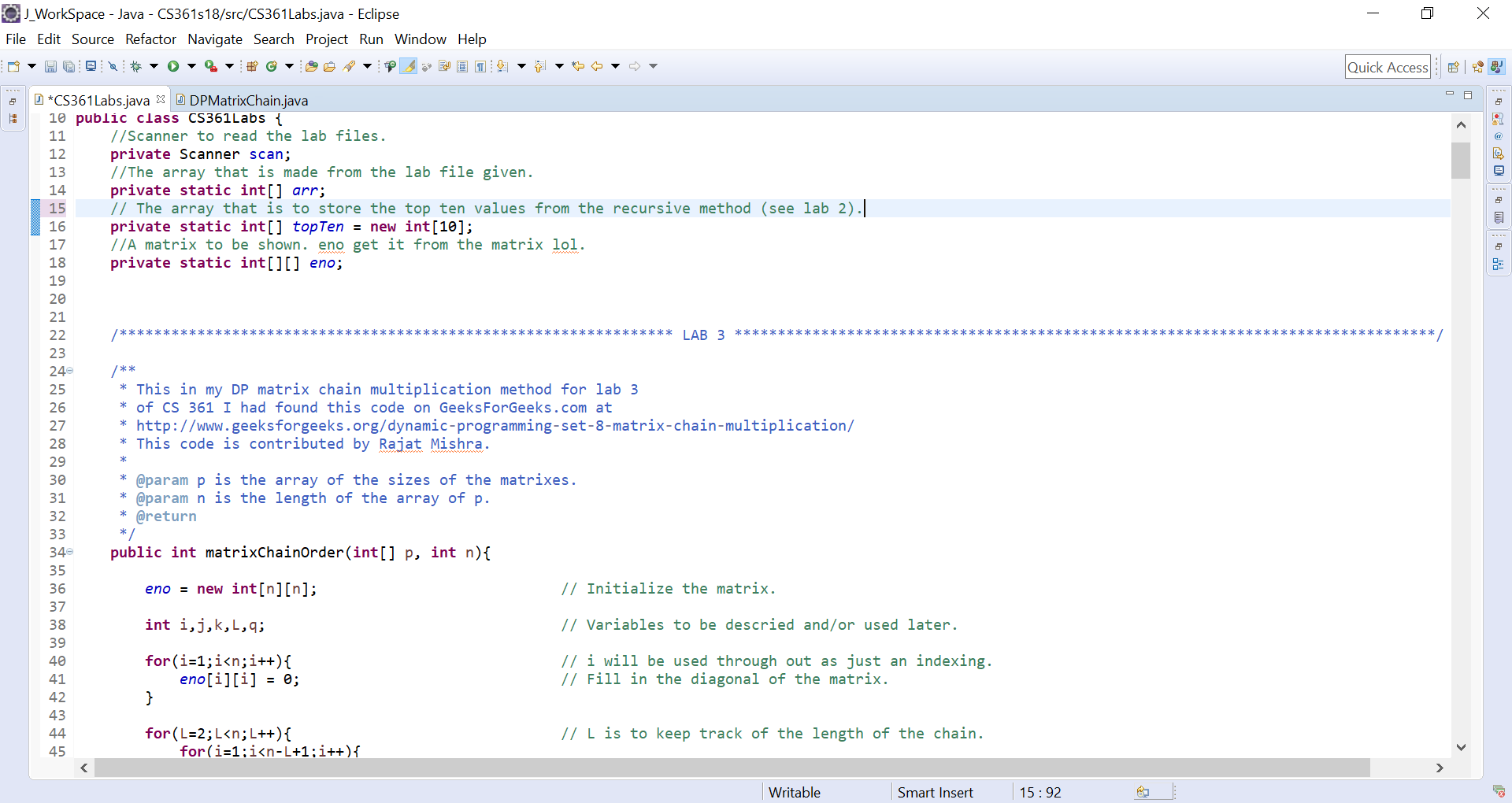
# CS361 Algorithm Lab 3

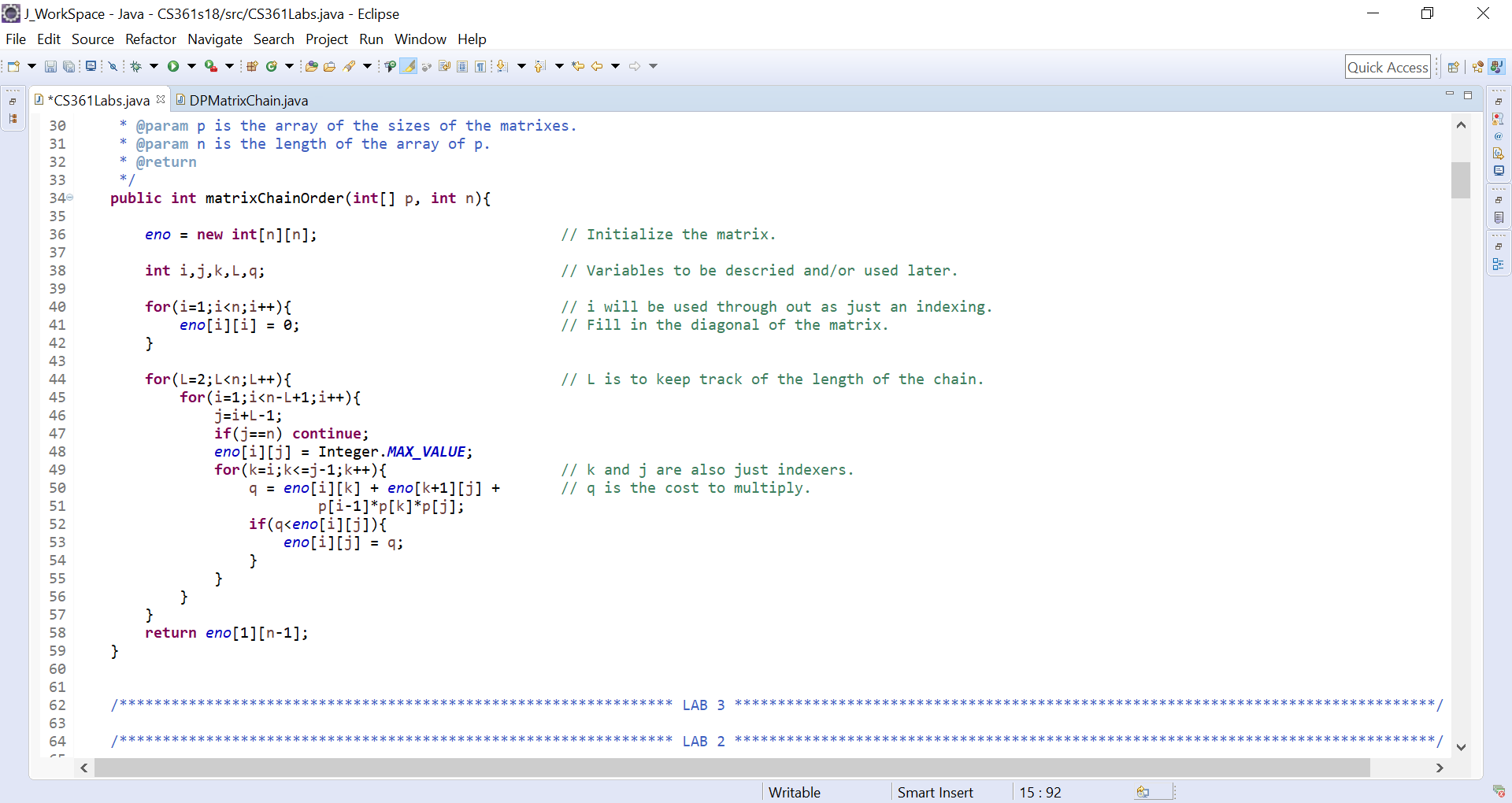
