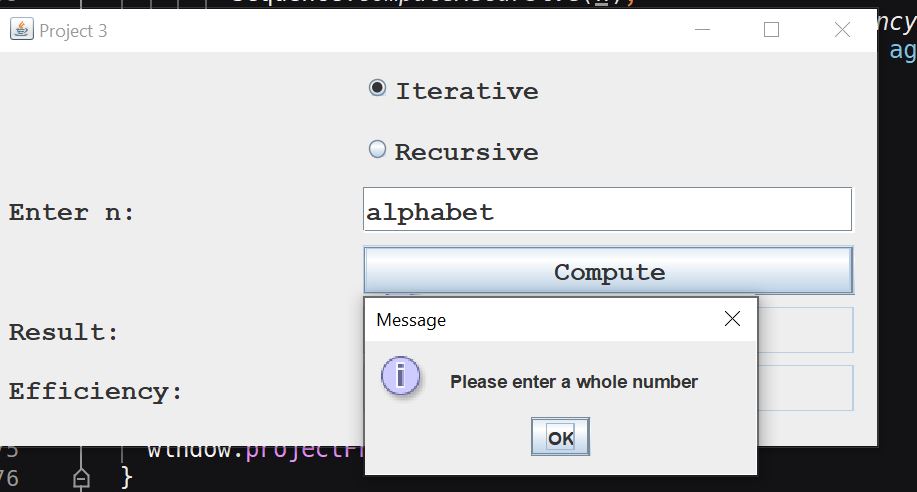
Ben Brandhorst

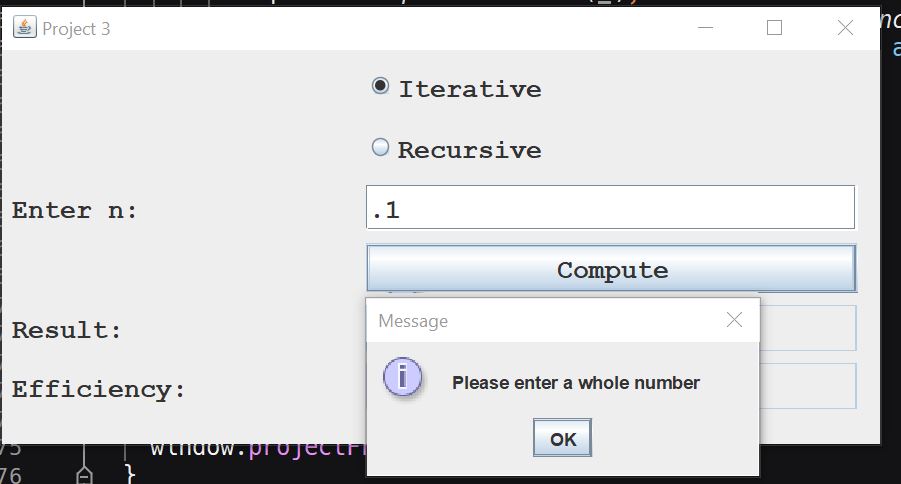
CMIS 242

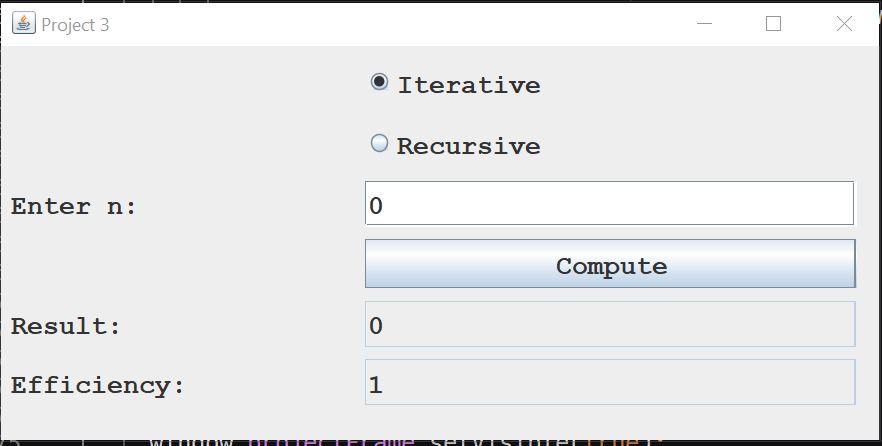
April 26th, 2019

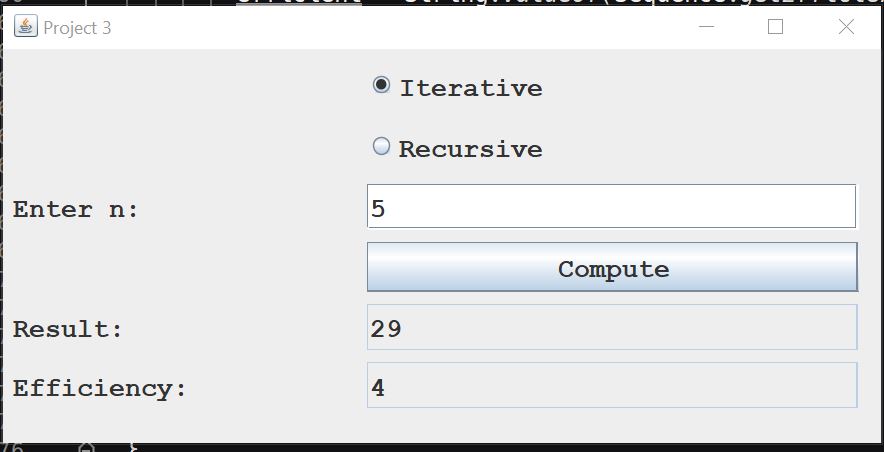
**Project 3  
Number Sequence Calculation**

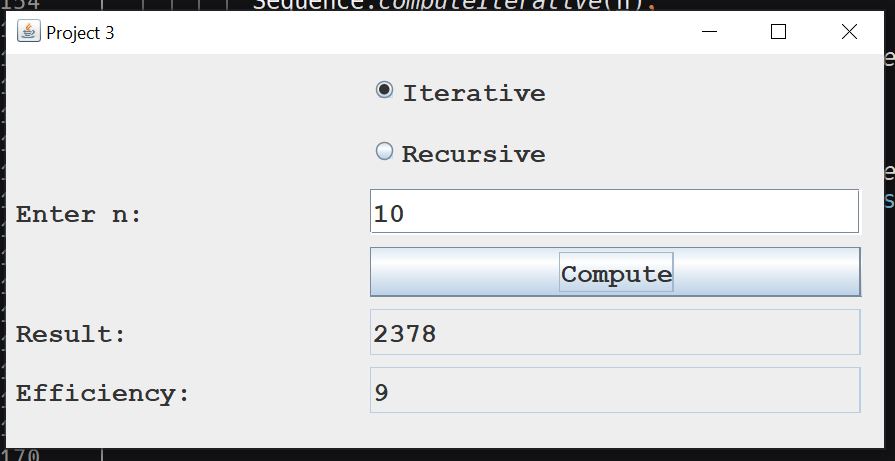
|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **PASS??** |
| **Radio Button Selected:** Iterative  **Enter n Entry:** alphabet | **JOptionPane:”** Please enter a whole number” | **JOptionPane:”** Please enter a whole number” | YES |
| **Radio Button Selected:** Iterative  **Enter n Entry:** .1 | **JOptionPane:”** Please enter a whole number” | **JOptionPane:”** Please enter a whole number” | YES |
| **Radio Button Selected:** Iterative  **Enter n Entry:** 0 | **Result:** 0  **Efficiency:** 1 | **Result:** 0  **Efficiency:** 1 | YES |
| **Radio Button Selected:** Iterative  **Enter n Entry:** 5 | **Result:** 29  **Efficiency:** 4 | **Result:** 29  **Efficiency:** 4 | YES |
| **Radio Button Selected:** Iterative  **Enter n Entry:** 10 | **Result:** 2378  **Efficiency:** 9 | **Result:** 2378  **Efficiency:** 9 | YES |
