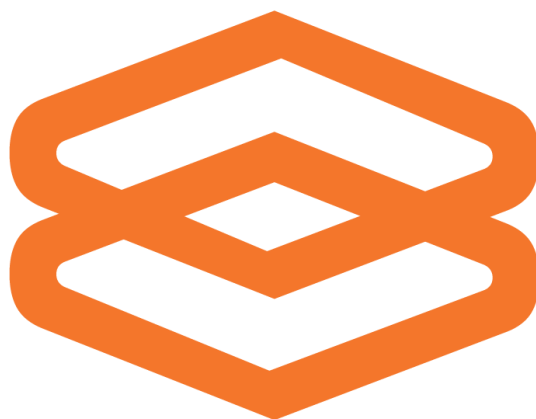


Renova Soluções em Tecnologia

Lista de Atividades



RENOVA
S O L U Ç Õ E S E M T E C N O L O G I A

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

Faxina

Autor: Gabriel Dahia

Você decidiu se livrar dos livros em sua casa. Seu critério para decidir se um livro vai ser doado ou não é a quantidade de consoantes em seu título. Caso o título possua mais do que T consoantes, ele será doado; caso contrário, ele ficará em sua estante.

Sua tarefa é determinar, dados os títulos dos livros, quais serão doados.

Entrada

A primeira linha da entrada contém dois inteiros N e T , respectivamente o número de livros na estante e o número máximo de consoantes permitidas no título de cada livro. As próximas N linhas contêm os títulos dos livros, que tem no máximo 20 símbolos. Estes são apenas letras minúsculas e espaços.

Saída

Para cada livro, seu programa deve imprimir 0 caso o livro deva ser doado e 1 caso ele fique na estante, seguidos de um fim de linha.

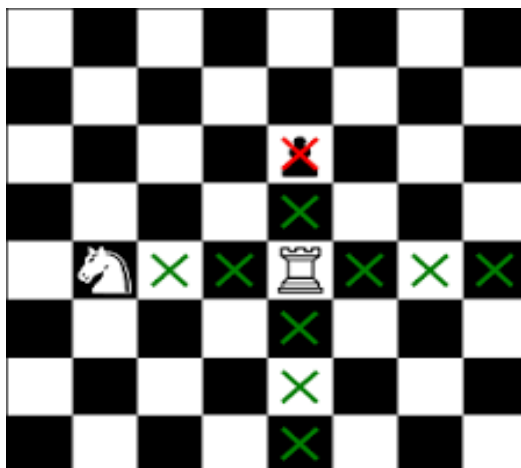
Limites

- $1 \leq N \leq 10^5$;
- $1 \leq T \leq 20$.

Exemplos

Entrada	Saída
3 4	
harry potter	0
senhor dos aneis	0
aleph	1

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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 2 0 0 0 0 0 2 2	2

Palíndromos

Autor: Ubiratan Neto

Luiz e seus colegas estão participando de uma série de jogos na aula de português em sua escola. Em um dos jogos, é dado uma palavra para cada um deles, e eles precisam dizer se ela é um palíndromo ou não. Uma palavra é um palíndromo se ela é a mesma palavra lida de trás pra frente. Por exemplo “arara”, “osso”, “ralar”, etc. Luiz deseja vencer o jogo mesmo que seja trapaceando, e pediu a você que fizesse um programa que respondesse se uma palavra era um palíndromo ou não.

Entrada

A primeira linha da entrada possui uma palavra S , contendo apenas letras minúsculas, a palavra dada por Luiz.

Saída

A saída deve conter numa única linha a palavra “Sim”, caso S seja um palíndromo, e “Nao” caso “S” não seja um palíndromo.

Limites

- $1 \leq S \leq 100$

Exemplos

Entrada	Saída
arara	Sim

Entrada	Saída
reviver	Sim

Entrada	Saída
rezar	Nao

Xeroque Rolmes

Autora: Laila Mota

Xeroque Rolmes é um primo distante de um detetive renomado e agora quer seguir os passos do seu primo. Em seu primeiro caso, uma testemunha lhe disse que na casa do suspeito existe um cofre com uma pasta com várias provas de seus vários crimes. Ao investigar o lugar, Xeroque Rolmes percebe que do lado do cofre existem 6 pedaços de papel colados na parede com palavras escritas e, mais que isso, ele descobriu que a quantidade de letras de cada palavra corresponde a um dígito da senha do cofre.

Entrada

A entrada é composta por 6 linhas, cada linha contém uma palavra encontrada.

Saída

A saída é composta por uma linha com a senha do cofre.

Limites

$$1 \leq N \leq 10$$

Exemplos

Entrada	Saída
sh embhtots m qgexyzbcu wwhzzw rdfxs	281965

Entrada	Saída
a a yni xbzaxh lyg q	113631

Caçando Pokémons



Ash e seus amigos estão em uma aventura em busca de novos pokémons. Além disso, decidiram ver ao final da busca quem conseguiu mais pokémons de um certo tipo. Sua tarefa é criar um programa que **diga quantos pokémons de um certo tipo Ash pegou**. Para isso será dada uma matriz representando a área onde ele está. Para cada posição (x ; y) da matriz, **se o valor for 0 (zero), significa que não tem nenhum pokémon naquela posição e se o valor for um número 'T', diferente de zero, significa que tem um pokémon do tipo 'T' naquela posição** que pode ser pego por Ash.