Nathan Stark

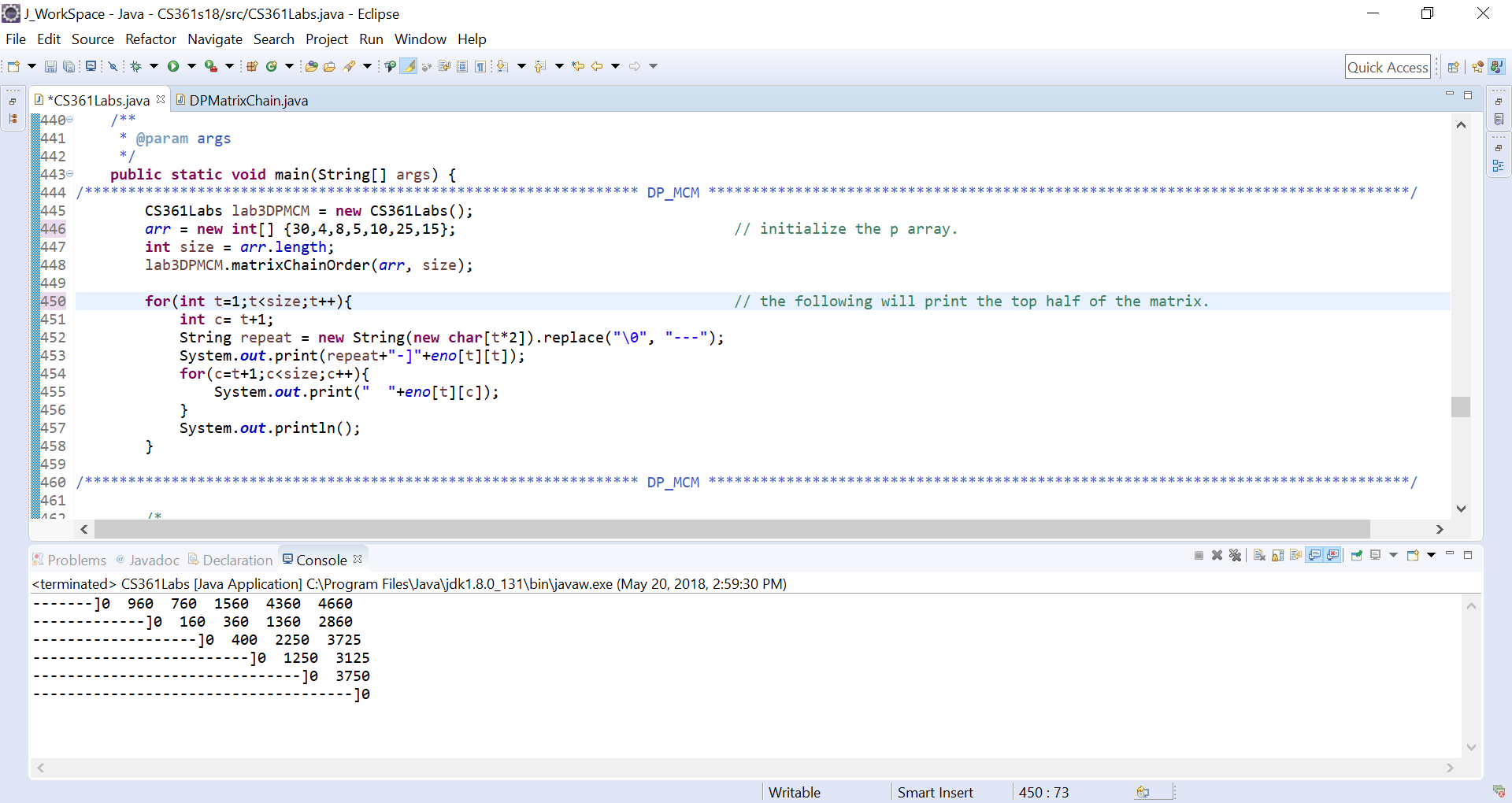
CS 361 Lab 3

# What to do

