns 1 2. otherwise    1. returns fact = number × call factorial (number -1 ) |

**Note**: The main method should not change.

**Part B:**

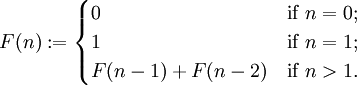
A string can be reversed by using the following method:

|  |
| --- |
| public static String reverseWord(String word)  {  String letter, reversed = ""; // 2 variables for a letter and reversed string    for (int i = word.length()-1; i >=0; i--) // Start at the end of the word and  {  letter = word.substring(i, i+1); // take one letter at a time  reversed = reversed + letter; // add it to the end of a new string  }    return reversed; // return the reversed word  }//reverseWord |

1. Rewrite the same method using recursion.
2. Add code to your main method to show that it works by calling your new method.

**Part C:**

The **Fibonacci numbers** form a sequence defined by the following recurrence relation:



Add a method named fibbonacci to your program and add code to your main method so that it displays a list of the Fibonacci numbers given the upper limit.

For Example:

If the number 15 is entered, then the following sequence would be displayed:

0, 1, 1, 2, 3, 5, 8, 13,

If the number 100 is entered, then the following sequence would be displayed:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,

and so on…

Evaluation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Category**  **(Thinking/Application)** | **Marking Scheme** | | | | **Comments** |
| **R** | **Level 1 - 2** | **Level 3 – 3+** | **Level 4 – 4+** |
| **Part A:** Recursive program works as expected. (10 marks) (T) |  |  |  |  |  |
| **Part B:** Recursive program works as expected. (10 marks) (A) |  |  |  |  |
| **Part C:** Recursive program works as expected. (10 marks) (A) |  |  |  |  |
| Programs are fully commented (10 marks) (C) |  |  |  |  |

Thinking: \_\_\_\_\_ Application: \_\_\_\_\_ Communication \_\_\_\_\_