

## Description

You can't have a game without rules. It's time to introduce them!

Until now, the players were able to place their dominoes however they like. Now, it is considered a violation. According to the rules, the numbers on the ends of the two neighboring dominoes must match each other. This rule can also be described as a set of two requirements:

1. A player cannot add a domino to the end of the snake if it doesn't contain the matching number.
2. The orientation of the newly added domino ensures that the matching numbers are neighbors.

For example, consider the following situation:

We have a `[3,4], [4,4], [4,2]` snake and a `[1,2]` domino. The domino cannot be added to the left side of the snake because there is no 3 in `[1,2]`. However, the domino can be added to the right side of the snake because `[1,2]` contains a 2. If we were to place the domino on the right side of the snake, we would have to reorient it: `[3,4], [4,4], [4,2], [2,1]`.

These two requirements are strict for both the player and the computer.

## Objectives

Add the following functionality to your code. When it's a player's turn, the program should:

1. Verify that the move entered by the player is legal (requirement #1).  
If not, request a new input with the following message: `Illegal move. Please try again..`
2. Place dominoes with the correct orientation (requirement #2).

When it's a computer's turn, the program should:

1. Try random moves until it finds a legal one.

A set of possible moves ranges from `-computer_size` to `computer_size` (where the `computer_size` is the

number of dominoes the computer still has). Skipping a turn (move 0) is always legal.

## 2. Place dominoes with the correct orientation.

The end-game condition can be achieved in two ways: One of the players runs out of pieces. The first player to do so is considered a winner. The numbers on the ends of the snake are identical and appear within the snake 8 times. For example, the snake below will satisfy this condition:  
[5,5],[5,2],[2,1],[1,5],[5,4],[4,0],[0,5],[5,3],[3,6],[6,5] If this condition is satisfied, it is no longer possible to go on with this snake. Essentially, the game has come to a permanent stop, so we have a draw.

## Examples

The greater-than symbol followed by a space (> ) represents the user input. Note that it's not part of the input.

### Example 1

#### *Invalid move*

```
=====
Stock size: 14
Computer pieces: 6

[6, 6]

Your pieces:
1:[0, 5]
2:[1, 5]
3:[2, 4]
4:[2, 6]
5:[0, 1]
6:[1, 6]
7:[5, 6]

Status: It's your turn to make a move. Enter your command.
> 5
Illegal move. Please try again.
>
```

## Example 2

*Valid move (with corrected domino orientation)*

```
=====
Stock size: 14
Computer pieces: 6

[6, 6]

Your pieces:
1:[0, 6]
2:[5, 5]
3:[4, 4]
4:[4, 6]
5:[0, 1]
6:[0, 5]
7:[1, 6]

Status: It's your turn to make a move. Enter your command.
> 7
=====
Stock size: 14
Computer pieces: 6

[6, 6][6, 1]

Your pieces:
1:[0, 6]
2:[5, 5]
3:[4, 4]
4:[4, 6]
5:[0, 1]
6:[0, 5]

Status: Computer is about to make a move. Press Enter to continue...
>
```