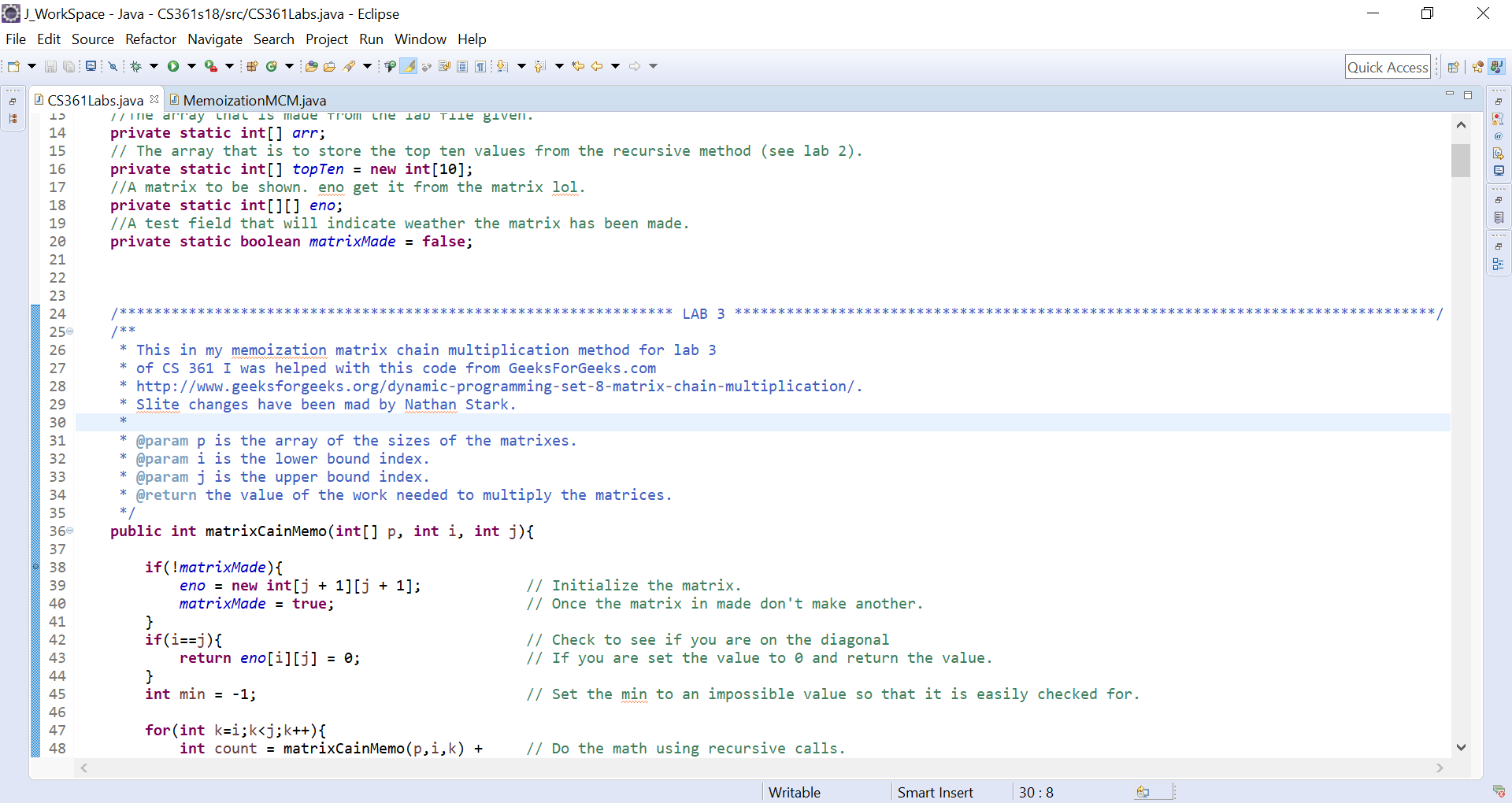
1. Implement the DP version of MCM algorithm. Show commented code.

