

Arimaa is a strategic two player game that takes place on a regular 8 * 8 board. Each player has 16 pawns which he plays on the board. They are 1 elephant, 1 camel, 2 horses, 2 dogs, 2 cats and 8 rabbits.

The objective of the game is to move the rabbit from one's own color to the home rank of the opponent.

The game begins with an empty 8 * 8 board. Player 1 places his 16 pawns in any order on the 0th and the 1st row of the board. Looking at the position of the pawns of the player 1, player 2 places his pawns on 6th and the 7th row of the board. This counters the advantage of the player 1 going first.

After the pawns are placed on the board, players alternate with player 1 going first. A turn consists of a player making 1 to 4 moves with each move consisting of a pawn occupying an empty square 1 step left, right, forward or backward. Rabbits are not allowed to move backward.

It is to be noted that a turn must change the state of the board, failing which the player is disqualified and the opponent wins.

A player can use 2 moves of his turn to push/pull an opponent. This can only be done on weaker opponents. The strength of various pawns are as follows.

Elephant > Camel > Horse > Dog > Cat > Rabbit

While pushing, the pawn occupies the square of the pushed opponent's pawn and the opponent in turn occupies any of its orthogonally adjacent squares.

While pulling, the opponent's pawn occupies the square of the pulling pawn and the pawn in turn occupies any of its orthogonally adjacent squares.

Pawns of the same player cannot be pushed/pulled and a pawn cannot pull/push simultaneously. Also, it can be seen that an Elephant cannot be pulled/pushed by any opponent.

A pawn which is adjacent to a stronger opponent pawn is considered frozen. It can be unfrozen by making a friendly pawn stand adjacent to it or by removing the adjacent strong opponent pawn.

Pawns can be taken out of the board by moving them into one of the 4 trap squares at (2,2), (2,5), (5,2) and (5,5). An unprotected pawn (which is not adjacent to any friendly piece) is automatically removed from the trap square.

The objective of the game is to either, remove all opponent's rabbits, place one's rabbit in the opponent's home row or block moves of all existing pawns.

Input Format

First line contains an integer 1/2 indicating whether its the first or the second player.

8 lines follow each line containing 8 characters.

- Empty square is represented by `.`
- Elephant is represented by `E/e`.
- Camel is represented by `M/m`.
- Horse is represented by `H/h`.
- Dog is represented by `D/d`.
- Cat is represented by `C/c`.
- Rabbit is represented by `R/r`.

for 1st and the 2nd player respectively.

Output Format

Initially you are given an empty board. If you are the first player, print the configuration of the pawns that you want to place on row 0 and row 1.

The first input for 1st player is

```
1
.....
.....
.....
.....
.....
.....
.....
.....
```

Your output must contain two rows which has the initial setup of pawns.

Ex,

```
EMHHDDCC
RRRRRRRR
```

This is now given as input to the 2nd player as

```
2
EMHHDDCC
RRRRRRRR
.....
.....
.....
.....
.....
.....
```

The 2nd player now decides his position of pawns by looking at the arrangement of the 1st player.

Ex,

```
emhddcc
rrrrrrrr
```

If you are the second player, print the configuration of the pawns that you want to place on row 6 and row 7.

A player can make utmost 4 moves with a single pawn or split across multiple pawns. A push/pull is counted 2 moves.

Sample Input #00

```
1
EMHHDDCC
RRRRRRRR
.....
.....
.....
.....
emhddcc
rrrrrrrr
```

Sample Output #00

```
1 0 2 0 -1 -1 -1 -1
2 0 2 1 -1 -1 -1 -1
```

```
2 1 3 1 -1 -1 -1 -1
```

Explanation #00

The resulting board is

```
EMHHDDCC
.RRRRRRR
.....
.R.....
.....
.....
emhddcc
rrrrrrr
```

The sample output has 3 lines. There can be utmost 4 and each line must contain 8 space separated integers. If its a non-capturing move, each step must be printed in 1 line. i.e.,

```
old_pos.x old_pos.y new_pos.x new_pos.y -1 -1 -1
```

from the output, we can see that, rabbit of first player at **1 0** is moved to **2 0** and then to **2 1** and subsequently to **3 1**.

Sample Input #01

```
2
.RRREHMR
.rD.R..R
...D....
..Rh....
...d....
R.....C.
...r.H.m
r...rchr
```

Sample Output #01

```
3 3 3 4 3 2 3 3
```

Explanation #01

The resulting board is

```
.RRREHMR
.rD.R..R
...D....
..Rh....
...d....
R.....C.
...r.H.m
r...rchr
```

horse of the 2nd player at **3 3** goes to **3 4** by pulling rabbit of the opponent from **3 2** to **3 3**