

# Gra terenowa (gra-terenowa)

Memory limit: 128 MB      Time limit: 1.00 s

Jasio zabrał się ostatnio za pisanie gry komputerowej – *Symulator Gry Terenowej*. Jednym z problemów jaki musi rozwiązać jest generowanie planszy dla graczy.

Każda plansza reprezentuje kawałek lasu o długości  $N$  i szerokości  $M$  metrów. Każdy metr kwadratowy lasu może być albo zarośnięty przez drzewa i krzewy (oznaczamy to pole za pomocą symbolu #), albo kawałkiem polany (używamy symbolu 0).

Na początku rozgrywki, gracz musi wybrać lokalizację swojej *bazy*, gdzie będzie się znajdowała flaga, którą będzie chciał wykraść przeciwnik. Z oczywistych względów, najlepszym kandydatem na bazę jest pole, które jest polaną, oraz dla którego wszystkie pola, które mają z nim chociaż jeden punkt wspólny są zarośnięte.

Jasio ma przygotowaną pewną liczbę *szablonów map*, dla których niektóre pola mają już przypisane wartości, niektóre zaś jeszcze nie (te komórki oznaczmy symbolem ?). Wiadomo, że dla każdej komórki ? ma ona **co najwyżej dwóch** sąsiadów o wartości ?.

Jasio zastanawia się teraz dla danego szablonu, ile maksymalnie pól będzie spełniało wymogi dobrej bazy, jeżeli w optymalny sposób przypisze wartości wszystkim komórkom (takie plansze będą ciekawsze dla graczy).

Napisz program, który odpowie na pytanie Jasia i niniejszym przyspieszy publikację gry Jasia.

## Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne  $N$  i  $M$  oznaczające wymiary kawałka lasu, na którym będzie toczyła się rozgrywka.

W kolejnych  $n$  wierszach znajduje się po  $m$  symboli 0, # lub ?, o podanym wyżej znaczeniu.

## Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć maksymalna liczba pól spełniających kryteria dobrej bazy po przypisaniu wartości komórkom ?.

## Ograniczenia

$1 \leq N, M \leq 1\,000$ .

## Przykład

Input	Output
5 6	3
#####?	
#?0##?	
##?###	
###0?	
#00##0	

Input	Output
1 1	1
?	