| **Radio Button Selected:** Iterative  **Enter n Entry:** 36 | **Result:** 1,260,868,716  **Efficiency:** 35 | **Result:** 1260868716  **Efficiency:** 35 | YES |
| **Radio Button Selected:** Iterative  **Enter n Entry:** 37 | **Result:** 3,409,318,269  **Efficiency:** 36 | **Result:** -885649027  **Efficiency:** 36 | NO |
| **Radio Button Selected:** Recursive  **Enter n Entry:** 0 | **Result:** 0  **Efficiency:** 1 | **Result:** 0  **Efficiency:** 1 | YES |
| **Radio Button Selected:** Recursive  **Enter n Entry:** 1 | **Result:** 1  **Efficiency:** 1 | **Result:** 1  **Efficiency:** 1 | YES |
| **Radio Button Selected:** Recursive  **Enter n Entry: 5** | **Result:** 5  **Efficiency:** 15 | **Result:** 5  **Efficiency:** 15 | YES |
| **Radio Button Selected:** Recursive  **Enter n Entry:** 36 | **Result:** 1,260,868,716  **Efficiency:** unk | **Result:** 1260868716  **Efficiency:** 48315633 | YES |
| **Radio Button Selected:** Recursive  **Enter n Entry:** 37 | **Result:** 3,409,318,269  **Efficiency:** unk | **Result:** -885649027  **Efficiency:** 78176337 | NO |

