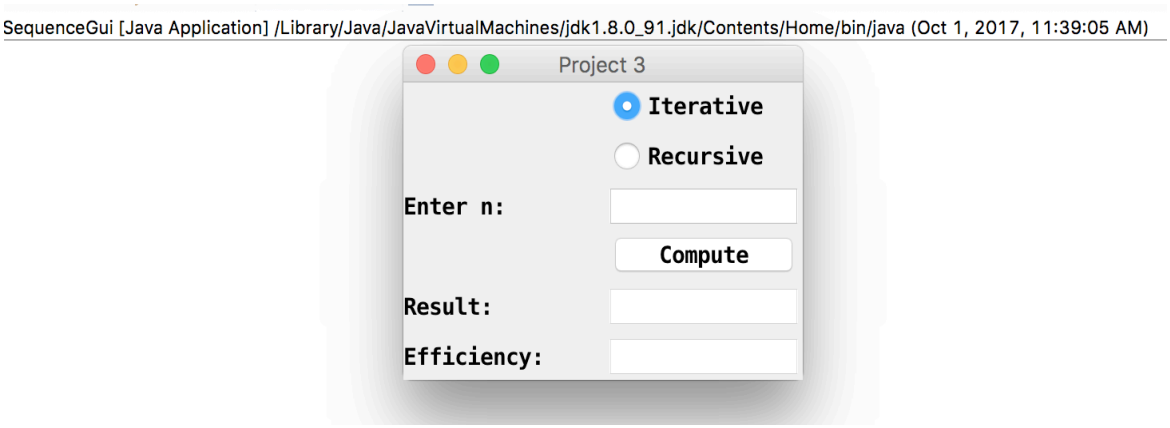
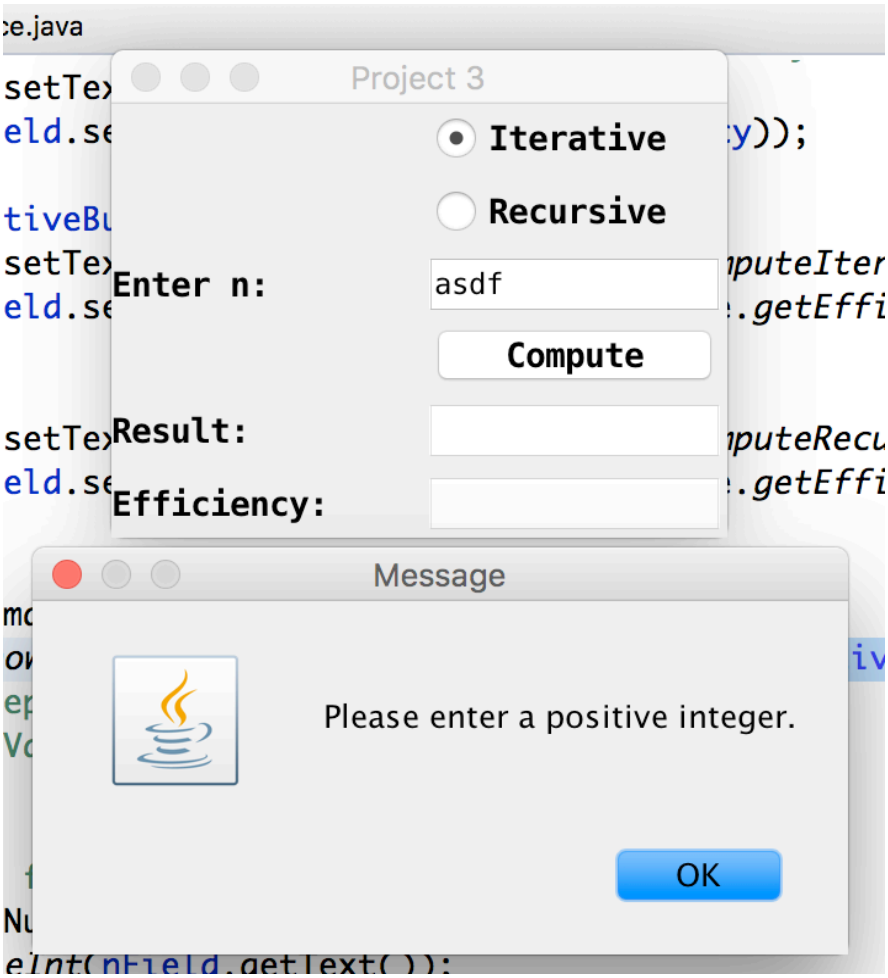
Nicole Donnelly
CMIS 242
Project 3
October 1, 2017

