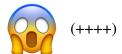
## 11 Batalha naval





Pedro e Paulo gostam muito de jogar *Batalha Naval*. Apesar de serem grandes amigos, Pedro desconfia que Paulo não esteja jogando *honestamente* e, para tirar essa dúvida, decidiu usar um programa de computador para verificar o resultado de cada jogo.

Acontece que Pedro não sabe programar e, por isso, pediu a sua ajuda para elaborar este *programa de auditoria naval*, explicando-lhe que cada jogador do jogo *Batalha Naval* possui um tabuleiro retangular com n linhas e m colunas  $(n, m \in \mathbb{N}^*)$  para representar o "campo de batalha", onde:

- cada posição é um quadrado que pode conter água ('a') (letra 'a', minúscula) ou uma parte de um navio ('#') (um símbolo *hashtag*);
- dois quadrados são ditos *vizinhos* se possuem um lado comum, ou seja, um lado que pertence a ambos quadrados;
- se duas partes de navio estão em posições vizinhas, então essas duas partes pertencem ao mesmo navio;
- é proibido que quadrados de duas partes de navios distintos tenham um *canto* em comum, ou seja, que quadrados de duas partes de navios distintos compartilhem um *vértice*;
- para que um navio de um jogador seja destruído por *disparos* de seu oponente é necessário que o oponente acerte todas as partes do navio, o que pode exigir um número indeterminado de disparos.

O jogo consiste em *disparos* alternados entre os dois jogadores, sendo que cada disparo que um jogador faz em *direção* ao tabuleiro do seu oponente deve ser feito tendo como *alvo* um único quadrado daquele tabuleiro. Para fazer um *disparo*, um jogador informa ao outro a linha L  $(1 \le L \le n)$  e a coluna C  $(1 \le C \le m)$  do quadrado alvo de seu disparo. Considere que os jogadores não se esquecem de seus disparos anteriores e, por isso, nunca *atiram* no mesmo lugar mais de uma vez.

Sua tarefa é escrever um programa em  $\mathbb{C}$  que *simule* uma partida deste jogo a partir da configuração do tabuleiro e de uma sequência de disparos feitos por um dos jogadores, determinando o número de navios do outro jogador que foram destruídos pela sequência de disparos.

#### **Entrada**

A primeira linha da entrada contém dois números inteiros n e m ( $1 \le n, m \le 100$ ) representando, respectivamente, o número de linhas e de colunas do tabuleiro.

As *n* linhas seguintes correspondem ao tabuleiro do jogo. Cada uma dessas linhas contém *m* caracteres, sendo que cada caractere indica o conteúdo da posição correspondente no tabuleiro. Se esse caractere for 'a' (letra 'a', minúscula), essa posição contém água; se o caractere for '#' (*hashtag*), essa posição contém uma parte de um navio.

A próxima linha contém um número k ( $1 \le k \le n \times m$ ) que representa o número de disparos feitos pelo jogador em direção ao tabuleiro de seu oponente.

As próximas k linhas indicam os *disparos* feitos pelo jogador, sendo que cada linha contém dois inteiros L e C, indicando a linha e a coluna do disparo feito, lembrando que  $1 \le L \le n$  e  $1 \le C \le m$ .

#### Saída

Seu programa deve imprimir uma única linha contendo um único número natural: o número de navios destruídos do jogador oponente ao que realizou os disparos.

### **Exemplos**

Entrada	Saída
5 5	4
aa#a#	
#aaaa	
aaa#a	
#aaaa	
aaa#a	
5	
1 3	
1 4	
1 5	
2 1	
3 4	

Entrada	Saída	
5 5	2	
aa###		
aaaaa		
#####		
aaaaa		
#a##a		
5		
5 1		
5 2		
1 3		
1 4		
1 5		

Entrada	Saída
7 7	1
a#aaaa#	
###-a##	
a#aaaa#	
aaaa#a#	
a#aa#a#	
a####a#	
aaaaaaa	
8	
1 1	
1 2	
2 1	
2 2	
2 3	
3 2	
5 2	
6 2	

# Observações

Se você domina alguma linguagem de programação que permita o desenvolvimento de programas que apresentem uma GUI (*Graphic User Interface*) para seu usuário, que tal pensar em desenvolver este *joguinho* a utilizando?