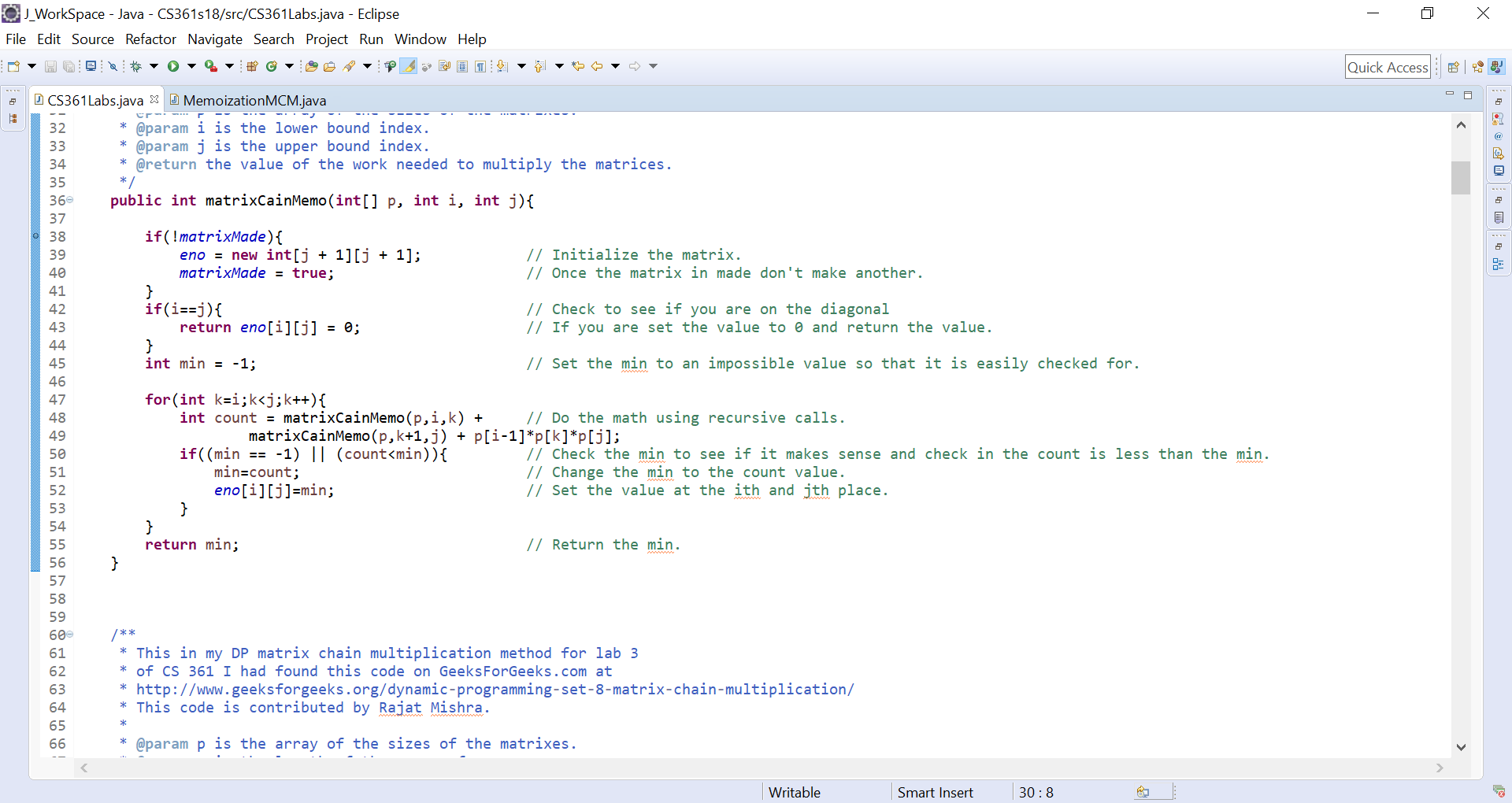




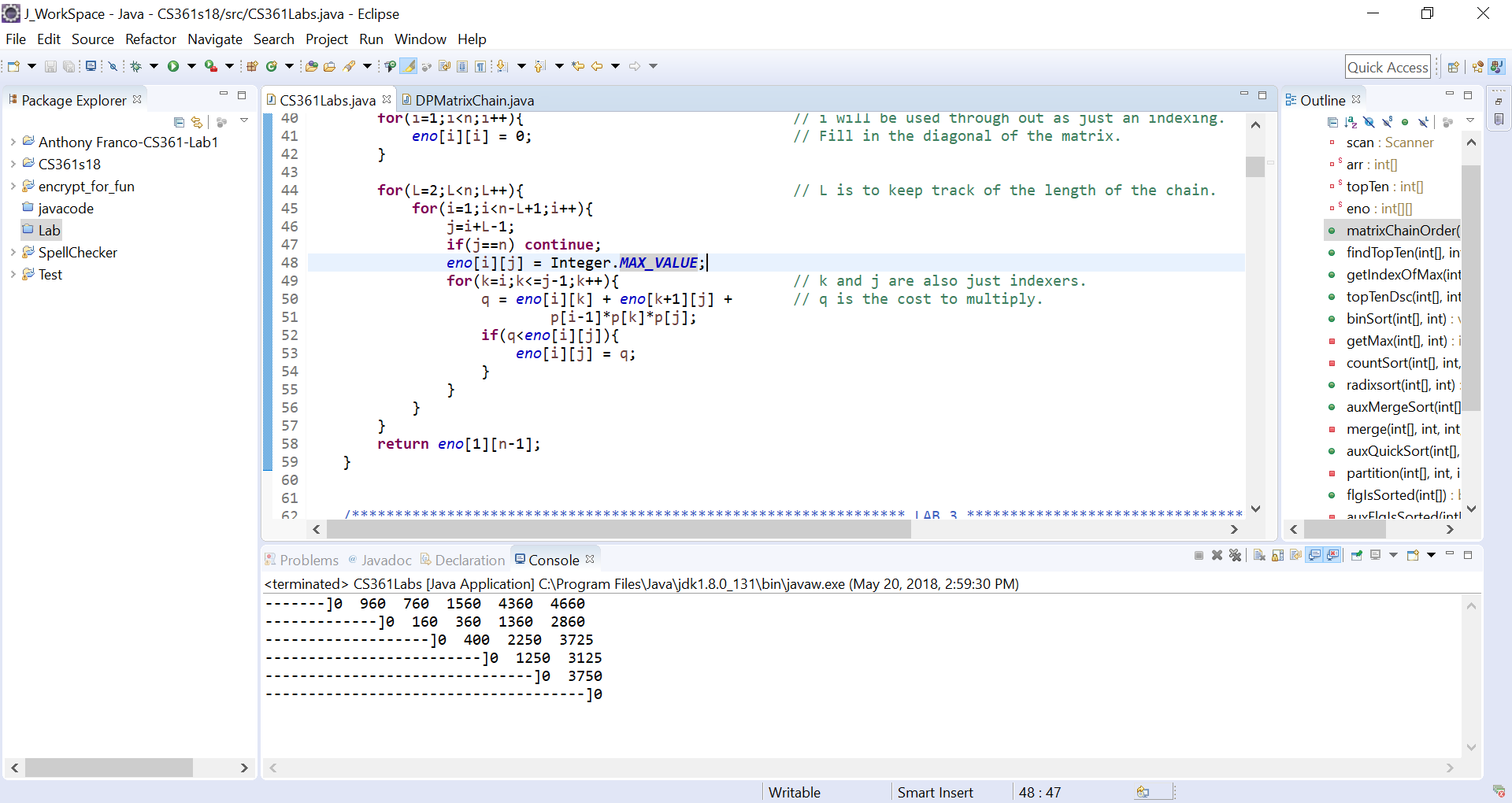


1. Implement the memoization version of MCM