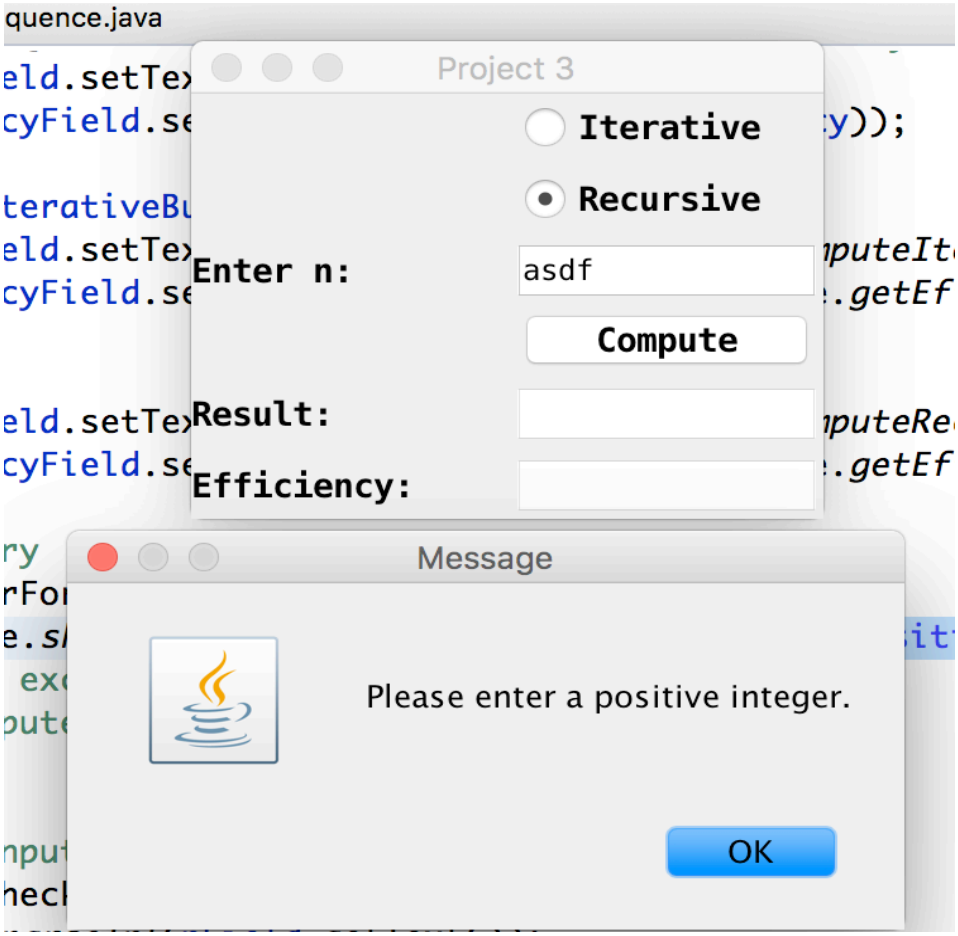
Screenshot – Successful Compilation



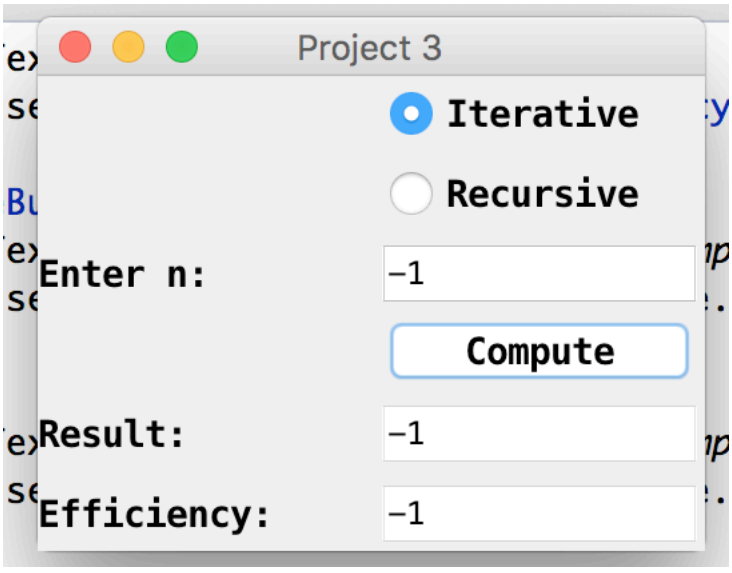
Screenshot – Test Case 1



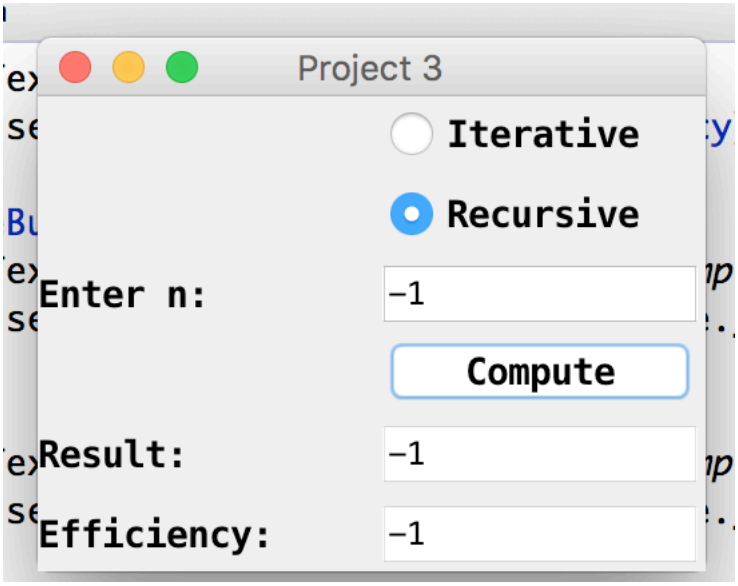
Screenshot – Test Case 2



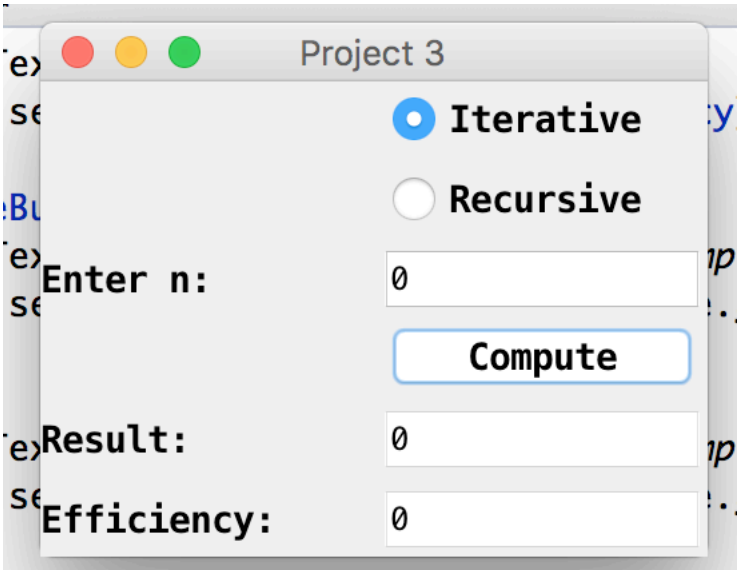
Screenshot – Test Case 3



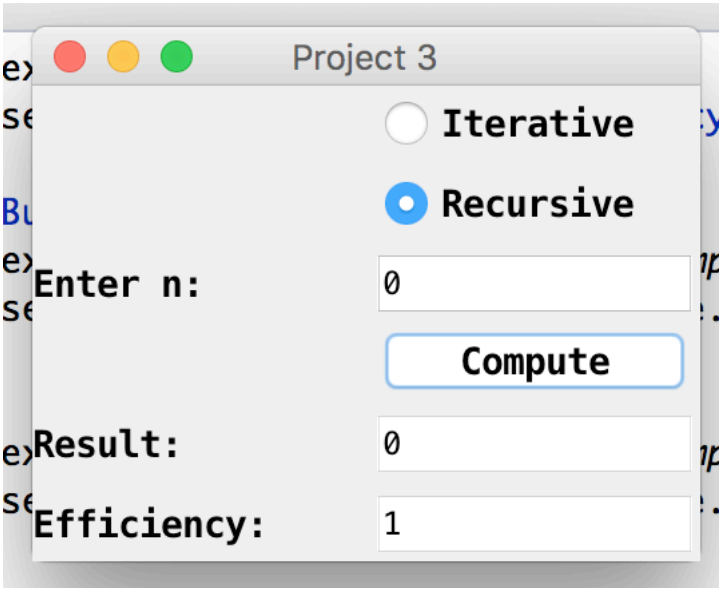
Screenshot – Test Case 4



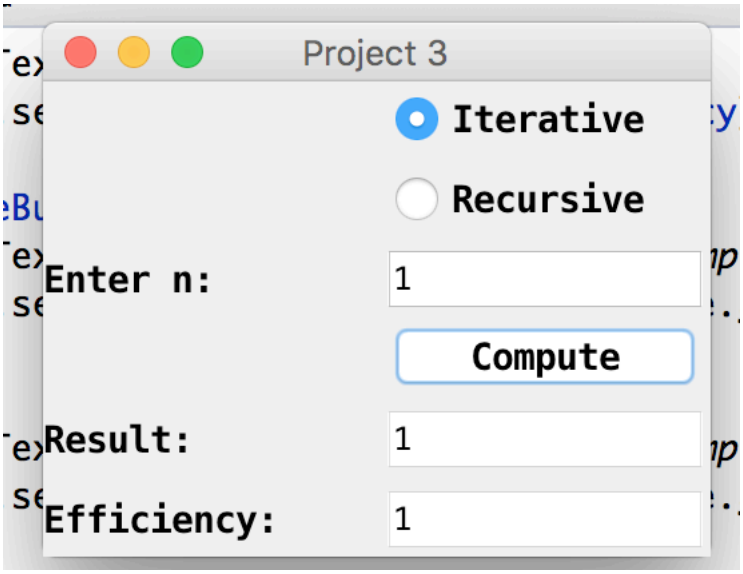
Screenshot – Test Case 5



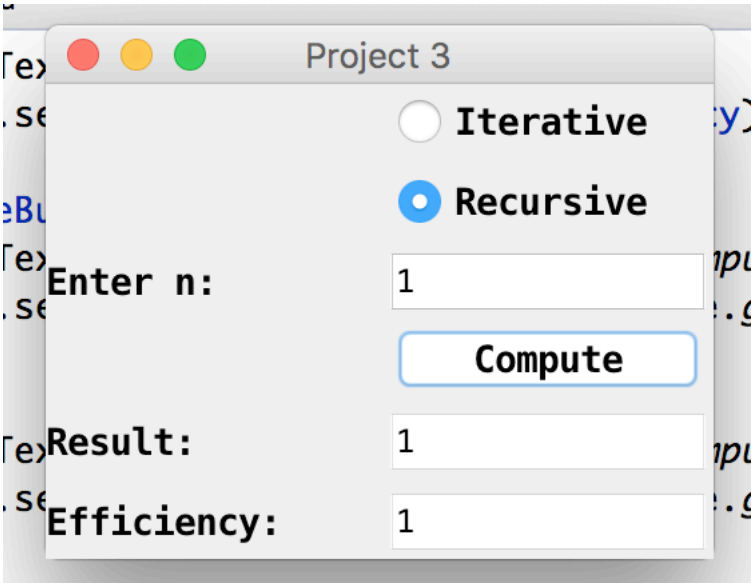
Screenshot – Test Case 6



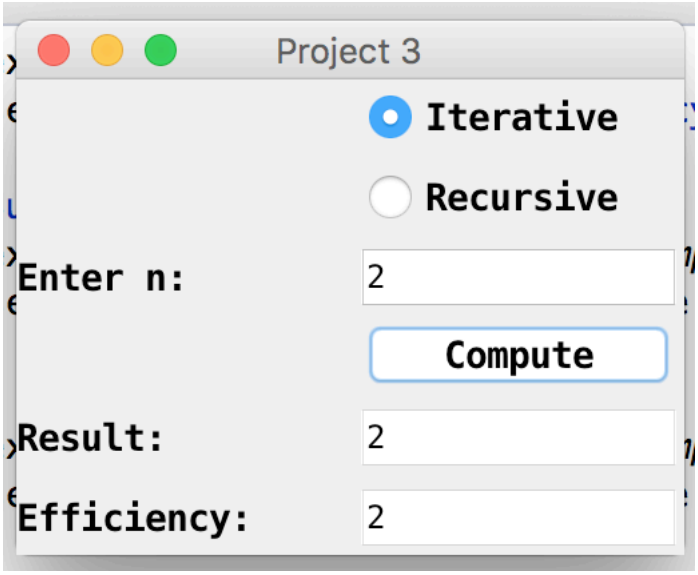
