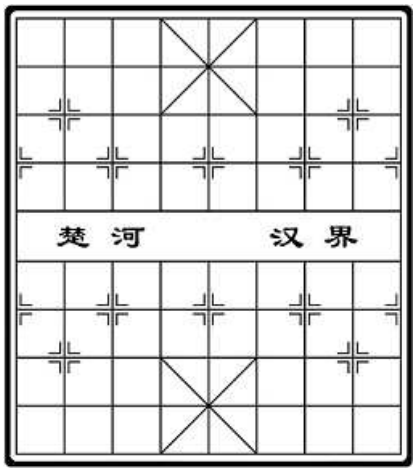
Problem C. Chinese Chess

Time limit: 1 second
Memory limit: 1024 megabytes

This is an interactive problem.

Chinese chess is a two-player strategy board game that is very popular in China. There are 10 rows and 9 columns on a Chinese chess board. We denote the point on the r-th row from bottom to top and on the c-th column from left to right by (r, c) ($0 \le r \le 9$, $0 \le c \le 8$).



There are six types of pieces in this problem: kings, mandarins, rooks, knights, bishops, and pawns. Note that cannons are not considered. Additionally, all pieces can be moved anywhere on the board, which differs slightly from the original rules. The movement rules for each piece are as follows (assuming the current position is (r, c)):

Chess	Symbol	Possible positions in one move
King	J	$(r+1, c)$ $(r-1, c)$ $(r, c+1)$ $(r, c-1)$
Mandarin	S	$(r+1, c+1)$ $(r+1, c-1)$ $(r-1, c+1)$ $(r-1, c-1)$
Rook	C	all (r', c') such that $r' = r$ or $c' = c$ except (r, c) itself
Knight	M	$(r+2, c+1)$ $(r+2, c-1)$ $(r-2, c+1)$ $(r-2, c-1)$ $(r+1, c+2)$ $(r+1, c-2)$ $(r-1, c+2)$ $(r-1, c-2)$
Bishop	X	$(r+2, c+2)$ $(r+2, c-2)$ $(r-2, c+2)$ $(r-2, c-2)$
Pawn	B	$(r+1, c)$ if $r \le 4$, otherwise $(r+1, c)$ $(r, c+1)$ $(r, c-1)$

Of course, pieces cannot move outside the board's boundaries. Now that you understand the rules, let's play a game.

I have placed a piece on the board, and you cannot see its position or type. However, I will provide a set A of n possible positions where the piece might be located. Your goal is to determine the type of the piece using the minimum number of queries. For each query, you can select a position on the board, and I will tell you the minimum number of moves required for the piece to move to that position, or I will tell you if it is impossible for the piece to reach that position.

Interesting, isn't it? What is even more interesting is that I am actually gaming against you in secret. Truth be told, although set A is fixed, I have not fixed the type and position of this piece from the start. My strategy is to adjust the answer to each of your queries in a way that maximizes the number of queries you make, without contradicting the information already given. In the case where both of our strategies are optimal, your number of queries m is actually determinable. I am sure you are smart enough to figure it out. Let the game begin!

Interaction Protocol

Firstly, you should read an integer n ($1 \le n \le 90$) from standard input, indicating the size of the set A.

Then, read n lines. Each line contains two integers r and c, indicating a position in the set A. All positions are guaranteed to be distinct.

After reading these $n+1$ lines, you should output an integer m to standard output, indicating the number of queries. If your output m is incorrect, you will get a `Wrong answer` verdict. You must then make exactly m queries.

To make a query, output `? r c` ($0 \le r \le 9, 0 \le c \le 8$) in a single line. Then you should read an integer from standard input, indicating the minimum number of moves required for the piece to reach the selected position, or -1 if the piece cannot move to that position. If your program makes an invalid query or exceeds the number of queries, the interactor will terminate immediately, and you will receive a `Wrong answer` verdict.

To output your guess about the type of the piece, you need to output `! ch`, where `ch` means the symbol representing the piece that is shown in the statement.

After you make a query or output your answer, including the m and the type of the piece, do not forget to output the End of Line `\n` and flush the output. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;

Examples

standard input	standard output
1 9 0 6	 1 ? 3 6 ! S
4 2 1 2 3 2 5 2 7 3 1	 2 ? 0 0 ? 2 0 ! J

Note

For the first example, the number of moves required for each of the six types of pieces to move from position $(9, 0)$ to $(3, 6)$ is as follows:

King (J)	Mandarins (S)	Rooks (C)	Knights (M)	Bishops (X)	Pawns (B)
12	6	2	4	3	-1

Thus, you only need to make one query to determine the answer.