algorithm, using -1 for . Show commented code.

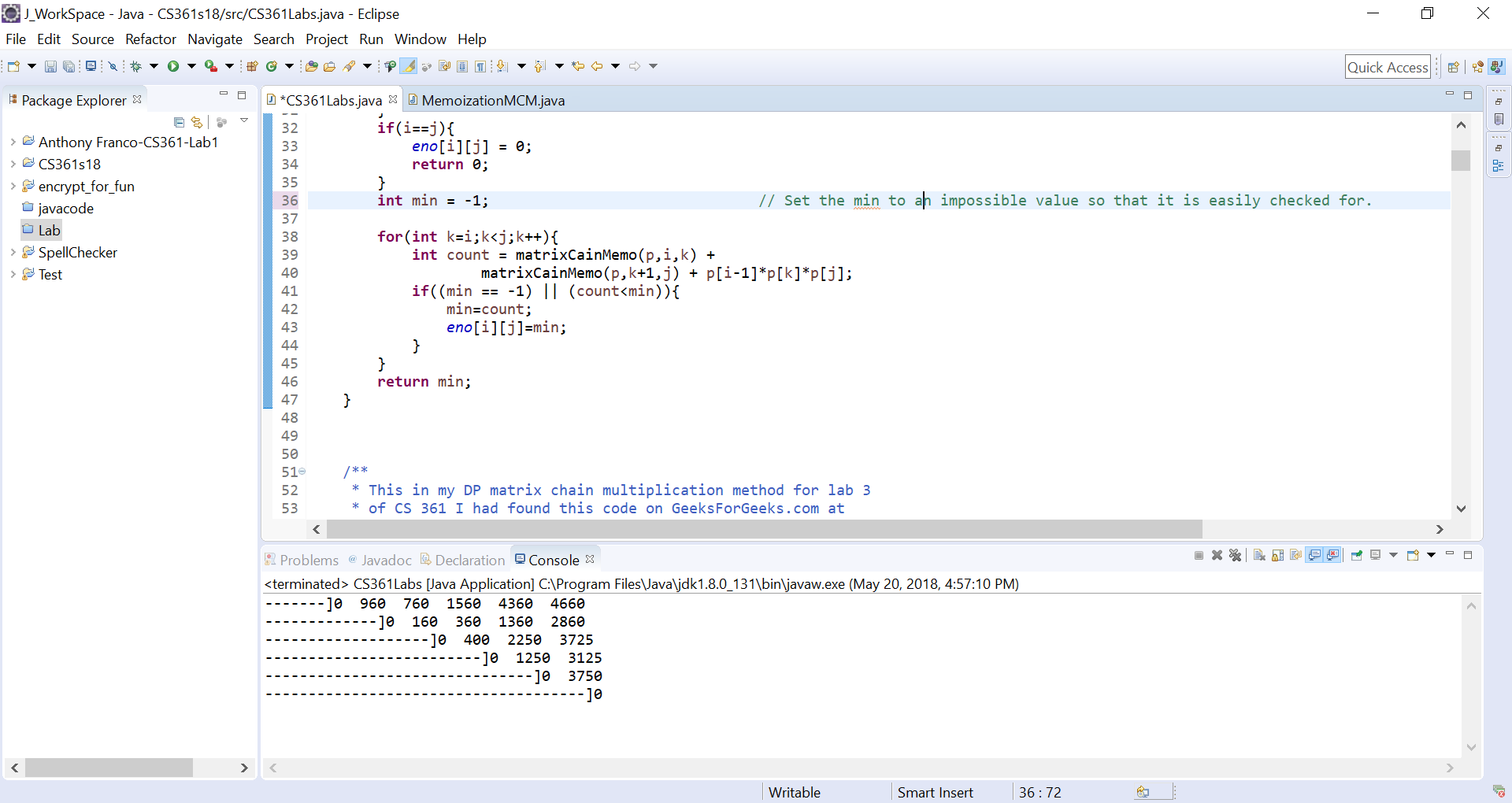




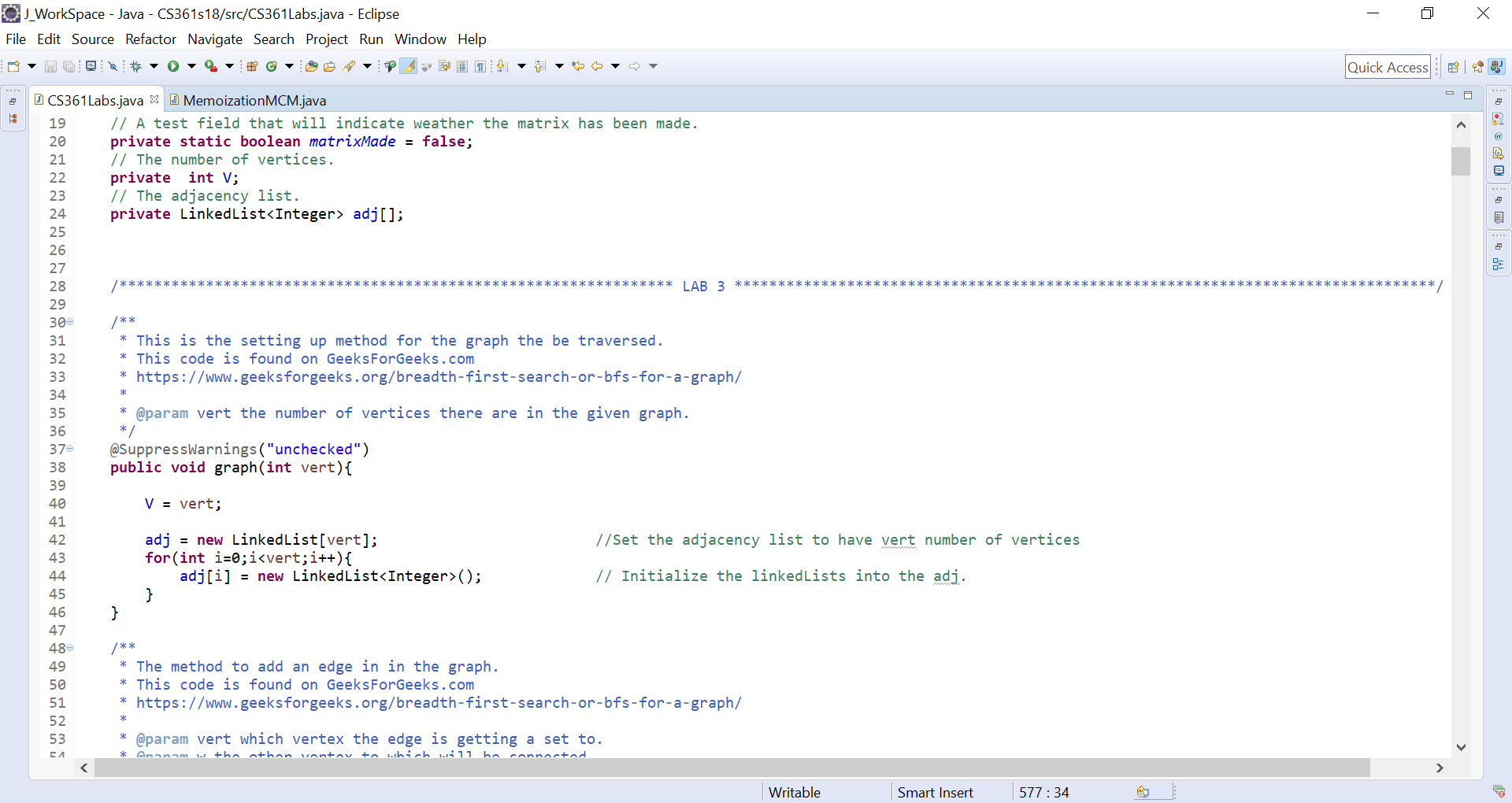
1. Show the output for your DP version of MCM algorithm for p being < 30, 4, 8, 5, 10, 25, 15>, including where the parenthesis should be located.

