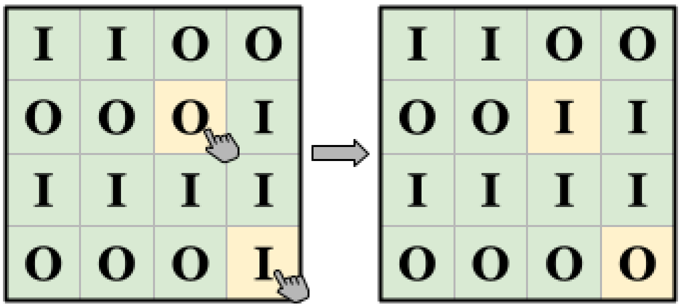
Code Jam to I/O for Women 2022 - Code Jam to I/O for Women

Inversions Organize

Problem

After the troubles with printing advertising for IO two years ago, the marketing team of the conference decided to use an interactive installation. It consists of a matrix of $2\mathbf{N}$ rows and $2\mathbf{N}$ columns of touchscreens. Each touchscreen can display either an uppercase `I` or an uppercase `O`. When one of the screens is touched, it switches the letter it displays to the one it was not displaying right before the touch occurred.

You are looking at one of those installations, and find it to be disorganized. You want to change some of the letters such that the top $\mathbf{N}$ rows show the same number of letter `I`'s as the bottom $\mathbf{N}$ rows, and at the same time, the leftmost $\mathbf{N}$ columns show the same number of letter `I`'s in total as the rightmost $\mathbf{N}$ columns.



I	I	O	O
O	O	O	I
I	I	I	I
O	O	O	I

I	I	O	O
O	O	I	I
I	I	I	I
O	O	O	O

For example, in the left picture above, $\mathbf{N} = 2$. The top 2 rows show 3 letter `I`'s in total, while the bottom 2 rows show 5. On the other hand, both the leftmost 2 columns and the rightmost 2 columns show 4 letter `I`'s. By touching the two highlighted screens we can change the state to that shown in the right picture, which shows 4 letter `I`'s in the top 2 columns and in the bottom 2 columns, and also maintains the balance between the left and right sides.

Given the state of the installation, can you find the minimum number of letter changes needed to achieve your organizational goal?

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow. Each test case starts with a line containing a single integer $\mathbf{N}$, half the number of rows and columns of the matrix. Then, $2\mathbf{N}$ lines follow. The i-th of these contains a string of $2\mathbf{N}$ characters $\mathbf{C_{i,1}C_{i,2}} \cdots \mathbf{C_{i,2N}}$. $\mathbf{C_{i,j}}$ is the letter currently displayed in the screen in the i-th row and j-th column of the matrix.

Output

For each test case, output one line containing `Case #`x`:` y, where x is the test case number (starting from 1) and y is the minimum number of touches required to make the installation simultaneously display the same number of letter `I`'s in its top and bottom halves, and the same number of letter `I`'s in its left and right halves.

Limits

Time limit: 20 seconds.
Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 100$.
$\mathbf{C_{i,j}}$ is either an uppercase `I` or an uppercase `O`, for all i, j.

Test Set 1 (Visible Verdict)

$1 \leq \mathbf{N} \leq 2$.

Test Set 2 (Hidden Verdict)

$1 \leq \mathbf{N} \leq 100$.

Sample

Sample Input

```
3
2
IIOO
OOOI
IIII
OOOI
1
IO
OO
2
OIOI
IOIO
OIOI
IOIO
```

Sample Output

```
Case #1: 2
Case #2: 1
Case #3: 0
```

Sample Case #1 is the one explained in the statement. Not touching anything does not work, and a single touch would leave an odd number of letters `I` in total, so the result cannot be balanced. It is explained in the statement how it can be balanced with two touches (there are other ways).

In Sample Case #2, changing the top left corner to `O` leaves no letter `I`, so all halves have the same amount (0).

In Sample Case #3, the installation is already organized according to your requirements, so no touch is needed.