American Computer Science League

2021 Finals ● King in Danger ● Intermediate & Senior Divisions

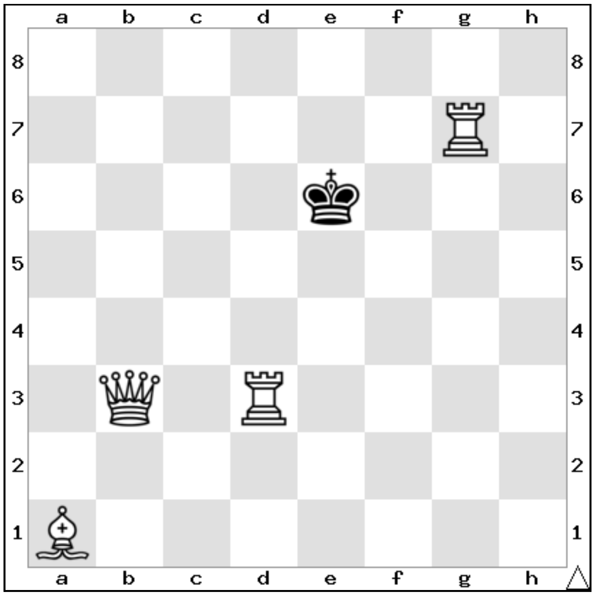
PROBLEM: Given the state of a game of chess, determine whether the black king is *safe*, in *check*, in *checkmate*, or in *stalemate*. The king will be the only black piece on the board. Its position, as well as all of the opponent's pieces, will be specified in the input.

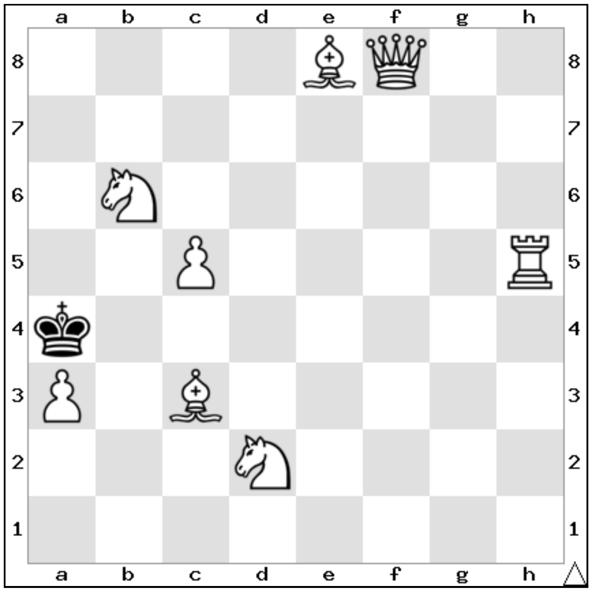
A king is said to be in *check* if it is on a square that can be captured by an opponent. The king can move one space in any direction. If there is an opponent in the destination square, the king captures that piece and that piece is removed from the board; however, this move would be illegal if it results in the king's new position being in check.

There are four situations that a king can be in, and in this program, you need to identify each situation:

Safe	The king is not in check and can move to a location where he would not be in check.
Check	The king is in check but can move to a location where he would not be in check.
Checkmate	The king is in check and every move he could make would put him in check.
Stalemate	The king is not in check and every move he could make would put him in check.

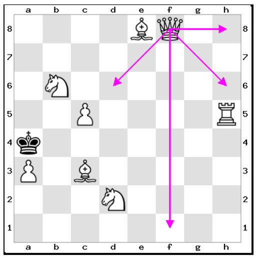
Here are two sample boards. The left board corresponds to the first line of Sample Input; the right board corresponds to the last line of Sample Input.



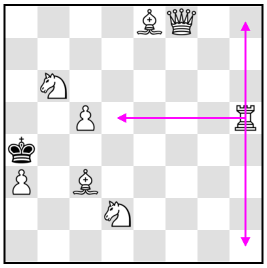


The chess pieces are as follows and can move as follows:

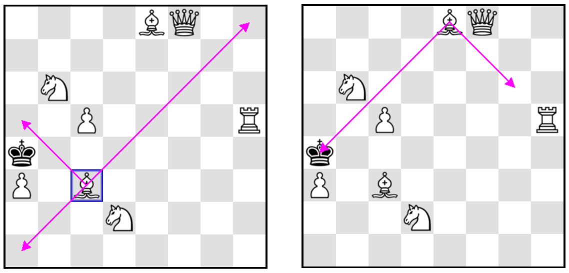
Queen (Q) The queen can move horizontally, vertically, or diagonally in any direction. If it encounters a piece of its color, it may not move there or any further. If it encounters a piece of the other color, it captures that piece on the board and assumes that position.