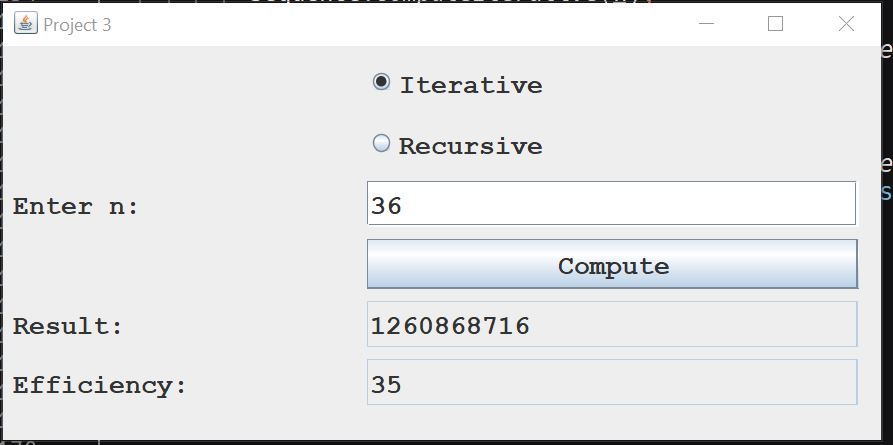
**Test Case 1 Output Example: (Non numeric entry))**

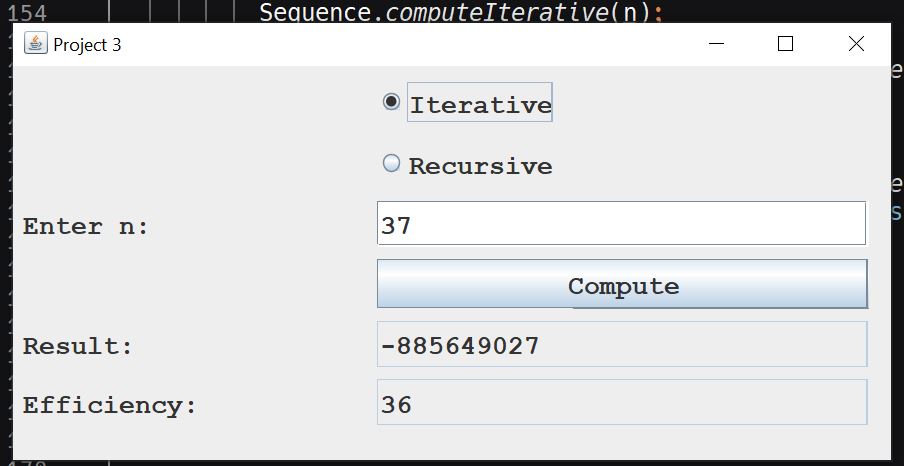
**Test Case 2 Output Example: (Non integer entry))**

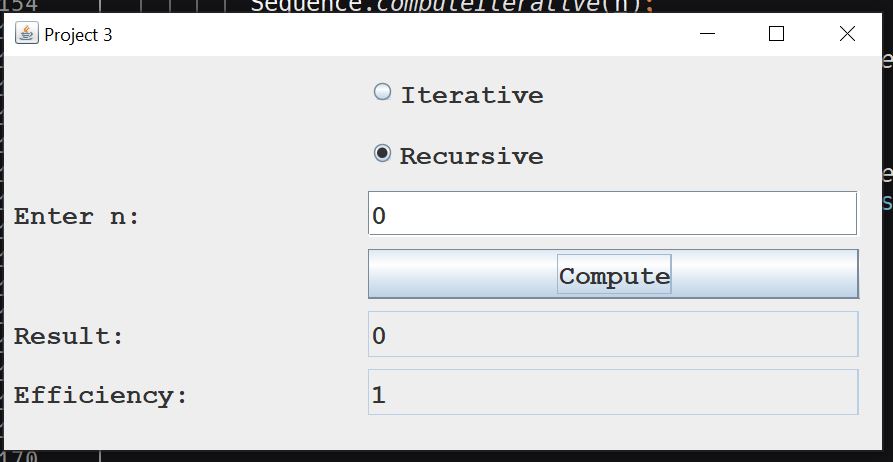
**Test Case 3 Output Example: (0 entry, Iterative))**

