**23A. SuperChess**

# Program Name: Super.java Input File: super.dat

SuperChess is a free version of chess you pirated off some Russian site, and as it turns out, it’s more than just free in cost- it’s also free of other pieces on the board. Given the name of the piece you are controlling, a starting point and a target, calculate the minimum number of moves to get from the starting point to a target. The pieces are:

King (K) can move one step vertically, horizontally, or diagonally in any direction per turn.

Queen (Q) can move an infinite number of steps in a straight line, vertically, horizontally, or diagonally each turn.

Rook (R) can move an infinite number of steps in a straight line, vertically or horizontally each turn.

Bishop (B) can move an infinite number of steps in a straight line, diagonally each turn.

Knight (N) can move 2 steps in any direction and another 1 step in a perpendicular direction each turn, for a total distance of sqrt(5).

**Input**

The first line of input will contain a single integer n that indicates the number of boards. The first line of each board will contain an integer m, the number of moves to consider for this test case. The next m lines will contain a single letter that represents a piece listed above (ignore pawns), the coordinates of the starting point x y where a matrix of the board mat can access the point as m[x][y], and the coordinates of the destination i j where a matrix of the board mat can access the point as m[i][j]. 0 <= x, y, i, j <= 8.

**Output**

For each test case on each board, print out the minimum number of moves required to move the piece from the start to the destination. Separate each board with ----- (5 dashes).

**Example Input File**

2

5

R 2 3 5 6

B 0 0 7 7

K 1 2 5 7

Q 1 2 5 7

N 0 0 4 2

2

N 0 0 0 1

N 0 0 0 2

**Example Output to Screen**

2

1

5

2

2

-----

3

2

-----