Entrada

Na primeira linha serão dados dois inteiros '**N**' e '**M**' ($1 \leq N, M \leq 100$) que diz a quantidade de linhas e colunas da matriz, respectivamente. As próximas '**N**' linhas terão '**M**' inteiros '**T**' ($0 \leq T \leq 100$) em cada, representando o tipo do pokémon ou se não há pokémon. Por fim, na última linha, será dado um inteiro '**P**' ($1 \leq P \leq 100$), representando o tipo do pokémon a ser pego por Ash.

Saída

A saída consiste em 1 linha contendo a frase "**Ash pegou 'Q' pokemon**" onde '**Q**' deve ser a quantidade de pokémons do tipo '**P**' pegos por Ash, podendo ser inclusive 0 (zero).

Entrada	Saída
4 4 0 1 0 0 2 0 2 0 0 1 0 0 0 0 0 2 2	Ash pegou 3 pokemon
5 10 0 1 0 0 0 3 0 0 0 0 0 2 0 0 0 1 0 0 0 2 0 3 0 0 0 0 2 0 0 0 8 0 1 0 0 3 0 8 0 0 0 0 0 0 0 0 0 0 1 0 1	Ash pegou 4 pokemon

Inventário caótico

Autor: Gustavo Amaral

Jônatas da Silva após várias horas jogando Dark Souls III percebeu que seu inventário está caótico e não consegue encontrar seus itens. **Seu objetivo é identificar se um item está no inventário.**



Entrada

A entrada cosistirá de várias linhas, onde em cada uma teremos **uma string representando o nome de determinado item. A entrada com o nome dos itens termina** quando for digitado **a palavra “fim”**. Por fim, **será dado o nome do item que Jônatas quer saber se está no inventário dele.**

Saída

Deverá ser impresso **“item encontrado”** caso o item esteja no inventário ou **“voce ainda nao descobriu este item”**, caso contrário.

Entrada	Saída
Storm Ruler Scholar's Candlestick Broadsword Anri's Straight Sword Sunlight Straight Sword Gael's Greatsword Ringed Knight Paired Greatswords fim Carthus Shotel	voce ainda nao descobriu este item
Frayed Blade Follower Sabre Sellsword Twinblades Harald Curved Greatsword Thrall Axe Butcher Knife fim Sellsword Twinblades	item encontrado

Vamos jogar um jogo

Autor: Danilo de A. Peleteiro



E agora?!?! Você foi capturado por Jigsaw em mais um de seus planos contra aqueles que ele julga não valorizar a própria vida! Para provar que Jigsaw está enganado, você deverá resolver um de seus peculiares enigmas e assim garantir a sua liberdade (talvez). Você encontrou uma gravação que explica passo a passo o que deverá ser feito. Em sua sala, haverá um papel com uma frase acompanhada de um número e uma palavra. O que Jigsaw deseja é muito simples: **que você diga se a quantidade de ocorrências daquela palavra na frase é a mesma da escrita no papel**. Caso seja, você deverá dizer “**SIM!**”, do contrário, deverá falar “**NAO!**”. Um detalhe essencial é que **todas as letras são minúsculas** e **Jigsaw ignora espaços em branco na frase no momento de contar as ocorrências da palavra**.

Tarefa

Para sua sorte, você encontrou um computador velho na sala onde está, e como é conhecido por ser viciado em programar, decidiu desenvolver um programa que o auxiliasse (e, quem sabe, outros futuros prisioneiros) nesse enigma. Portanto, você deverá computar a frase solicitada por Jigsaw e posteriormente avaliar se existe a quantidade ‘**Q**’ de ocorrências de uma dada palavra **P**.

Entrada

A primeira linha da entrada consiste de uma string **S**, que indica a frase a ser avaliada. A segunda linha contém um inteiro ‘**Q**’ ($1 \leq Q \leq 30$), informando a quantidade de ocorrências, seguido de uma palavra **P**, que indica o que deve ser detectado na frase **S**.

Saída

Seu programa deverá imprimir a quantidade de ocorrências de **P** em uma linha. Na outra, deverá imprimir “**SIM!**” caso essa quantidade seja igual à ‘**Q**’ e, caso contrário, deverá imprimir “**NAO!**”.

Entrada	Saída
eu quero jogar um jogo jogando limpo 3 jog	3 SIM!
xhuisyd xnzyxe nxnzzz zx x ify zzuzzzz z zjx 4 zz	6 NAO!
eu adoro sao joao e eu amo suas comidas 3 ua	3 SIM!

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 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

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.

							H
H			H		H		
					X		
	3		H		2		
						H	
		X			H		
	H						
		1					

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