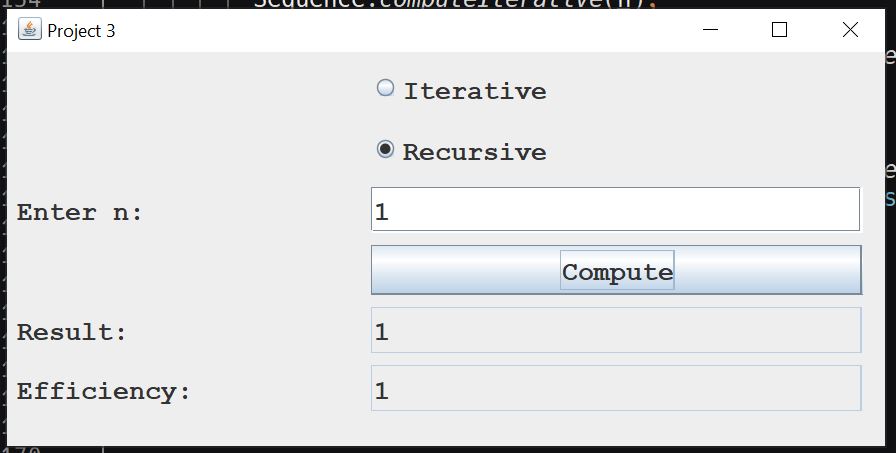
**Test Case 4 Output Example: (5 entry, Iterative))**

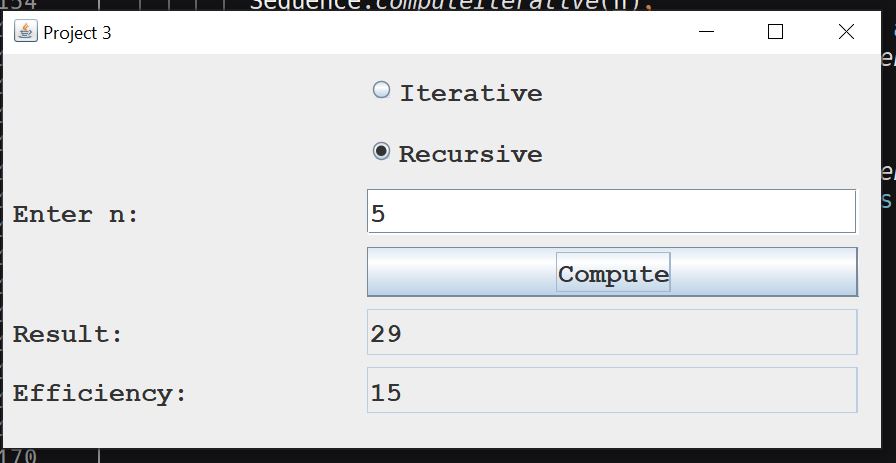
**Test Case 5 Output Example: (10 entry, Iterative))**

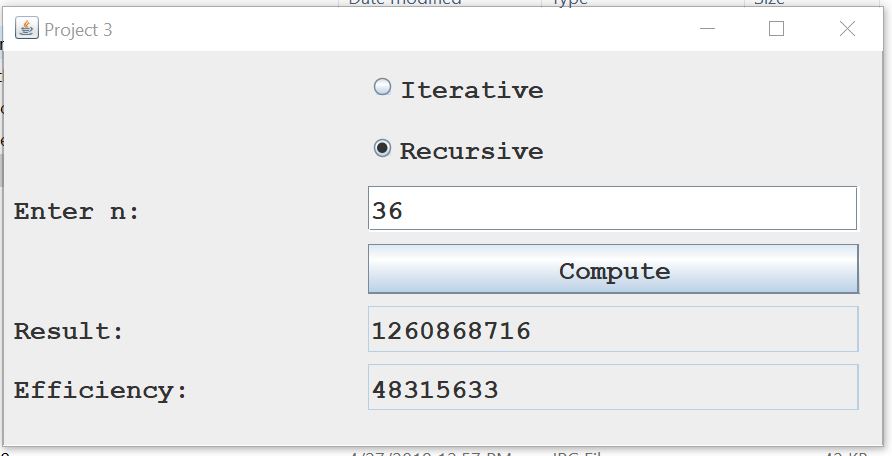
**Test Case 6 Output Example: (36 entry, Iterative))**

