



## Powódź

W zeszłym miesiącu okrutna pogoda doprowadziła do powodzi w Hotelu Pela. Szkody sporządzone przez wodę były ogromne, a właściciele hotelu – zdeterminowani, by uniknąć podobnych powodzi w przyszłości. Zdecydowali więc skorzystać z pomocy paru zawodowych programistów i poprosili ich o to, by zaplanowali możliwie najbardziej efektywną naprawę szkód, mając do dyspozycji budżet K euro.

W hotelu Peja jest N kwadratowych pokoi, z których każdy ma rozmiar RxR komórek. Komórki mogą mieć wysokości oznaczone cyframi od '0' do '9', gdzie '0' oznacza najniższą, a '9' – najwyższą możliwą wysokość. Niektóre z komórek nazwiemy ciekącymi, ponieważ nad każdą z tych komórek jest szczelina w suficie, prowadząca do przecieków. Miejsca, w których sufit przecieka, są oznaczone literą 'L'.

Właściciele hotelu byli zdesperowani i w celu powstrzymania powodzi postanowili postawić wysokie wiadra na wodę na ciekących komórkach. Niestety, jeśli szczelina w suficie nie zostanie usunięta, pojemnik na wodę prędko się napełni, a woda wyleje się na sąsiednie komórki. Aby temu zapobiec, zadaniem programistów było zdecydowanie, które szczeliny w suficie należy załatać (jedno łatanie kosztuje 1 euro). Powiemy, że dwie komórki sąsiadują, jeśli mają wspólny bok (zatem żadna komórka nie ma więcej niż 4 sąsiadów). Gdy tylko woda wyleje się na niecieknącą komórkę o wysokości H, rozleje się na wszystkie sąsiednie komórki o wysokości  $\leq H$ .

Jak widzisz, programiści poradzili sobie z tym zadaniem doskonale i Hotel Pela wygląda świetnie. A co by było, gdyby to na Ciebie spadła odpowiedzialność rozwiązania problemu? Czy jesteś w stanie napisać program, który obliczy największą możliwą liczbę komórek (zsumowaną ze wszystkich pokoi), które hotel jest w stanie uchronić przed wodą używając przydzielonego budżetu? Zatkana ciekąca komórka to komórka sucha. Oczywiście nie musisz wykorzystać całego budżetu.

## Wejście

W pierwszej linii standardowego wejścia zapisano trzy liczby całkowite: N ( $1 \leq N \leq 300$ ) – liczbę pokoi, K ( $1 \leq K \leq 3\,000$ ) – przydzielony budżet oraz R ( $1 \leq R \leq 20$ ) – rozmiar boku każdego pokoju.

Następnie zapisano opisy kolejnych pokoi (jest ich N), oddzielone pustymi liniami.

Opis pojedynczego pokoju wygląda następująco: R linii po R znaków. Każda linia składa się tylko z cyfr 0-9, oznaczających wysokości niecieknących komórek i liter L, symbolizujących ciekące komórki.

Każdy pokój może mieć od 0 do 10 ciekących komórek. Żadne dwie ciekące komórki nie sąsiadują ze sobą.



## Wyjście

Twój program powinien wypisać jedną liczbę całkowitą: maksymalną liczbę komórek (zsumowaną ze wszystkich pokoi), które można uchronić przed wodą.

## Przykład

Wejście	Wyjście
2 1 4 0123 1234 2345 3L56  L876 8765 7654 6543	20

Jeśli szczelina w pierwszym pokoju nie zostanie załatana, pokazane niżej komórki zostaną zalane:

\*\*\*3

\*\*\*4

\*\*\*5

\*L\*6

Jeśli szczelina w drugim pokoju nie zostanie załatana, woda zaleje cały pokój! Ponieważ budżet wystarczy na załatanie tylko jednej dziury, lepiej jest załatać tę w drugim pokoju. Zatem cały drugi pokój pozostanie suchy ( $4 * 4 = 16$  komórek) i 4 komórki z pierwszego pokoju (pokazane powyżej) również pozostaną suche. Łączna liczba ocalałych komórek wyniesie więc 20.

## Punktacja

W testach wartych łącznie 10% wszystkich punktów będzie tylko jeden pokój (tzn.  $N = 1$ ).

W innych testach wartych łącznie 20% wszystkich punktów łączna liczba wszystkich ciekących komórek we wszystkich pokojach nie przekroczy 20.

W jeszcze innych testach wartych łącznie 20% wszystkich punktów w każdym pokoju będzie co najwyżej jedna szczelina w suficie.