Screenshot – Test Case 7



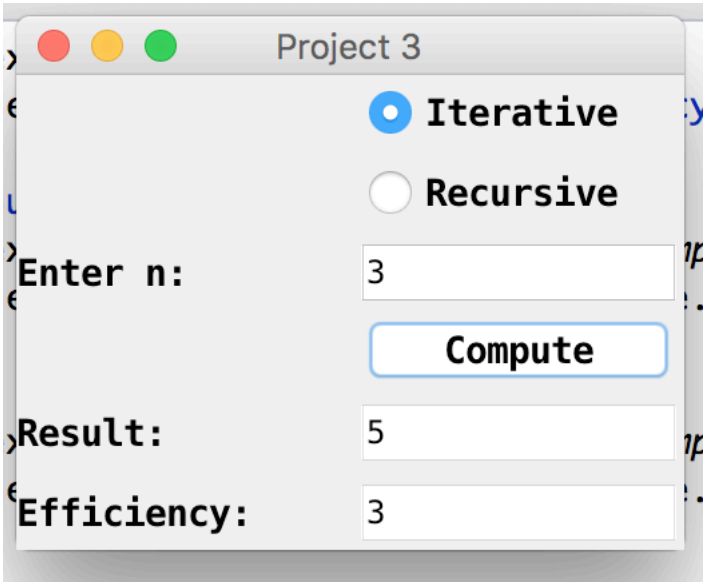
Screenshot – Test Case 8



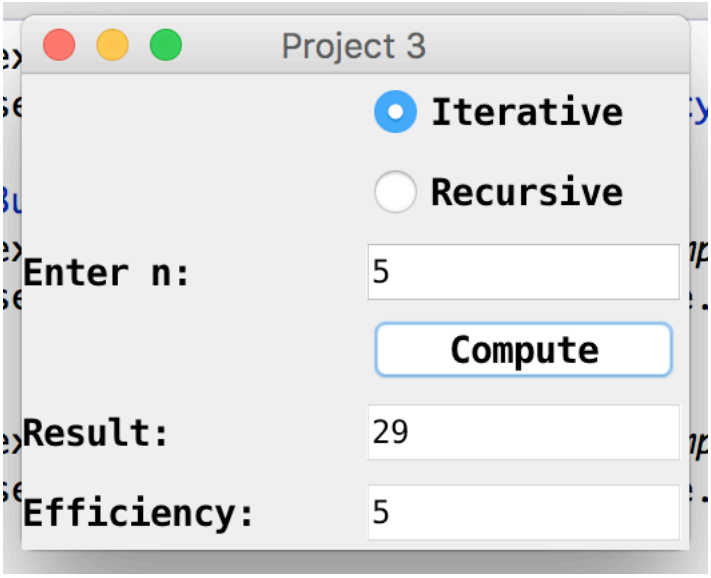
Screenshot – Test Case 9



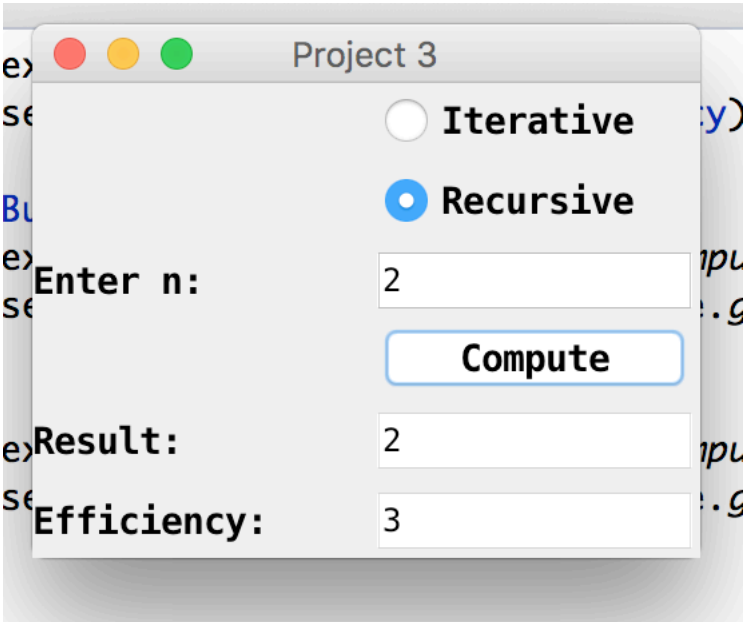
Screenshot – Test Case 10



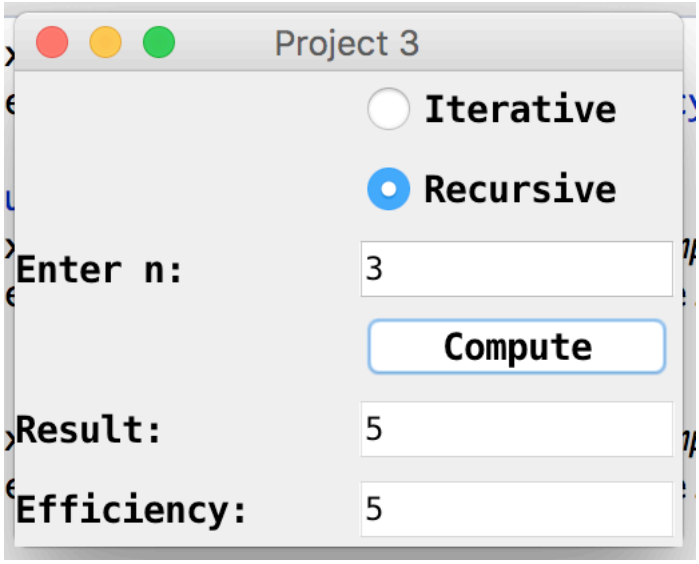
Screenshot – Test Case 11



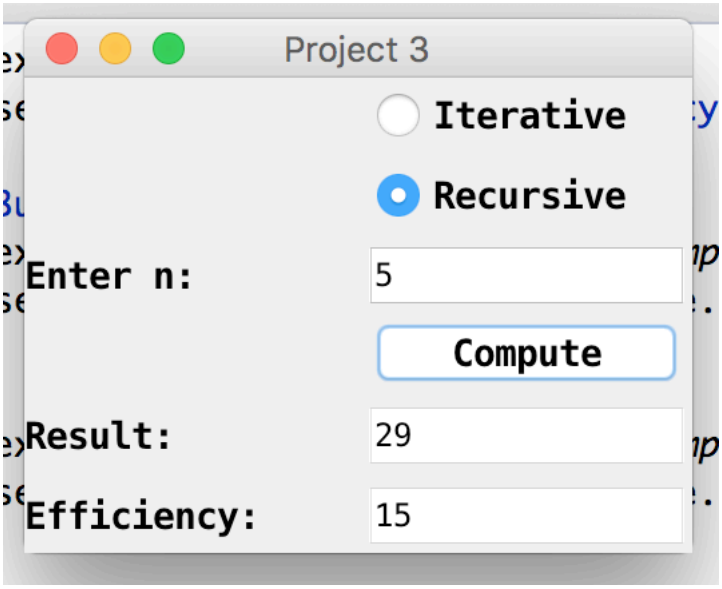
Screenshot – Test Case 12



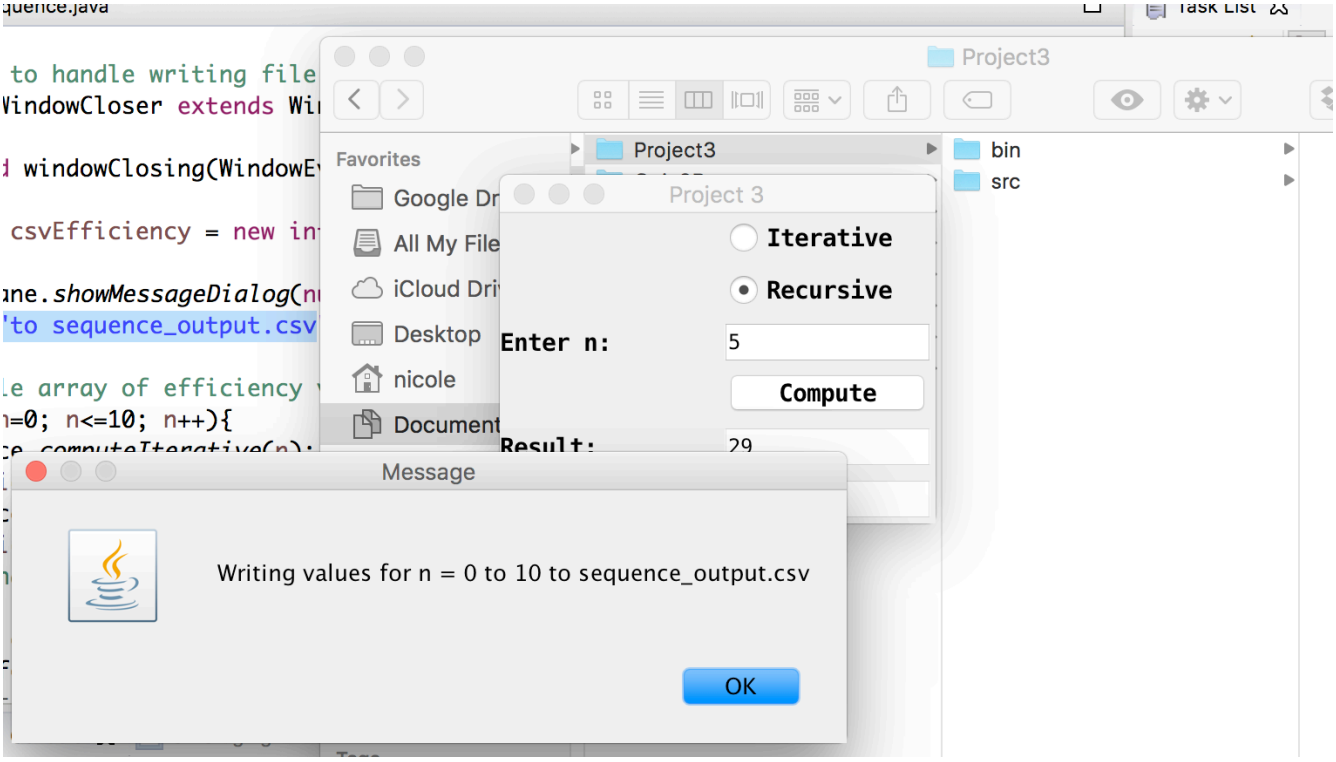
Screenshot – Test Case 13

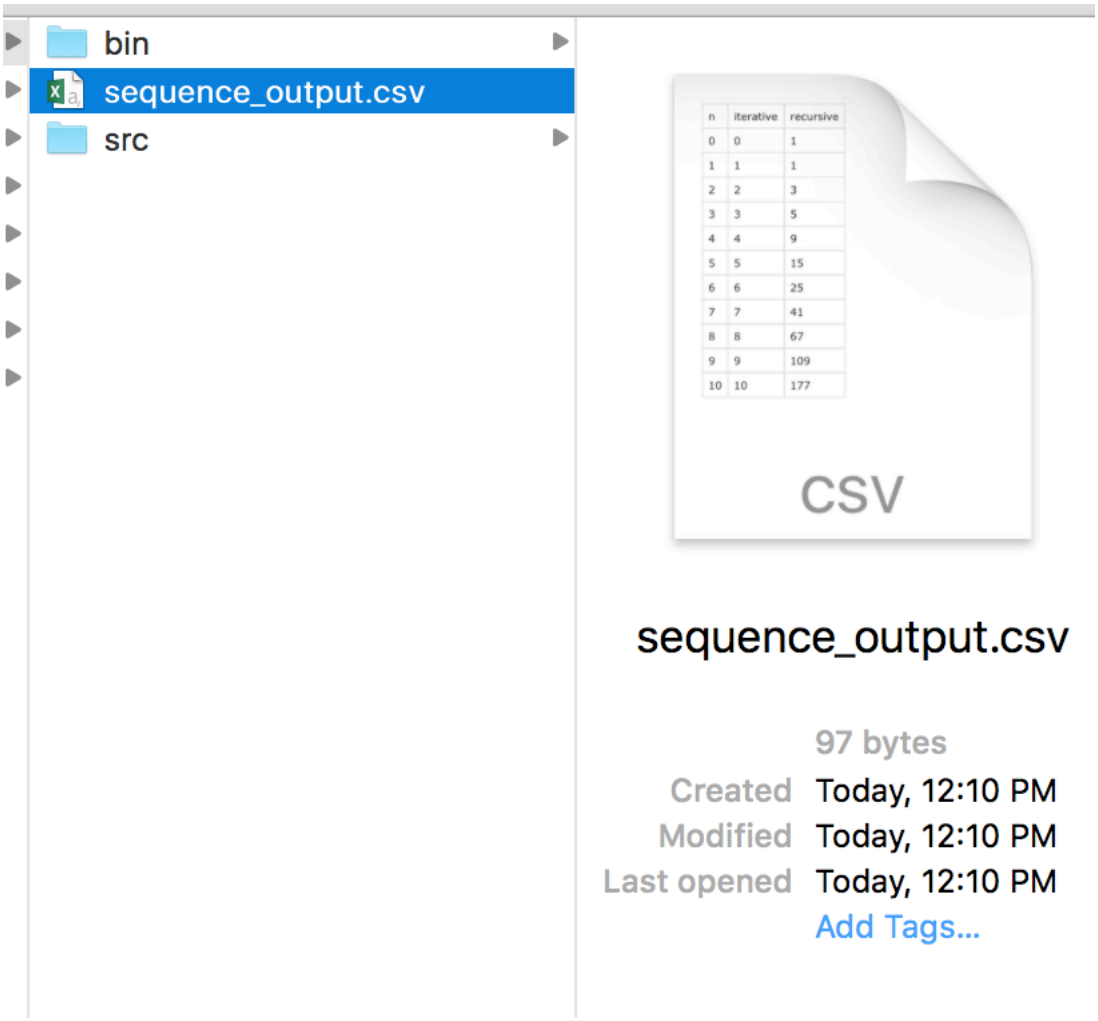