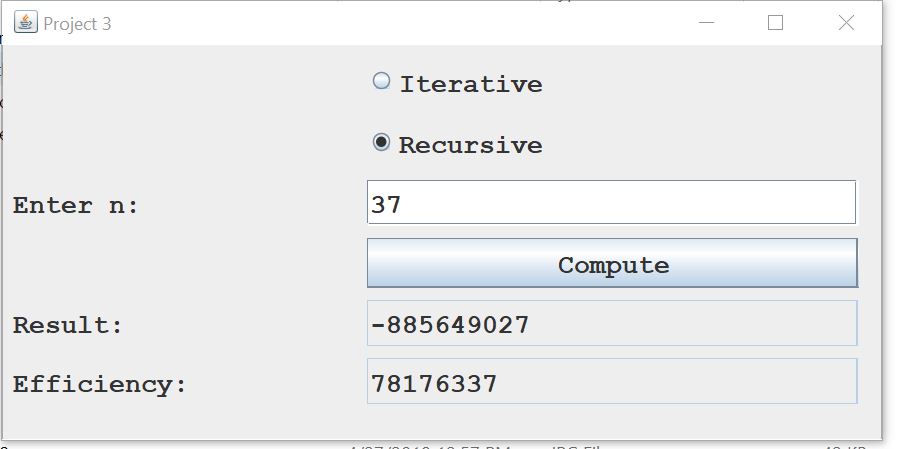
**Test Case 7 Output Example: (37 entry, Iterative))**

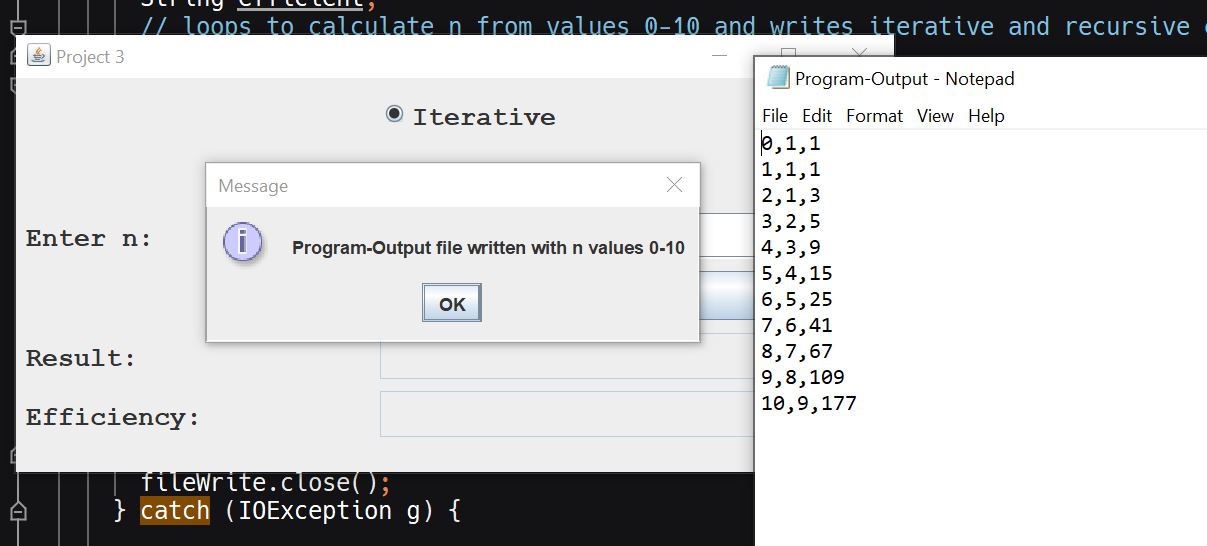
**Test Case 8 Output Example: (0 entry, Recursive))**

**Test Case 9 Output Example: (1 entry, Recursive))**

**Test Case 10 Output Example: (5 entry, Recursive))**

**Test Case 11 Output Example: (36 entry, Recursive))**

**Test Case 12 Output Example: (37 entry, Recursive))**

**Window Close (Message and Text File Output)**

The iterative method cycles through its loop at a constant rate. Using the iterative method to find the term for n when n is 0, 1, or 2 requires a single loop. For any integer greater than 2, the iterative method requires one less loop than the integer entered for n (e.g. if a user enters 5 for “Enter n:” the efficiency returned is 4).

The recursive method on the other hand requires many more loops to calculate the same information. As users try to calculate larger numbers in the sequence, the number of loops required grows at an extremely fast rate. For example: If users input 36 for “Enter n:” the iterative method requires 35 loops to calculate for n. The recursive method on the other hand requires 48,315,633 loops to calculate the same thing.