

Campo de abóboras



Hagrid está tendo um dia cheio com Bicuço e pediu ajuda a Harry e Ron. Eles precisam ajudá-lo a coletar abóboras. **A plantação é grande e para acelerar o trabalho, Hagrid pede que Harry fique responsável pela colheita em uma determinada linha, começando na esquerda e indo até o fim dela na direita. Da mesma forma, Hagrid pede que Ron colha as abóboras em outra linha, porém agora, Ron começa em cima e vai até o fim dela na parte de baixo da plantação.**

Hagrid quer saber ao fim da tarefa qual dos dois coletou mais abóboras, levando-se em consideração seu peso. **Só tome cuidado com o ponto de intersecção entre as duas linhas que Harry e Ron irão coletar as abóboras. Somente um deles fica com a abóbora que está lá, ela é do primeiro que lá chegar, ou seja, ela pertence àquele que estiver mais próximo dela a partir do ponto de início de sua colheita, e se der empate na distância, a abóbora fica com Ron.**

Entrada

Seu programa receberá primeiramente um inteiro '**N**' ($1 \leq 'N' \leq 100$), representando o tamanho da plantação de abóboras, que é um campo de proporção NxN (N linhas horizontais por N linhas verticais). A seguir serão dadas '**N**' linhas, onde em cada uma serão dados '**N**' inteiros '**P**' ($1 \leq P \leq 100$), que representam o peso de cada abóbora no campo. Há uma abóbora em cada posição do campo NxN. Por fim, a última linha da entrada contém as linhas '**X**' e '**Y**' ($0 \leq X, Y < N$) que Harry e Ron irão coletar, respectivamente. **Cuidado que a linha de Ron na verdade se trata de uma coluna na matriz da plantação.**

Saída

Imprima o peso total da colheita de Harry e a seguir, na linha de baixo, imprima o peso total da colheita de Ron, como nos exemplos abaixo.

Entrada	Saída
4 1 2 3 4 5 6 7 8 1 3 5 7 2 4 6 8 1 2	Harry 19 Ron 21
4 1 2 3 4 5 6 7 8 1 3 5 7 2 4 6 8 2 1	Harry 16 Ron 12
3 1 2 3 4 5 6 7 8 9 1 1	Harry 10 Ron 15

Oxi véi, cadê a praia?

Autor: Hérus Conceição

Você está desenvolvendo um programa de navegação marítima e exploração de ilhas. Você percebeu que o Dev responsável pelo mapa se esqueceu de um detalhe: as praias. Você precisa resolver esse problema criando um programa que transforma em praia aquilo que deveria ser praia.

Entrada

A entrada será dada por um mapa de dimensões **10x10**. Cada uma das **10** linhas de entrada possuirá **10 caracteres separados por espaços**, sendo cada caractere ou **'*'** para representar a **água** ou **'t'** para representar a **terra**.

Saída

O programa deve imprimir o mapa corrigido, transformando em praia **'p'**, todo pedaço de **terra 't'** que estiver em **contato direto** com a **água '*'**, **verticalmente** ou **horizontalmente**. O que estiver fora do mapa não deve ser considerado como água.



Entrada	Saída
* * * * * * * * * *	* * * * * * * * * *
* * t t t t * * * *	* * p p p p * * * *
* t t t t t t * * *	* p p p t t p * * *
* * * * t t t t t *	* * * * p t t p p *
* * * * * t t t t *	* * * * * p t t p *
* * * * * t t t *	* * * * * p t p * *
* * * * t t t t * *	* * * * p t t p * *
t t t t t t t t *	p p p p t t t t p *
t t t t t t t t *	t t t t t p p p * *
t t t t t * * * *	t t t t p * * * *
t * * * * * * * t	p * * * * * * * p
* t t t t t t t *	* p p p p p p p *
* t t t t t t t *	* p t t t t t t p *
* t t t t t t t *	* p t t t t t t p *
* t t t t t t t *	* p t t t t t t p *
* t t t t t t t *	* p t t t t t t p *
* t t t t t t t *	* p t t t t t t p *
* t t t t t t t *	* p t t t t t t p *
* t t t t t t t *	* p p p p p p p *
t * * * * * * * t	p * * * * * * * p

Super drone



A Amazônia é grande e exuberante por natureza. Para fazer um levantamento das espécies de árvores, os cientistas desensolveram um super drone que é capaz de sobrevoar uma determinada região e catalogar quantas espécies diferentes de árvores existem por quilômetro quadrado.

