



# Simplified Chess Engine

by [pkacprzak](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

Chess is a very popular game played by hundreds of millions of people. Nowadays, we have chess engines such as [Stockfish](#) and [Komodo](#) to help us analyze games. These engines are very powerful pieces of well-developed software that use intelligent ideas and algorithms to analyze positions and sequences of moves, as well as find tactical ideas. Consider the following *simplified version of chess*:

- **Board:** It's played on a  $4 \times 4$  board between two players named *Black* and *White*.
- **Pieces and Movement:**
  - White initially has  $w$  pieces and Black initially has  $b$  pieces.
  - There are no Kings and no Pawns on the board. Each player has exactly *one* Queen, at most *two* Rooks, and at most *two* minor pieces (i.e., a Bishop and/or Knight).
  - Each piece's possible moves are the same as in classical chess, and each move made by any player counts as a single move.
  - There is no draw when positions are repeated as there is in classical chess.
- **Objective:** The goal of the game is to capture the opponent's Queen without losing your own.

Given  $m$  and the layout of pieces for  $g$  games of simplified chess, implement a very basic (in comparison to the real ones) engine for our simplified version of chess with the ability to determine whether or not White can win in  $\leq m$  moves (regardless of how Black plays) if White always moves first. For each game, print YES on a new line if White can win under the specified conditions; otherwise, print NO.

## Input Format

The first line contains a single integer,  $g$ , denoting the number of simplified chess games. The subsequent lines define each game in the following format:

- The first line contains three space-separated integers denoting the respective values of  $w$  (the number of White pieces),  $b$  (the number of Black pieces), and  $m$  (the maximum number of moves we want to know if White can win in).
- The  $w + b$  subsequent lines describe each chesspiece in the format  $t \ c \ r$ , where  $t$  is a character  $\in \{Q, N, B, R\}$  denoting the type of piece (where  $Q$  is Queen,  $N$  is Knight,  $B$  is Bishop, and  $R$  is Rook), and  $c$  and  $r$  denote the respective column and row on the board where the figure is placed (where  $c \in \{A, B, C, D\}$  and  $r \in \{1, 2, 3, 4\}$ ). These inputs are given as follows:
  - Each of the  $w$  subsequent lines denotes the type and location of a White piece on the board.
  - Each of the  $b$  subsequent lines denotes the type and location of a Black piece on the board.

## Constraints

- It is guaranteed that the locations of all pieces given as input are distinct.
- $1 \leq g \leq 200$
- $1 \leq w, b \leq 5$
- $1 \leq m \leq 6$
- Each player initially has exactly one Queen, at most two Rooks and at most two minor pieces.

## Output Format

For each of the  $g$  games of simplified chess, print whether or not White can win in  $\leq m$  moves on a new line. If it's possible, print YES ; otherwise, print NO .

### Sample Input 0

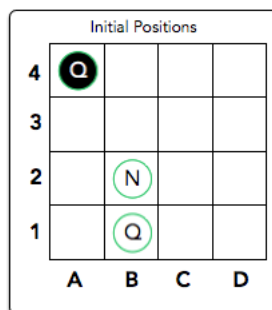
```
1
2 1 1
N B 2
Q B 1
Q A 4
```

### Sample Output 0

```
YES
```

### Explanation 0

We play  $g = 1$  games of simplified chess, where the initial piece layout is as follows:



White is the next to move, and they can win the game in 1 move by taking their Knight to A4 and capturing Black's Queen. Because it took 1 move to win and  $1 \leq m$ , we print YES on a new line.

f t in

Submissions: 560

Max Score: 40

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable) 🔗 ↺

Java 7



```
1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8
9     public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }
```

Line: 1 Col: 1

 [Upload Code as File](#)☐ Test against custom input[Run Code](#)[Submit Code](#)

---

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)