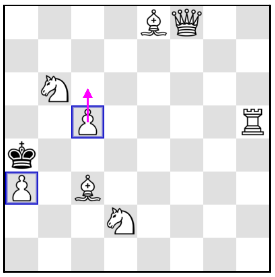
Rook (R) A rook can only move horizontally or vertically in any direction. If it encounters a piece of the other color, it captures that piece on the board and assumes that position.



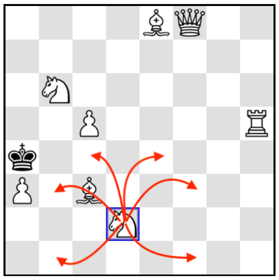
Bishop (B) A bishop can only move diagonally in any direction. If it encounters a piece of the other color, it captures that piece on the board and assumes that position.



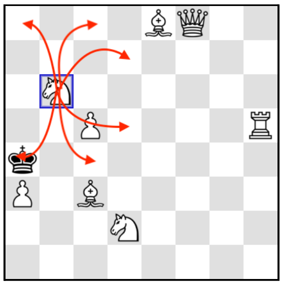
Pawn (P) The pawn can only move one square forward from row 2 toward row 8 if the square is unoccupied; it can capture an enemy piece by moving one square diagonally forward. For example, in the diagram at the right, the pawn at c5 could move to c6. If an enemy piece were on d6, it could move there and capture that piece. The pawn on a3 cannot move to a4 because that position is occupied by a piece. Were an enemy piece on b4, the pawn could move there and capture that enemy piece.



Knight (N) The knight can move 2 spaces horizontally and 1 space vertically, or 1 space horizontally and 2 spaces vertically in any direction and may jump over another piece. If the destination square is occupied by the enemy, it captures that enemy piece; if the square is occupied by a piece of its own color, it may not move there.



In the diagram at the right (top), the knight at d2 can move to b1, b3, c4, e4, f3, or f1. In the diagram at the right (bottom), the knight at b6 can move to a8, c8, d7, d5, c4, or a4. Were it to move to a4, it would capture the black king that is there.



As in the actual game of chess, there will be at most 2 rooks, 2 bishops, 2 knights, 8 pawns, and 1 queen among the white pieces. Our boards will not have a white king. The king will be the lone black piece. There will be 5 Sample Inputs and 10 Test Inputs, each representing a current state of a chessboard as described above. The first 6 Test Inputs will not include pawns or knights; inputs 7 and 8 may include pawns; inputs 9 and 10 may include knights and pawns.

For example, in both of the two sample boards above, the black king is SAFE. For the board on the left, the king is in check by the queen at position b3, but can safely move out of being in check by moving to position f5. For the board on the right, the king is in check by the knight at position b6, but can safely move out of being in check by moving to position a3.

INPUT: A string representing the location of pieces on the chessboard. Each piece is a 3-character string: the piece (RNBQPK), the column (a through h), and the row (1 through 8). All pieces are white, other than the king. The king is the only black piece. The pieces are separated by a single space.

OUTPUT: Output one of the following in all capital letters: SAFE, CHECK, CHECKMATE, or STALEMATE, depending on the king's status.

SAMPLE INPUT:

```
Ba1 Rd3 Rg7 Qb3 Ke6
Rc1 Kd8 Qb6 Re5 Bh3
Qf4 Be5 Rc1 Kd3
Ra1 Pb5 Pc5 Rc3 Bd6 Qg4 Kb7
Ka4 Be8 Rh5 Qf8 Nb6 Nd2 Pc5 Pa3 Bc3
```

SAMPLE OUTPUT:

```
CHECK
CHECKMATE
SAFE
STALEMATE
CHECK
```

TEST DATA

TEST INPUT:

```
Rh8 Qh2 Be8 Rg1 Kh4
Ba1 Rf6 Rg2 Qc1 Ke3
Rc8 Qc3 Re4 Bh5 Kf7
Kd6 Qf5 Be4 Rc7 Re1 Ba5
Rb8 Ka3 Qe5 Bd1 Bc3 Rh3
Rh8 Qa8 Bh2 Kc7 Bd7 Rb2

Bf8 Rb8 Pc5 Qh7 Kc6 Bf1 Pb5
Ba2 Pg5 Re3 Qd8 Pd5 Kf7 Ph6

Kb8 Qb7 Nb5 Nd6
Ra1 Ka8 Nc8 Nd7 Bg1 Qe8 Pa6 Rh8
```

