

# DMPG '17 S4 - Artillery Battery

---

<b>Time limit:</b> 1.0s	<b>Memory limit:</b> 64M
Java 8: 2.5s	Java 8: 128M
Python: 2.5s	Python: 128M

---

One sunny afternoon, you find yourself in your Geography class watching your teacher Mr. Singh play against a fellow classmate.

Instead of studying for your exams, you decide to visualize the maximum number  $p$ , out of the  $E$  opposing pieces Mr. Singh would be able to capture using just his Cannon. You have also decided to find the minimum number of legal moves  $m$  required to accomplish such a feat.

Since everybody plays this board game in their spare time, it is common knowledge within the [hallways of DMCI](#) that the Cannon is a powerful piece found in both armies. Of course, there is also no need to remind you that there is **exactly one of two** moves to select from on every turn:

**1. Travelling:** The Cannon moves any number of tiles horizontally or vertically, ending on an unoccupied tile. It can continue to move as long as it is not obstructed by any pieces and does not exceed the boundaries of the battlefield. The number of pieces on the board remains unchanged.

**2. Capturing:** The Cannon jumps over *exactly one* piece (called the screen) on the same rank or file, and *must* land on the first enemy piece encountered along the same rank or file on the other side of the screen. The captured piece is then removed from the board.

Even if a cannon is in the position to make a capture, the decision to make one is **not** obligatory.

**Note:** A *rank* is a row and a *file* is a column.

**Note:** By definition, the only tiles that exist between the screen and the starting and destination tiles of the Cannon must be *empty tiles*, if there are any at all.

## Constraints

---

### Subtask 1 [10%]

$$E = 2$$

### Subtask 2 [30%]

$$0 \leq E \leq 8$$

### Subtask 3 [60%]

$$0 \leq E \leq 15$$

There will never be more than 15 enemy pieces on the board: the number of pieces on the opposing army that can theoretically be captured. (The exception is the **general** because it is the only piece that results in *checkmate* instead of it leaving the board when captured.)

## Input Specification

---

Every input file will contain exactly 10 lines of input, staying true to the size of the traditional board.

Each line will contain 9 characters, which will be one of the following:

- denotes Mr. Singh's Cannon (there will be exactly 1 on each board)
- denotes an enemy piece (there will be  $E$  of them)
- denotes an empty tile

## Output Specification

---

If Mr. Singh is able to capture at least one enemy piece, output the two integers  $p$  and  $m$  on one line separated by a space, in the form .

Here,  $p$  is the *maximized* number of pieces captured, and  $m$  is the *minimized* number of legal moves required to capture  $p$  pieces.

If for some strange reason Mr. Singh is unable to capture any pieces using his Cannon, output .

## Sample Input

---

```
.....  
.....  
....E....  
.....  
.....  
....C....  
.....  
....E....  
.....  
.....
```

## Sample Output

---

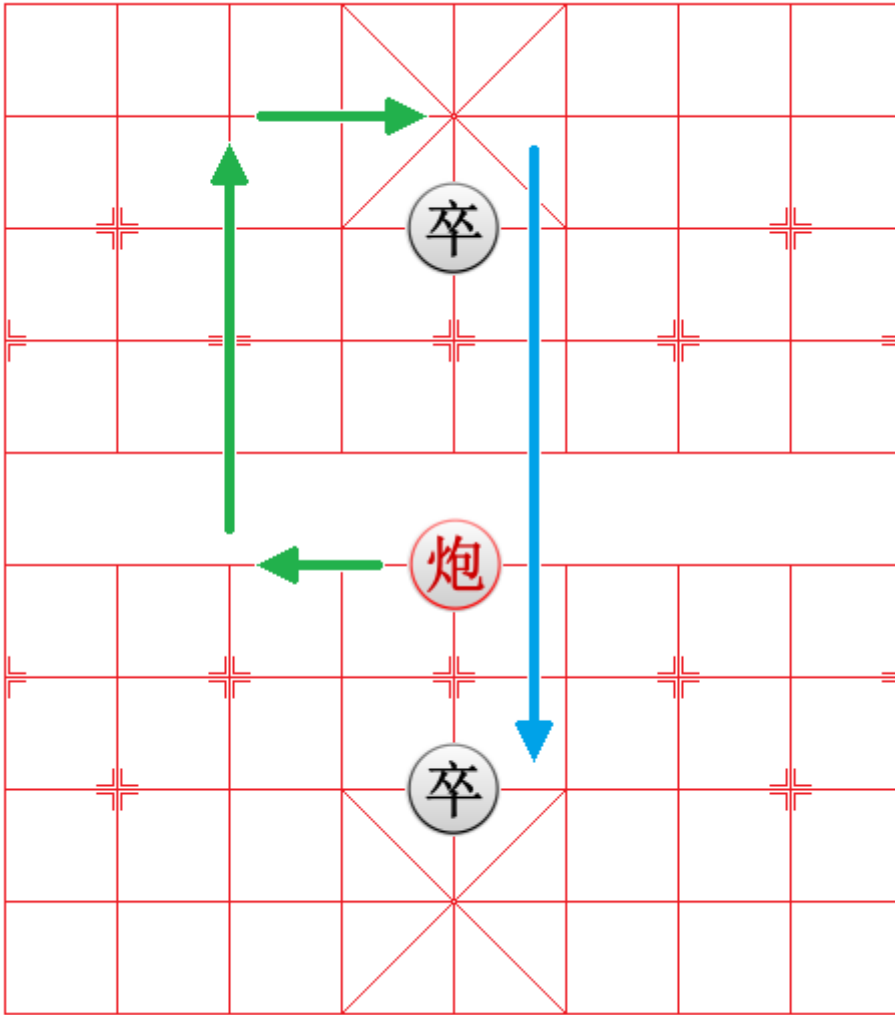
```
1 4
```

## Explanation

---

The diagram representing one solution to the setup is shown below:

Note that Chinese chess is played on the points of intersecting lines ("tiles") instead of the squares themselves.



Here, the red piece represents the Cannon and the two black pawns represent the opposing team.

The 3 green arrows represent the three **traveling** moves required to get the Cannon into position.

The 1 blue arrow represents the **capturing** move made by the Cannon, jumping over exactly 1 screen.

The bottom opposing pawn is captured and removed from the board, and that point becomes the Cannon's new location.

The remaining piece cannot be captured regardless of the legal moves performed by the Cannon, so there is nothing more to be done.

Therefore, the Cannon can capture at most 1 piece, requiring  $3 + 1 = 4$  legal moves to do so.