

## 2. Warships

You have been tasked with the development of an online version of the famous Warships game. You start by developing a console version.

We get as input **the size** of the **field** in which our Warships are situated. The field is **always a square**. After that we will receive the coordinates for the attacks for each player in pairs of integers (row, column). If there is an attack on invalid coordinates - **ignore the attack**. The possible characters that may appear on the screen are:

- \* – a regular position on the field
- < – ship of the first player.
- > – ship of the second player
- # – a sea mine that explodes when attacked

Each time when an attack hits a ship or a mine **replace it with 'X'**. Keep track of the **count of ships remaining for each player**. If a player hits a mine it explodes destroying all ships in adjacent fields (friend or foe). If a player destroys all enemy ships, the program stops and you have to print the following message: **"Player {One/Two} has won the game! {totalCountShipsDestroyed} ships have been sunk in the battle."**

If the list of attack commands ends before any of the player has won, you have to print the following message:

**"It's a draw! Player One has {countOfShips} left. Player Two has {countOfShips} left."**

### Input

- **Field size** – an integer number.
- **Attack commands** – All attack coordinates will be received on a single line. There will be a coma (,) between each set of coordinates and a whitespace (" ") between every two integers representing the row and column to attack.
- **The field: some of the following characters (\*, <, >, #), separated by whitespace (" ");**

### Output

- There are three types of output:
  - If player One wins, print the following output: **"Player One has won the game! {totalCountShipsDestroyed} ships have been sunk in the battle."**
  - If player Two wins, print the following output: **"Player Two has won the game! {totalCountShipsDestroyed} ships have been sunk in the battle."**
  - If there are no more commands and none of the above cases had happened, you have to print the following message: **"It's a draw! Player One has {countOfShips} ships left. Player Two has {countOfShips} ships left."**

### Constraints

- The **field size** will be a 32-bit integer in the range [4 ... 2 147 483 647].
- Player One always starts first.
- A player will **never attack** the coordinates of one of **his own ships**.
- There will be **no incomplete sets** of attack coordinates – e.g. – "0 1,2 3,2".
- There will never be 2 mines in adjacent fields.
- There will not be a test where both players lose their last ships in the same turn.

## Examples

Input	Output	Comment
<pre> 5 0 0,-1 -1,2 2,4 4,4 2,3 3,3 6 # &lt; * &lt; * &gt; &gt; * &lt; * * * &gt; * * &lt; * * * * * * &gt; * * </pre>	<p>Player One has won the game! 5 ships have been sunk in the battle.</p>	<p><b>Player One</b> attacks first at coordinates 0,0 and hits a mine. All adjacent fields (including diagonally) are blasted. The enemy ships at 1,0 and 1,1 are destroyed as well as the friendly ship at 0,1. The field now looks like this:</p> <pre> # X * &lt; * X X * &lt; * * * &gt; * * &lt; * * * * * * &gt; * * </pre> <p><b>Player Two</b> attacks on invalid coordinates so we ignore it.</p> <p><b>Player One</b> attacks 2,2 and destroys an enemy ship.</p> <p><b>Player Two</b> attacks 4,4 and misses.</p> <p><b>Player One</b> attacks 4,2 and destroys the last enemy ship. The game is over so we ignore any remaining attack coordinates and print message. The final state of the map is the following:</p> <pre> X X * &lt; * X X * &lt; * * * X * * &lt; * * * * * * X * * </pre>
<pre> 6 0 0,1 0,5 5 * * * * * &lt; * * * &lt; * * * &gt; &gt; * * * * * * * * * * * * * * * * * </pre>	<p>It's a draw! Player One has 1 ships left. Player Two has 2 ships left.</p>	
<pre> 5 1 1,0 3,2 1,1 3,4 1,3 0 # * * &lt; * * * * &lt; * * * &gt; * * &lt; * * &gt; * * * &gt; * * </pre>	<p>Player Two has won the game! 3 ships have been sunk in the battle.</p>	