Fischer’s Chess Guessing Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

This is an interactive problem.

Fischer’s Chess is a game in which the starting position is randomly chosen from 960 positions. The starting position satisfies the following requirements:

- The white pieces are shuffled (the king, queen, two rooks, two bishops, and two knights are placed on the first rank).
- The black pieces are placed on the eighth rank in the same order as the white pieces.
- The king is placed between the two rooks (there may be other pieces between the king and rooks). This is necessary for castling rule.
- The white bishops are placed on opposite-colored squares. In other words, one bishop is on a square of even file, and the other is on a square of odd file.

Jill has chosen one of the starting positions. Jack’s task is to guess it. Jack places any legal starting position on the chessboard, and Jill tells him how many of the white pieces he has placed correctly. Your task is to help Jack guess the position by placing no more than six positions.

Interaction Protocol

The interaction consists of no more than 100 games. At the beginning of each game, you need to read a line from the input containing the word “`GAME`” and an integer n corresponding to the game number (starting from 1).

When attempting to guess the position, output a string of 8 characters representing the arrangement of the white pieces. The king is denoted by character “`K`”, the queen by “`Q`”, the rook by “`R`”, the bishop by “`B`”, and the knight by “`N`”. In response, read a line from the input containing an integer from 0 to 8: the number of correctly placed white pieces. The game ends when you read number 8 (all pieces are placed correctly) or the limit on the number of guesses is exceeded (in this case, the outcome will be “`Wrong Answer`” if the solution terminates immediately, or it can be any non-OK if the solution tries to read more input).

At the end of the interaction, read a line from the input containing the word “`END`”.

After outputting each position, don’t forget to print the newline character and flush the correct output buffer, or the outcome will be “`Idleness Limit Exceeded`”. To flush the buffer, one can call, for example, `fflush (stdout)` in C or C++, `System.out.flush ()` in Java, `flush (output)` in Pascal or `sys.stdout.flush ()` in Python.

In each game, the permutation is fixed in advance and does not change.

Example

standard input	*standard output*
GAME 1	
	RQKBBNRN
4	
	RQKNBRNB
3	
	RKRNBBQN
5	
	RKRBBQNN
8	
END	