Assignment 15
(Full Score: 200 points)

(**There are 3 deadlines for this Group Project**)

Your name:	Score:

Playing Chess with Java

Implement in Java a special Chess Game: there are only two pieces, one piece Knight and another one that is called Jack. Jack does not move and Knight moves in the same way as the knight does in a regular chess. The chessboard is the same as the regular chessboard except that its dimension does not have to be strictly **8 x 8** but can be ***d* x *d*** for any integer ***d*** that is greater than or equal to 8.

How to play: with initial setting of Jack and Knight's positions, Jack gives an integer number, ***s***, and challenges Knight to reach/attach Jack in ***s*** or fewer moves. If Knight can reach Jack in ***s*** or fewer moves, Knight wins; otherwise, Jack wins.

Now we use the computer and Java to simulate the game. For each round, the following are given: (1) chessboard dimension ***d***, (2) the number of moves given by Jack ***s***; (3) Knight's position ***Kx*** and ***Ky***; and Jack's position ***Jx*** and ***Jy***. (You may assume that chessboard is like a coordinate system and one pair of two farthest opposite squares have the following coordinates: (1, 1) and (*d*, *d*). Thus, 1 <=***Kx, Ky, Jx, Jy*** <= ***d***. You may assume that ***d*** <= 1,000,000, and ***s*** <= 256).

Your program must read 3 lines of numbers as input and produce one single line output saying "Jack wins" or "Knight wins" according to the following format:
Input: 3 lines:

- First line: ***d*** and ***s***;
- Second line: ***Kx*** and ***Ky*** (Kngiht's position);
- Third line: ***Jx*** and ***Jy*** (Jack's position).

(Numbers one the same line are separated with one white space).

Output: 1 line:

- "Jack wins." or "Knight wins."

Hints:

1. Using Java's standard *Queue, Set, TreeSet* API's.

Submissions (one submission for one team):

1. Report One: **due on 4/22/**Friday Midnight at Moodle; **10 points**.
    - One page report with names of your team (***at most 3 people on one team***).
2. Report Two: **due on 4/29/**Friday Midnight at Moodle; **40 points**.
    - A two-page report with additional cover page with team names on it;
    - Describe your algorithms that you will use to design/implement the game;
    - What datastructures/API you will use and short description of the purpose for each of the API/datastructure.
    - Pseudocode for your design and implementation.
3. Report Three: **due on 5/9/**Monday Midnight at Moodle; **150 points**.
    - An updated Report Two;
    - Java code for the project;
    - Javadoc file.
    - (Your code must follow **Java Code/Javadoc Convention** (20% of the total credits.)).