

Carlos always loved playing video games, especially the well-known computer game "Pipes". Today he finally decided to write his own version of the legendary game from scratch.

In this game the player has to place the pipes on a rectangular field to make water pour from each source to a respective sink. He has already come up with the entire program, but one question still bugs him: how can he best check that the arrangement of pipes is correct?

It's your job to help him figure out exactly that.

Carlos has 7 types of pipes in his game, with numbers corresponding to each type:

- 1 - vertical pipe
- 2 - horizontal pipe
- 3-6 - corner pipes
- 7 - two pipes crossed in the same cell (note that these pipes are not connected)

Here they are, pipes 1 to 7, respectively:



Water pours from each source in **each** direction that has a pipe connected to it (thus it can even pour in all four directions). The puzzle is solved correctly only if all water poured from each source eventually reaches a corresponding sink.

Help Carlos check whether the arrangement of pipes is correct. If it is correct, return the number of cells with pipes that will be full of water at the end of the game. If not, return  $-x$ , where  $x$  is the number of cells with water before the first leakage point is reached, or if the first drop of water reaches an incorrect destination (whichever comes first). Assume that water moves from one cell to another at the same speed.

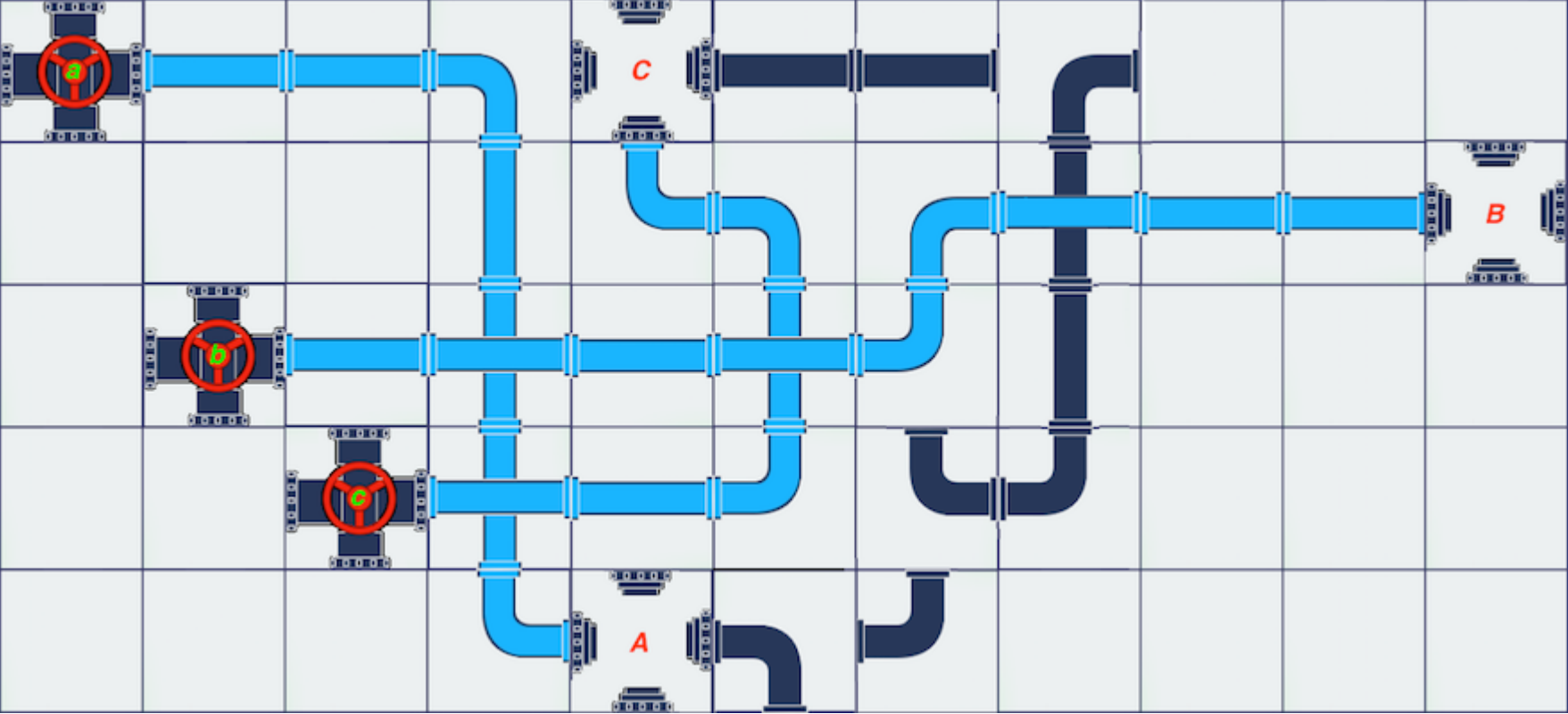
Example

For

```
state = ["a224C22300000",
        "0001643722B00",
        "0b27275100000",
        "00c7256500000",
        "0006A45000000"]
```

the output should be `pipesGame(state) = 19`.

Here is how the game will end:



Input/Output

- [execution time limit] 4 seconds (js)
- [input] array.string state

Array of strings of an equal length representing some state of the game. The pipes are represented by the numbers '1' to '7', the sources are given as lowercase English letters, and the corresponding sinks are marked by uppercase letters. Empty cells are marked with '0'.

It is guaranteed that each letter from the English alphabet either is not present in state, or appears there twice (in uppercase and lowercase).

Guaranteed constraints:

$1 \leq \text{state.length} \leq 100$ ,

$1 \leq \text{state[i].length} \leq 100$ ,

$\text{state[i][j]} \in \{0-7, a-z, A-Z\}$ .

- [output] integer

If the pipe arrangement is correct, return the number of cells with pipes that will be filled with water at the end of the game. If not, return  $-x$ , where  $x$  is the number of cells with water before the first leakage point is reached, or if the first drop of water reaches an incorrect destination.

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
  console.log("This prints to the console when you Run Tests");
  return "Hello, " + name;
}
```