

# Penalidade mínima

Nome do arquivo fonte: `penalidade.c`, `penalidade.cpp`, ou `penalidade.pas`

A Sra. Bastos é uma elaboradora de passatempos matemáticos e pediu para que você criasse um programa que conseguisse jogar de forma eficiente a sua mais nova criação.

O jogo consiste em um tabuleiro formado por casas dispostas em  $N$  linhas por  $N$  colunas. Cada casa contém um inteiro não-negativo. No começo do jogo, uma peça é colocada na casa localizada no canto superior esquerdo, ou seja, na posição (1,1). O objetivo do jogo é mover a peça até a casa localizada no canto inferior direito (posição (N,N)) somente movendo um único quadrado para baixo ou para a direita em cada passo. Além disso, a peça não pode ser colocada em nenhum quadrado que contenha o número zero.

O *custo* do caminho utilizado para percorrer o tabuleiro corresponde ao produto de todos os números das casas percorridos no caminho. A *penalidade* é definida utilizando a representação decimal do custo, sendo representada pelo número de dígitos zeros, contados da direita para a esquerda, antes do primeiro dígito diferente de zero. Por exemplo, um custo igual a 501000 tem penalidade 3, e um custo igual a 501 tem penalidade zero.

O objetivo do jogo é conseguir chegar à casa (N,N) através de um caminho “otimizado”. Dizemos que o caminho foi otimizado se a penalidade for mínima.

## Tarefa

Escreva um programa que, dado um tabuleiro, determine a penalidade do custo otimizado.

## Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha da entrada contém um inteiro  $N$  que indica o número de linhas e colunas do tabuleiro ( $1 \leq N \leq 1000$ ). As  $N$  linhas seguintes contém  $N$  inteiros  $I$  cada ( $1 \leq I \leq 1000000$ ), que representam o valor da casa do tabuleiro naquela posição. Existe pelo menos uma solução possível para todos os casos de teste.

## Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo a penalidade do custo “otimizado”.

Entrada	Entrada	Entrada
3	3	4
1 2 3	5 7 6	1 3 0 0
4 5 6	4 0 1	0 8 2 25
7 8 9	3 2 5	6 5 0 3
		0 15 7 4
Saída	Saída	Saída
0	1	2