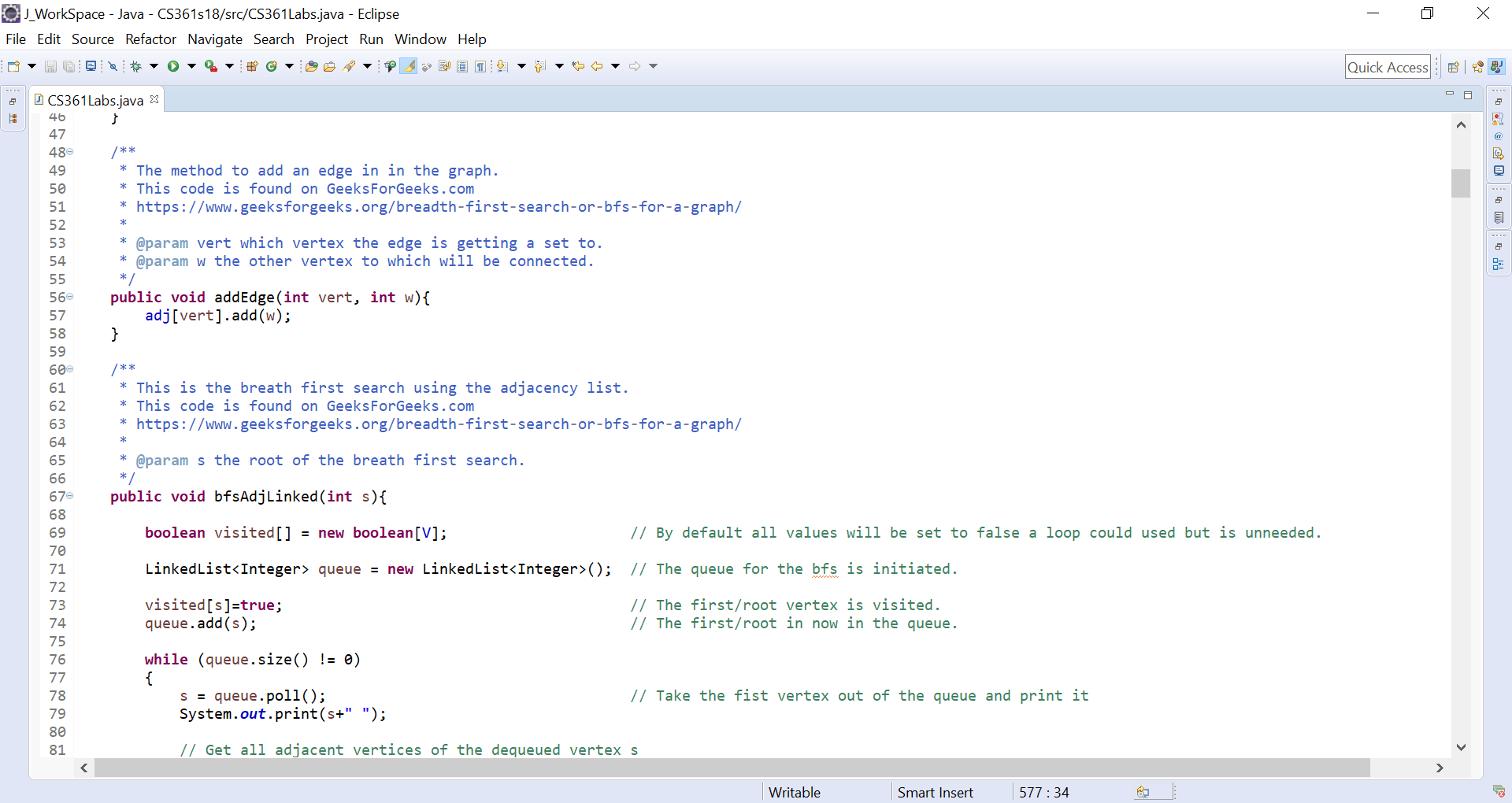


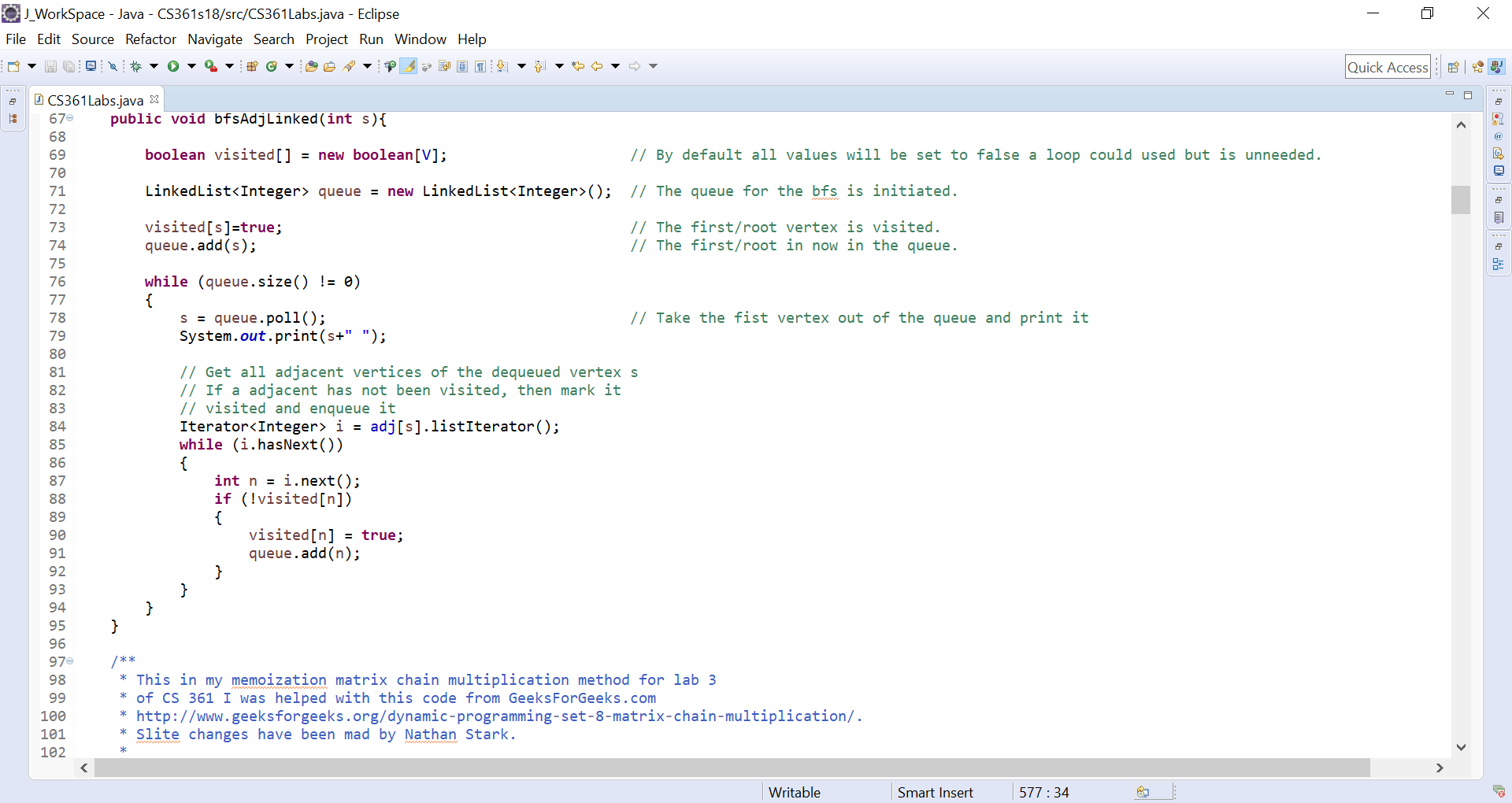
1. Show the output for your memoization version of MCM algorithm for p being < 30, 4, 8, 5, 10, 25, 15>, including where the parenthesis should be located.



