

Zepsuty wyświetlacz kontratakuje (zepsuty-wyswietlacz-kontratakuje)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Wielki wyświetlacz, przygotowany przez komitet główny Bitockiej Olimpiady Informatycznej Juniorów (ten sam, co w zadaniu *Zepsuty wyświetlacz*), jest bardziej wadliwy, niż początkowo przypuszczano. Nie dość, że źle prezentuje wyniki podzielne przez pewną liczbę, to w dodatku resetowanie go jest niesamowitą udręką! Za każdym razem, aby zmienić to, co w danym momencie się wyświetla, należy najpierw wyczyścić całą matrycę, a dopiero później rysować to, co powinno być w tym momencie wyświetlane.

Wyświetlacz składa się z matrycy $N \times N$ pikseli. Każdy piksel może przybrać dwa kolory, biały (oznaczony znakiem kropki `.`) lub czarny (oznaczone znakiem krzyżyka `#`). Resetowanie wyświetlacza polega na zmianie wszystkich pikseli tak, żeby były czarne. Jednakże, nie da się tego osiągnąć w bezpośredni sposób. Jedyna możliwość, żeby zmienić to, jak aktualnie świecą się piksele, jest wybranie pewnego wiersza pikseli oraz pewnej kolumny, a następnie przerysowanie tego wiersza w danej kolumnie. Rozważmy poniższy przykład wyświetlacza rozmiaru 3×3 :

```
.##  
#. #  
#..
```

W tej sytuacji, gdybyśmy wybrali trzeci wiersz, po czym przerysowali go w kolumnie pierwszej, wyświetlacz zmieniłby swój stan do poniższego:

```
###  
..#  
...
```

Gdyby teraz wybrać pierwszy wiersz, a następnie przerysować go w każdej z kolumn, otrzymalibyśmy wyświetlacz cały ustawiony na czarno:

```
###   ###   ###   ###  
..# -> #.# -> ### -> ###  
...   #..   ##.   ###
```

Okazuje się, że za czyszczenie wyświetlacza (30 razy w każdej sekundzie!) odpowiedzialny jest krasnal Bitek, ukryty pod podłogą sali zawodów (podobnie, jak krasnale sterujące światłami drogowymi, które, jak każdy wie, są ukryte pod jezdnią). Krasnal Bitek ledwo daje radę i bardzo przydałby mu się program, który pomoże mu w czyszczeniu wyświetlacza. Pomóż mu i napisz program, który policzy, ile razy Bitek musi wykonać operację przerysowania, żeby zmienić kolor wszystkich pikseli na czarny.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N oznaczająca wymiary wyświetlacza. W następnych N wierszach następuje opis początkowego stanu wyświetlacza, i -ty wiersz składa się z N znaków kropki `.` lub krzyżyka `#`, j -ty znak opisuje stan j -tego piksela w i -tym wierszu.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę całkowitą, oznaczającą minimalną liczbę operacji, która wystarczy, aby wszystkie piksele miały czarny kolor, lub -1 , jeżeli nie jest to możliwe.

Ograniczenia

$1 \leq N \leq 1000$.

Podzadania

Podzadanie	Warunki	Punkty
1	W każdej kolumnie jest co najmniej jeden czarny piksel.	21
2	W każdym wierszu jest co najwyżej jeden biały piksel.	23
3	$N \leq 50$.	18
4	Brak dodatkowych ograniczeń.	38

Przykład

Wejście

```
3
.##
#.#
#..
```

Wyjście

```
4
```

Wejście

```
6
.....#
#..#.#
....#.
#..#.#
...##.
.#.....
```

Wyjście

```
9
```

Wejście

```
2
..
..
```

Wyjście

```
-1
```