Sua tarefa é, dada uma lista de coordenadas no mapa onde está sendo feito o levantamento, determine quantas espécies de árvores existem dentro daquelas áreas.

Entrada

A entrada será primeiramente um inteiro '**N**' ($4 \leq N \leq 10$), representando as dimensões do mapa. Cada uma das '**N**' linhas seguintes possuirá '**N**' inteiros '**Q**' ($100 \leq Q \leq 1000$), separados por espaço. A seguir será dado um inteiro '**C**' ($1 \leq C \leq N*N$), que representa o número de coordenadas a serem verificadas. Por fim, serão dadas '**C**' linhas, onde em cada uma serão dados dois inteiros '**X**' e '**Y**' ($0 \leq X, Y \leq N-1$), representando as coordenadas a serem verificadas pelo drone, sendo que '**X**' representa determinada linha e '**Y**' representa determinada coluna no mapa.

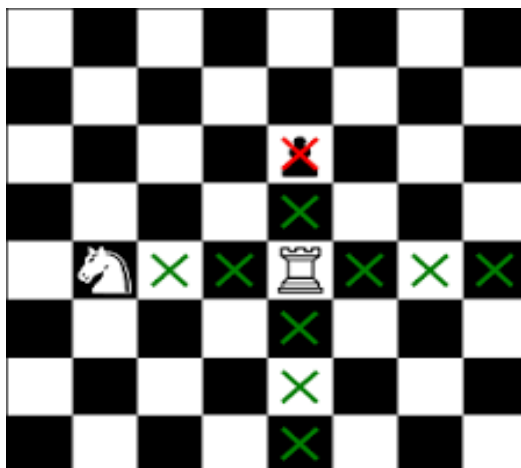
Cada coordenada no mapa representa uma região de 1 KM². Cada inteiro 'Q' em cada posição do mapa indica o número de espécies de árvores naquela coordenada. Não serão dadas duas coordenadas iguais a serem verificadas.

Saída

Você deverá imprimir a quantidade de espécies verificadas em todas as coordenadas solicitadas no mapa.

Entrada	Saída
4 100 200 300 400 100 200 300 400 400 300 200 100 400 300 200 100 3 0 2 2 3 3 3	500

Torre Xadrez



Xadrez é sem dúvida um dos jogos mais famosos e que exige grande capacidade intelectual e estratégica. São várias peças que possuem diferentes tipos de movimentos no tabuleiro. A torre, por exemplo, move-se em uma linha ou em uma coluna, como pode ser observado na figura acima, onde as posições marcadas com um X destacam aquelas onde a torre pode ir. Havendo uma peça inimiga a sua frente, esta peça pode ser derrotada pela torre, é o caso do peão na figura acima. Porém, se houver uma peça aliada a frente, a torre poderá se mover até a posição imediatamente anterior a peça aliada, é o caso do cavalo na figura acima.

Sendo assim, você foi escolhido para desenvolver um programa que diz quantas peças inimigas a torre poderá possivelmente derrotar, a partir de uma posição X, Y no tabuleiro que indica onde a torre está.

Entrada

A entrada será primeiramente uma grade de tamanho **'8 x 8'**, representando o tabuleiro de xadrez. Cada uma das **'8'** linhas do tabuleiro possuirá **'8'** inteiros **'Q'** ($0 \leq Q \leq 2$), separados por espaço. Portanto, cada posição do tabuleiro possuirá 3 valores possíveis: **0** - para indicar que naquela posição não tem peça; **1** - para indicar uma peça aliada; **2** - para indicar que naquela posição há uma peça inimiga. Por fim, serão dados dois inteiros **'X'** e **'Y'** ($0 \leq X, Y \leq 7$), representando a coordenada inicial da torre, sendo que **'X'** representa uma linha e **'Y'** representa uma coluna. Além disso, na posição **X - Y** terá o valor **1**, pois representa a própria torre.