1. Implement a breadth first search using an adjacency list. Show commented code.

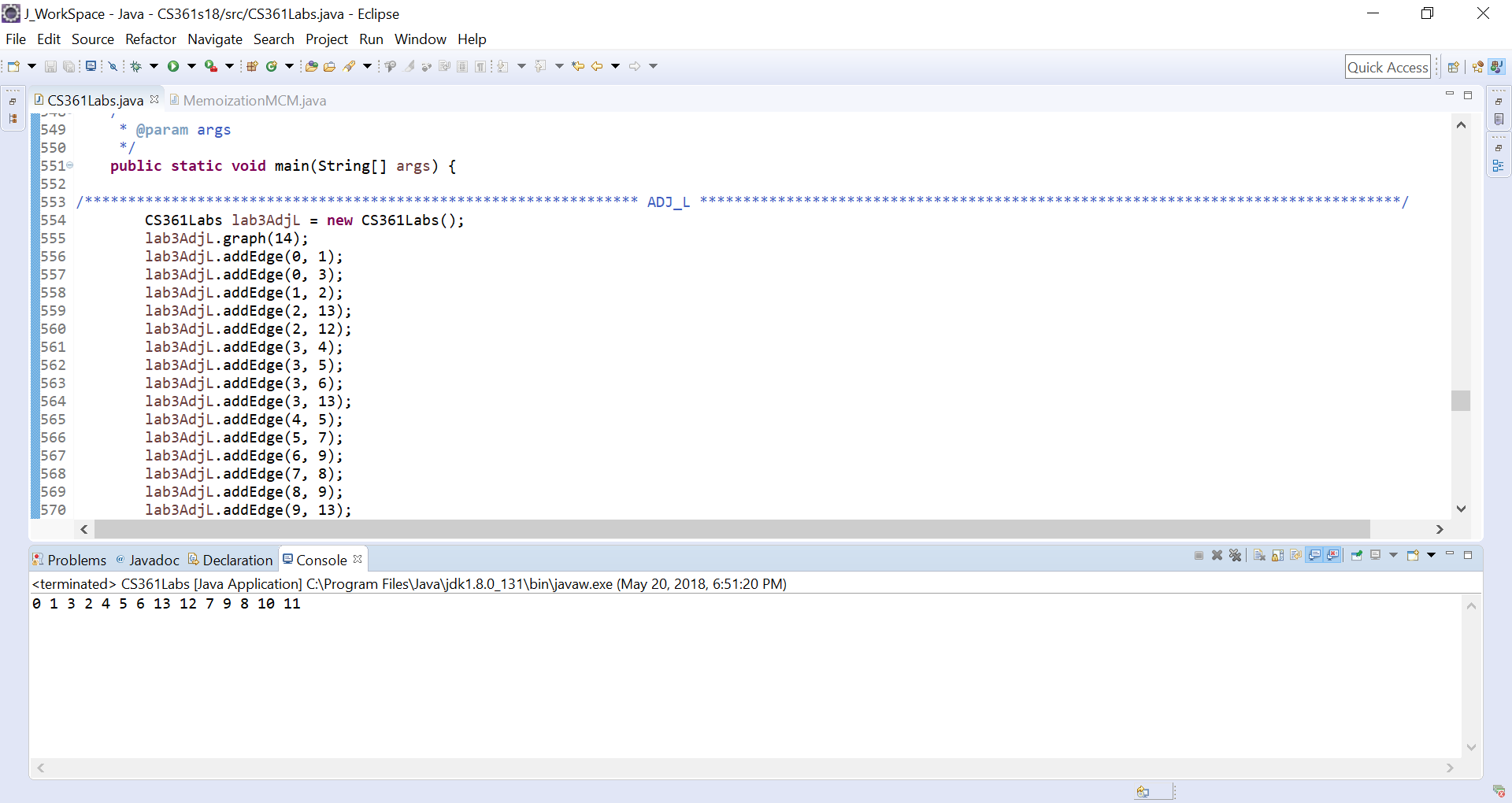






1. Implement a breadth first search using an adjacency matrix. Show commented code.
2. Show the output for the bfs using the adjacency list on the graph below.

**Let it be noted that I am converting the alphabetical letters to numeric values. Therefore, letting a=0 and so on. Thus, my results are as follows a, b, d, c, e, f, g, n, m, h, j, i, k, and l.**



1. Show the output for the bfs using the adjacency matrix on the graph below.