TEST OUTPUT:

```
  CHECKMATE
  CHECK
  CHECKMATE
  STALEMATE
  SAFE
  CHECK
  CHECK
  SAFE
  CHECKMATE
  STALEMATE
```

题目：观察棋局，判断黑色国王的状态是*安全*、被*将军*、被*将死*还是陷入*僵局*。国王是棋盘上唯一的黑色棋子。它和对方所有棋子的位置将会在输入中详细说明。

如果国王位于可以被对方棋子吃掉的格子中，则认为国王被*将军*。国王可以在任意方向走一步。如果国王走到对方棋子所在格子中，则吃掉该棋子，被吃的棋子从棋盘上移除；但是，如果国王走到的格子会再次被将军，则这一步算作犯规。

国王在棋盘上可以有四种状态，在这一程序中，你需要识别每种状态：

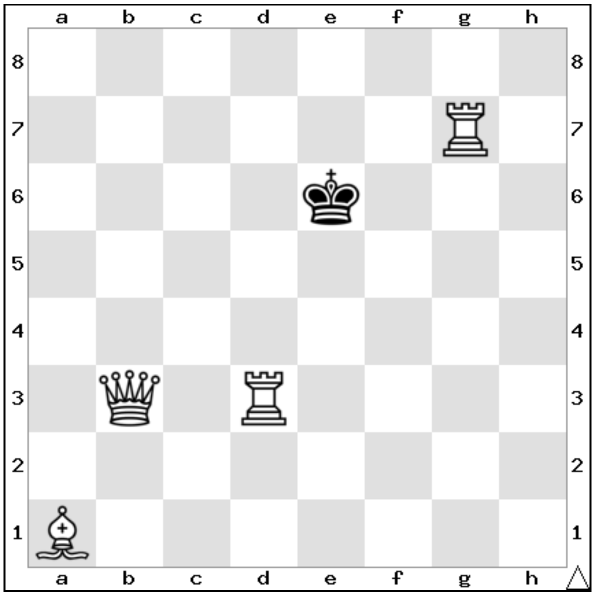
安全　　国王没有被将军且可以走到一个不会被将军的位置。

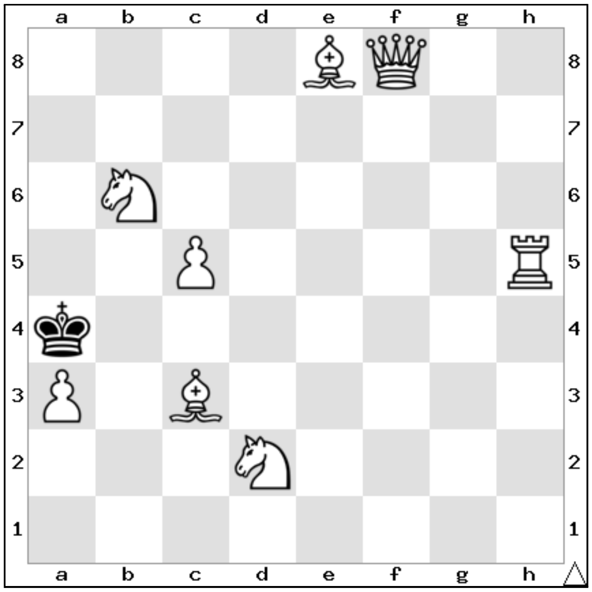
将军　　国王被将军，但是可以走到一个不会被将军的位置。

将死　　国王被将军且走到任何位置都会被将军。

僵局　　国王没有被将军，但是走到任何位置都会被将军。

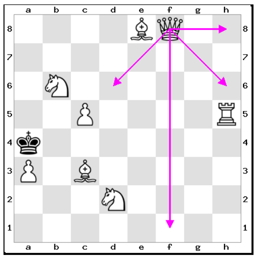
下方有两个棋盘示例。左侧棋盘对应样本输入的第一行；右侧棋盘对应样本输入的最后一行。





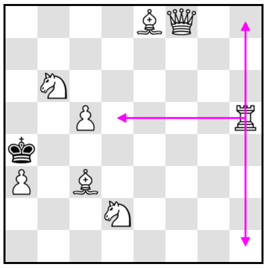
国际象棋中的棋子和行棋规则如下：

王后 (Q)　　王后可以横向、纵向或斜向走。如果遇到同色的棋子，它便不能走到该格或继续走。 如果遇到不同色的棋子，它可以吃掉这一棋子，并占据该格。



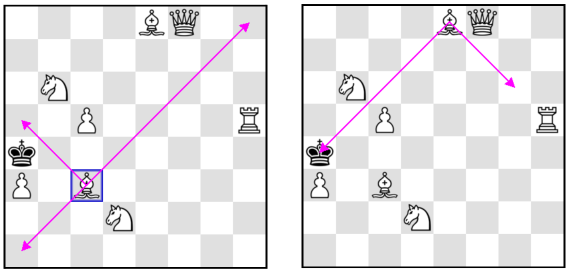
车 (R)

