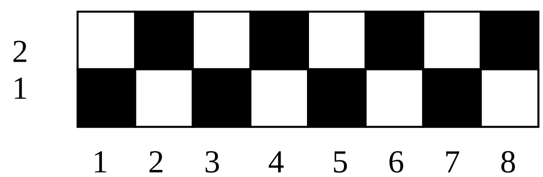
2001 - 2002 | **AMERICAN COMPUTER SCIENCE LEAGUE** | **Contest #2**

Junior Division Programming Problem

CHECKERS

PROBLEM: The game of Checkers is played on an 8 x 8 grid. The checkers are placed on the black squares only. Checkers move towards the other end of the grid by moving to an unoccupied, adjacent black square. That is a checker at location (1,1) would move to location (2,2) if it were unoccupied. Checkers may “jump” over and capture an opponent’s checker if that checker is in an adjacent black square and the landing black square is unoccupied. That is a checker at location (1,1) could ‘jump” over an opponent’s checker at location (2,2) if location (3,3) was unoccupied. For this game your “home row” is row 1 and your opponent’s home row is row 8. If your checker lands in your opponent’s home row, the checker becomes a “king” and can move both forwards and backwards from adjacent black square to adjacent black square.



For this program we are restricting the movement of your checker so that it can only move from row 1 to row 8. Further, if there is a jump to be made from a location, then just one jump will be possible.

INPUT: In this problem, you will be given the location of one of your checkers, followed by the number of your opponent’s checkers and their locations. Locations will be given in ordered pair format (row, column). Sample Input line #1 below indicates that your checker is at location (1,5) and your opponent has 3 checkers at locations (2,6), (4,6) and (6,6).

OUTPUT: For each input, it is your turn to move a checker. Print the maximum number of legal jumps that can be made by your checker. Also, if your checker finishes in the “home row” of your opponent, then print the word “king” next to your answer for the number of jumps.

SAMPLE INPUT	**SAMPLE OUTPUT**
1. 1, 5, 3, 2, 6, 4, 6, 6, 6	3
2. 2, 2, 4, 3, 3, 5, 3, 7, 3, 7, 5	3, KING

2001 **AMERICAN COMPUTER SCIENCE LEAGUE** CONTEST #2

JUNIOR DIVISION PROGRAMMING PROBLEM

TEST DATA

TEST INPUT	TEST OUTPUT
1. 1,7,2,2,6,4,4	1. 2
2. 2,2,3,3,3,5,5,7,7	2. 3, KING
3. 1,5,4,2,4,2,6,3,7,3,3	3. 0
4. 1,3,5,2,4,3,5,2,2,4,2,5,3	4. 1
5. 2,2,5,3,3,5,5,7,7,3,1,4,2	5. 3, KING