Problem 1 - Dangerous Floor

You have just entered the door at the final address. In the end of the room there is a safe which holds the treasure you are searching for. The floor between you and the safe though is dangerous! If you do a wrong move it will collapse and you will never reach the treasure …or anything …ever. Luckily you have gathered all the pieces of instructions left from this uncle of yours at each address you have visited before.

Imagine the floor in the form of a chess board. On different squares of this "board" are placed some chess pieces. If you follow all the instructions correctly (where possible) you will be able to pass safely on the other side.

The board is **8x8 squares**. There are **6 kinds** of **pieces**:

- **K**ing - moves exactly one square horizontally, vertically, or diagonally.
- **R**ook- moves any number of vacant squares in a horizontal or vertical direction.
- **B**ishop - moves any number of vacant squares in any diagonal direction.
- **Q**ueen - moves any number of vacant squares in a horizontal, vertical, or diagonal direction.
- **P**awn – moves straight forward **one square to the top.**

Yeah, we know, that there are some more moves and **figures**, but for now these are enough. You can check the visual representation of the moves here.

Input

On the **first 8 rows**, you will receive a board with **some pieces** placed on it. **Empty cells** will be marked with "**x**" and all squares will be separated by comma.

{x,x,Q,x,R,x,x,P}
{x,B,x,x,x,K,x,P}

There are 7 symbols all in all:

- **K** – King
- **R** – Rook
- **B** – Bishop
- **Q** – Queen
- **P** – Pawn
- **x** – Empty cell

On the **next lines**, you will receive **moves** that **need to be checked** and **if** they are **valid** you should **move** the **piece** to its **new position**. There are **3** kinds of **problems** that may occur and you need to be looking for. The **check must happen** in the **very same order** as shown **below**:

1. There is no such a piece on starting square.
2. Piece makes invalid move (look above).
3. Piece gets out of board.

So, if you find that there is no such a piece, you don't need to check for the other problems. **If** you **find any problem** then you must **cancel** the **move** and the **board stays the same** for the next move.

{Q01-12}

The **first symbol** is the **type of piece** you need to move. Then you will read **two digits** which sate the **position** on which the **piece should be** currently (row, col). The **next two digits** are the **final position** (row, col) where you should **try to move** the **piece** to.

Read moves till you find the keyword "**END**".

Output

You should **print** lines on the console **only when** the **move** is **not possible**. You can have three different types of output, depending on what kind of problem you hit.

1. **There is no such a piece!**
2. **Invalid move!**
3. **Move go out of board!**

You can print only **one message per** invalid **input move**, so **check** for the problems **in the order** shown **above**. The **check must happen** in the **very same order.**

Constraints

- Pieces will never go to a cell with another piece
- **Moves count** will be **in** the **range** [0…1000]
- Time limit: 0.3 sec. Memory limit: 16 MB.

Examples

Input	Output
x,x,x,x,x,x,x,x x,K,x,x,x,x,x,x x,x,x,x,x,x,x,x x,x,x,x,x,x,x,x x,x,x,x,x,x,x,x x,x,x,x,x,x,x,x x,x,x,x,x,P,x,x x,x,x,x,x,x,x,x K33-44 K11-02 P44-34 P65-55 P55-65 END	There is no such a piece! There is no such a piece! Invalid move!
x,x,x,x,x,x,x,x x,Q,x,x,x,x,x,x x,x,x,x,x,x,x,R x,x,x,x,x,x,x,x x,x,x,x,x,x,x,x x,x,x,B,x,x,x,x x,x,x,x,x,x,x,x x,x,x,x,x,x,x,x Q11-15 Q15-24 R27-36 R27-28 B53-55 Q24-26 R27-07 R27-07 END	Invalid move! Move go out of board! Invalid move! There is no such a piece!