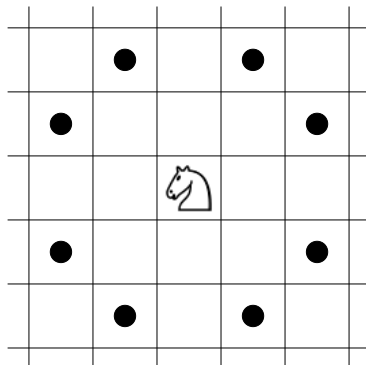


B. Skoczki

Dostępna pamięć: 64 MB

Jak wygląda szachowy skoczek, każdy chyba wie. Postawiony na polu szachownicy skoczek atakuje pola zaznaczone kropką na poniższym rysunku. Jeśli na atakowanym polu stoi inny skoczek, to mówimy, że para skoczków się (wzajemnie) atakuje.



Dana jest długa szachownica o rozmiarach $3 \times n$, gdzie n jest dodatnią liczbą całkowitą. Część pól szachownicy jest *stara* i nie nadaje się do użytku (a tym bardziej do stawiania na nich skoczków). Twoim zadaniem jest określenie, jaka jest największa liczba skoczków, które można ustawić na szachownicy, tak żeby żadna para skoczków się nie atakowała.

Specyfikacja danych wejściowych

W pierwszym wierszu danych wejściowych znajduje się liczba naturalna $1 \leq n \leq 2 \cdot 10^5$, będąca długością szachownicy. W każdym z kolejnych n wierszy znajduje się ciąg trzech znaków, opisujący n -ty wiersz szachownicy. Każdy z tych znaków jest równy albo „x” (stare pole szachownicy) albo „.”. Znaki nie są oddzielone spacjami.

Specyfikacja danych wyjściowych

W pierwszym i jedynym wierszu wyjścia Twój program powinien wypisać jedną liczbę naturalną będącą maksymalną liczbą nieatakujących się wzajemnie skoczków, które można umieścić na zadanej szachownicy, tak że żaden ze skoczków nie stoi na starym polu.

Przykład A

Wejście:

```
2
. . x
x . .
```

Wyjście:

```
3
```

Przykład B

Wejście:

```
3
...
...
...
```

Wyjście:

```
5
```

Przykład C

Wejście:

```
3
xxx
.x.
.x.
```

Wyjście:

```
2
```