Screenshot – Test Case 14



Screenshots – Test Case 15





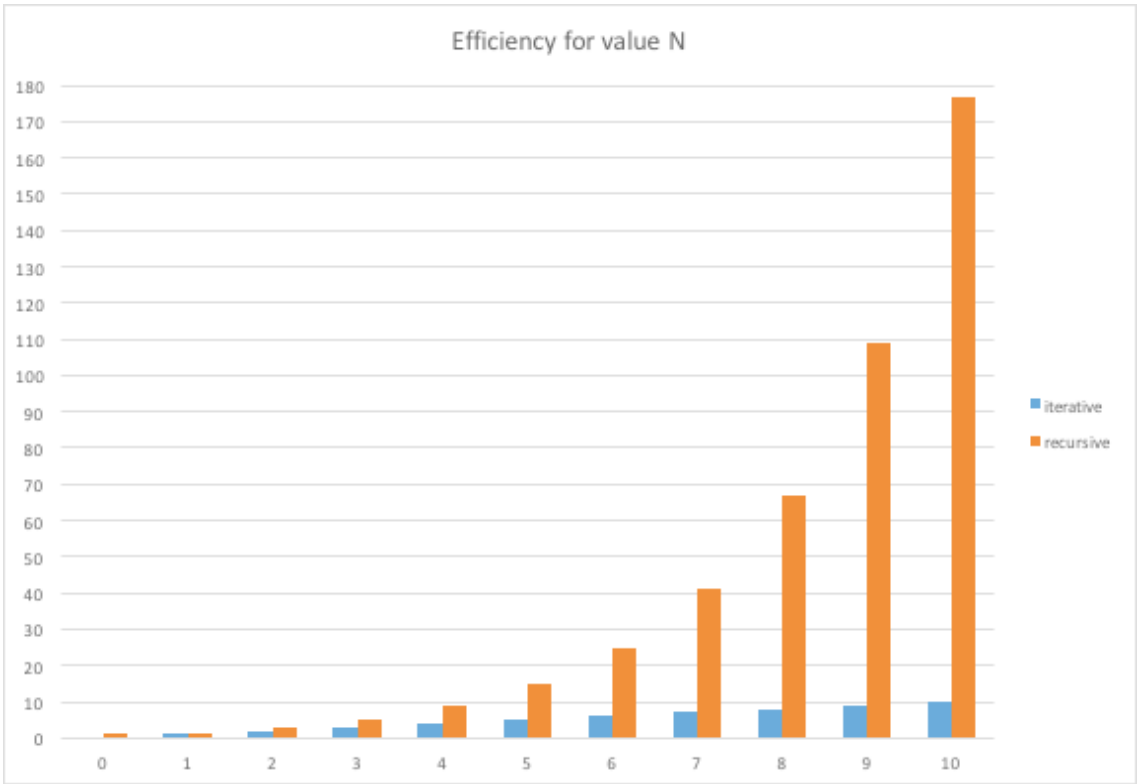
	A	B	C
1	n	iterative	recursive
2	0	0	1
3	1	1	1
4	2	2	3
5	3	3	5
6	4	4	9
7	5	5	15
8	6	6	25
9	7	7	41
10	8	8	67
11	9	9	109
12	10	10	177

Test Case	Expected Output	Actual Output	Pass?
1. Iterative, n = asdf	"Please enter a positive integer."	See Test Case 1 screenshot	Yes
2. Recursive, n=asdf	"Please enter a positive integer."	See Test Case 2 screenshot	Yes
3. Iterative, n = -1	Result = -1, Efficiency = -1	See Test Case 3 screenshot	Yes