车可以横向或纵向走。如果遇到不同色的棋子，它可以吃掉这一棋子，并占据该格。



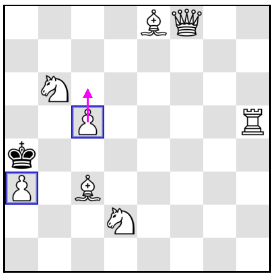
象 (B)

象只能斜走。如果遇到不同色的棋子，它可以吃掉这一棋子，并占据该格。



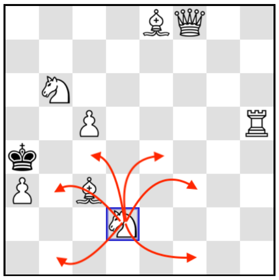
兵 (P)

如果格子没有被占据，兵只能从第2行向第8行走一步；它斜向前走一步可以吃掉一个对方棋子。例如，在右图棋盘中，c5格子上的兵可以走到c6。如果一个对方棋子在d6格子上，兵可以走到那里，并吃掉对方棋子。a3格子上的兵不能走到a4格子上，因为那个格子上有棋子占据。如果 b4格子上有一个对方棋子，兵可以走到那里，并吃掉对方棋子。

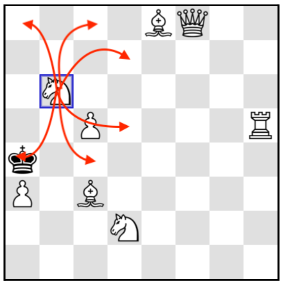


马 (N)

马可以先横向走两步再纵向走一步，或者先横向走一步再纵向走两步，可以越子。如果所到的格子上有对方棋子，它可以吃掉对方棋子；如果所到格子上 有己方棋子，则不可以走到该格。



在右上方的棋盘中，d2格子上的马可以走到b1, b3, c4, e4, f3或f1格子上。在右下方的棋盘中，b6格子上的马可以走到 a8, c8, d7,d5, c4或a4格子上。如果走到a4格子上，它就会吃掉黑色的国王。



由于在真实的国际象棋中，白棋最多有 2 个车，2 个象，2 个马，8 个兵和 1 个王后。我们的棋盘上不会出现白色的国王，只有一个黑色的国王。将会有 5 行样本输入（已提供）和 10 行测试输入，每行输入代表一个上述棋盘的当前状态。要求前 6 行测试输入没有兵和马；第 7 和第 8 行有兵；第 9 和第 10 行有马和兵。

例如，在上方的两个棋盘示例中，黑色国王都是安全的。在左侧棋盘中，国王被 b3 格子上的王后将军，但是走到 f5 格子上就不会再被将军。在右侧棋盘中，国王被 b6 格子上的马将军，但是走到 a3 格子上就不会再被将军。

输入：表示棋盘上棋子位置的字符串。每个棋子表示为一个由 3 个字符组成的字符串：棋子 (RNBQPK 表示)，纵向位置 (a 到 h 表示) 以及横向位置 (1 到 8 表示)。除了国王，所有棋子都是白色。国王是唯一的黑色棋子。棋子之间由一个空格分隔。

输出：以大写字母的形式，输出国王所处的状态：SAFE (安全), CHECK (将军), CHECKMATE (将死) 或 STALEMATE (僵局) 。

样本输入：

```
Ba1 Rd3 Rg7 Qb3 Ke6
Rc1 Kd8 Qb6 Re5 Bh3
Qf4 Be5 Rc1 Kd3
Ra1 Pb5 Pc5 Rc3 Bd6 Qg4 Kb7
Ka4 Be8 Rh5 Qf8 Nb6 Nd2 Pc5 Pa3 Bc3
```

样本输出：

```
CHECK
CHECKMATE
SAFE
STALEMATE
CHECK
```

测试数据

测试输入：

```
Rh8 Qh2 Be8 Rg1 Kh4
Ba1 Rf6 Rg2 Qc1 Ke3
Rc8 Qc3 Re4 Bh5 Kf7
Kd6 Qf5 Be4 Rc7 Re1 Ba5
Rb8 Ka3 Qe5 Bd1 Bc3 Rh3
Rh8 Qa8 Bh2 Kc7 Bd7 Rb2

Bf8 Rb8 Pc5 Qh7 Kc6 Bf1 Pb5
Ba2 Pg5 Re3 Qd8 Pd5 Kf7 Ph6

Kb8 Qb7 Nb5 Nd6
Ra1 Ka8 Nc8 Nd7 Bg1 Qe8 Pa6 Rh8
```

测试输出：

```
CHECKMATE
CHECK
CHECKMATE
STALEMATE
SAFE
CHECK
CHECK
SAFE
CHECKMATE
STALEMATE
```