Saída

Você deverá imprimir a quantidade de peças inimigas no caminho da torre.

Entrada	Saída
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 1 1 2 0 2 0 0 0 0 0 0 0 2 0 0 0 0 0 2 2	2

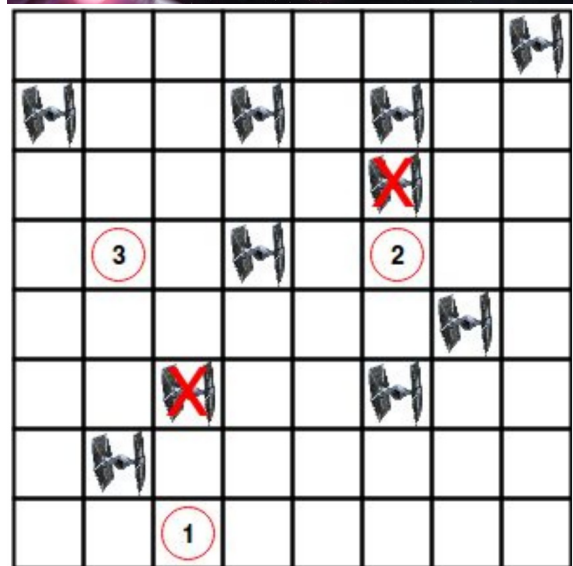
Batalha de Yavin



A Batalha de Yavin, a maior batalha da Guerra Civil Galáctica, resultou na destruição da primeira Estrela da Morte. Os dois lados do confronto foram o Império Galáctico e a Aliança Rebelde. A Estrela da Morte chegou no sistema escoltada por uma frota constituída por diversas naves, dentre elas, as velozes Tie Fighters.

O Comando Rebelde estava desacreditado. Com quase todos os pilotos destruídos, o destino da batalha estava nas mãos de um jovem piloto, mas este trazia consigo uma nova esperança, afinal, esse jovem era Luke Skywalker. Ele sabia que poderia usar a dobra espacial e o poder da força para se teletransportar com sua nave muito mais rápido de um lugar para outro no espaço e, assim, conseguir destruir o máximo possível de naves inimigas.

Dadas as coordenadas de cada nave inimiga e de cada teleporte de Luke, diga quantas naves ele consegue destruir. Para cada coordenada onde Luke teleporta, ele dá apenas um tiro de Prótons que é capaz de destruir a primeira nave que esteja em sua frente. No exemplo ao lado, ao teleportar para a posição marcada como (2), ele atira e destrói somente a nave que está em sua frente. Já ao teleportar para a posição (3), ele atira mas não acerta nenhuma nave.



Entrada

A primeira linha da entrada possui um inteiro '**N**' ($3 \leq N \leq 100$), indicando que a matriz que representa o espaço possui dimensões **NxN**, e um inteiro '**M**' ($1 \leq M \leq 1000$), indicando o número de teleportes realizados por Luke. As próximas '**N**' linhas possuem '**N**' inteiros em cada uma, cujos valores podem ser 0 (se não existe nave naquele quadrante) ou 1 (se existe uma nave naquele quadrante). Nas próximas '**M**' linhas serão dadas as coordenadas inteiras (**L** - linha **C** - coluna) de cada teleporte de Luke, um por linha. É garantido que Luke não aparecerá numa coordenada que possui uma nave, já que dois corpos não podem ocupar o mesmo lugar no espaço ao mesmo tempo. Os teleportes ocorrerão somente dentro do espaço dado.

Saída

A saída consiste de apenas um inteiro que é o número de naves destruídas por Luke.

Entrada	Saída
8 3 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 2 3 5 3 1	2
4 2 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 1 3 2 3 1	2