4. Recursive n = -1	Result = -1, Efficiency = -1	See Test Case 4 screenshot	Yes
5. Iterative, n=0	Result = 0, Efficiency = 0	See Test Case 5 screenshot	Yes
6. Recursive, n=0	Result = 0, Efficiency = 1	See Test Case 6 screenshot	Yes
7. Iterative, n=1	Result = 1, Efficiency = 1	See Test Case 7 screenshot	Yes
8. Recursive, n=1	Result = 1, Efficiency = 1	See Test Case 8 screenshot	Yes
9. Iterative, n=2	Result = 2, Efficiency = 2	See Test Case 9 screenshot	Yes
10. Iterative, n=3	Result = 5, Efficiency = 3	See Test Case 10 screenshot	Yes
11. Iterative, n=5	Result = 29, Efficiency = 5	See Test Case 11 screenshot	Yes
12. Recursive, n=2	Result = 2, Efficiency = 3	See Test Case 12 screenshot	Yes
13. Recursive, n=3	Result = 5, Efficiency = 5	See Test Case 13 screenshot	Yes
14. Recursive, n=5	Result = 29, Efficiency = 15	See Test Case 14 screenshot	Yes
15. close program	“Writing values for n = 0 to 10 to sequence_output.csv” and file created with output	See Test Case 15 screenshot	Yes

Efficiency Chart

The below chart compares the efficiency of the iterative and recursive functions for values of n between 0 and 10. The iterative efficiency is measured by the number of times the loop in the method iterates. In the case of n=0, the loop in the iterative method is not run at all. The iterative loop will always run n times.

The recursive efficiency is measured by the number of times the recursive method is called. In the base cases of n = 0 and n = 1, the recursive method only gets called once. As n increases, additional calls to the recursive method will be made as each value of n gets broken down to the base cases.

For example, when n=2 the efficiency is 3: one call is made for n=2, which then makes 2 calls for n=1 and n=0 (1 + 1 + 1). When n=3 the efficiency is 5 because the method is called once for the n value of 3 then you have the efficiency of n=2 plus the efficiency of n=1, i.e. 1 + e(n=2) + e(n=1). Similarly, the efficiency of :

n=4 is 9: 1 + e(n=3) + e(n=2) = 1 + 5 + 3
n=5 is 15: 1 + e(n=4) + e(n=3) = 1 + 9 + 5