

1. Um campo agrícola possui setores organizados de forma matricial. Cada setor é um de 3 tipos: terreno infértil (0), terreno fértil (1) e irrigador (2). Os irrigadores **nunca** estão localizados na borda da matriz e irrigam cada um dos 4 setores vizinhos (norte, sul, leste, oeste). Vide exemplo a seguir:

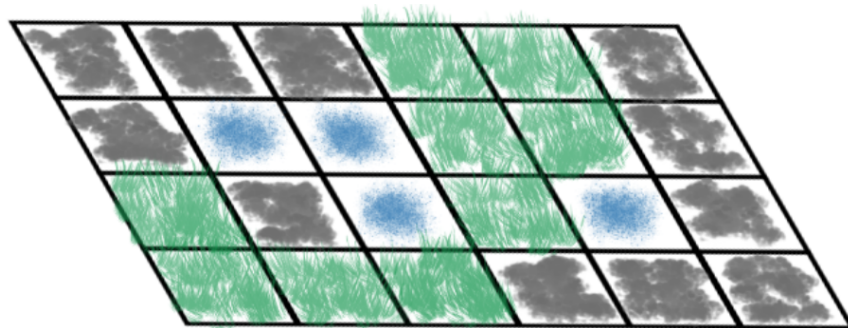


Figure 1: Imagem do exemplo 1

Implemente um programa que leia um inteiro M , um inteiro N ($M \leq 10$, $N \leq 10$), seguidos de $M \times N$ inteiros. Depois o programa deve escrever quantos **setores férteis** estão cobertos por pelo menos um irrigador e quantos não estão. Um 0 representa um setor infértil, 1 um setor fértil e 2 um setor ocupado por um irrigador. O setor onde está localizado o próprio irrigador não deve ser contabilizado.

Exemplo:

Input	Output
4 6 0 0 0 1 1 0 0 2 2 1 1 0 1 0 2 1 2 0 1 1 1 0 0 0	4 5

2. Você decidiu que está na hora de evoluir o jogo do campo minado para a segunda dimensão! A entrada do programa consiste em um inteiro n (assuma $3 \leq n \leq 20$), seguido de n palavras contendo entre 3 e 20 caracteres. Depois o programa deve ler dois inteiros, representando uma linha e uma coluna (começando de 0), e escrever na tela **bum!** caso nessa coordenada haja uma bomba ou escrever na tela um inteiro representando quantas bombas há na adjacência (incluindo diagonal) da coordenada em questão. Assuma que todas as palavras possuem mesmo tamanho.

Exemplos:

Input	Output
3 ..XX..X..X.. .XXX.....X. ..XX.XXXX. . . 0 0 4 ..XX..X..X.. .XXXX.....X. ..XX.XXXX. . . xxx.x..x.x.x 2 4 4 ..XX..X..X.. .XXXX.....X. ..XX.XXXX. . . xxx.x..x.x.x 2 5	1 5 bum!

3. Um campeonato de futebol conta com M times, numerados de 0 a $M - 1$. Cada time enfrenta cada um dos outros times uma única vez. Os resultados são armazenados em uma matriz, onde o elemento a_{ij} ($0 \leq i, j \leq M - 1$) representa quantos gols o time i fez contra o time j . Portanto, o resultado da partida $i \times j$ (ou $j \times i$, tanto faz pois é uma única partida) é a_{ij} gols para i e a_{ji} gols para j . No exemplo abaixo, o resultado do time 3 contra o time 0 foi a_{30} para o time 3 e a_{03} para o time 0, ou seja, 3×1 .

Implemente um programa que leia um inteiro M ($M \leq 20$), seguidos de $M \times M$ inteiros. Como um time não enfrenta ele próprio, assuma que os elementos da diagonal são sempre 0. O programa deve escrever **quantas** partidas terminaram **empatadas**.

Exemplo:

Input	Output
6 0 2 2 1 3 2 0 0 3 3 5 2 3 2 0 1 3 1 3 2 2 0 0 6 3 6 0 0 0 2 3 3 3 0 2 0	3

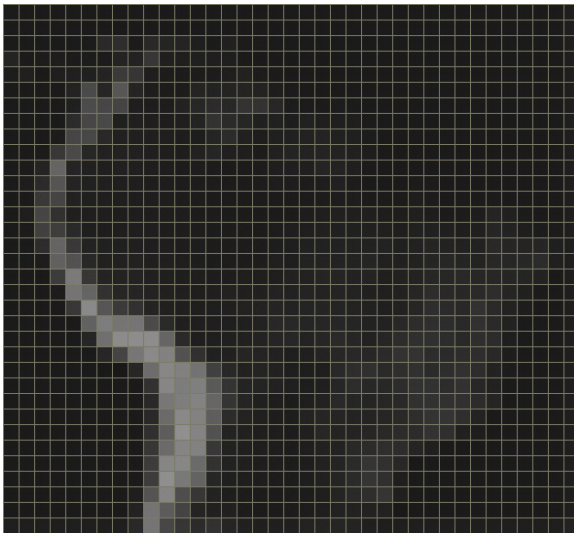
4. \triangleright Um quadrado metade-mágico é uma matriz quadrada $n \times n$ em que a soma de cada **coluna** resulta no mesmo valor. Escreva um programa em C que leia o valor de n (assuma $1 \leq n \leq 9$), seguido de $n \times n$ inteiros, **todos maiores que 0**. Depois o programa deve escrever **S** caso a matriz represente um quadrado metade-mágico e **N** caso contrário.

Pare a análise tão logo descobrir que não se trata de um quadrado mágico.

Exemplo:

Input	Output
3	S
2 7 1	
9 3 6	
4 5 8	

5. ▷ Escreva um programa em C para ler dois inteiros (**m** e **n**) (assuma $3 \leq m, n \leq 20$) e que leia em seguida $m \times n$ inteiros representando uma matriz. Cada elemento da matriz representa a altura de uma determinada região do mapa (observe a Cordilheira dos Andes na imagem da América do Sul a seguir). O programa deve escrever na tela as coordenadas de todos os picos do mapa, um por linha e no formato (x, y), conforme exemplo abaixo. Existe um pico em uma coordenada quando a altura nessa coordenada é maior que a altura de cada uma das 4 coordenadas adjacentes (não inclui diagonais). A ordem deve ser da esquerda para direita e de cima para baixo. Você não precisa reportar os picos localizados nas bordas!



Exemplo:

Input	Output
3 6	(1, 1)
1 2 3 3 1 2	(1, 4)
5 9 0 1 6 1	
2 4 6 2 5 2	

6. Escreva um programa em C para ler dois inteiros (**m** e **n**) (assuma $0 < m, n \leq 20$) e que leia em seguida $m \times n$ inteiros representando uma matriz. O programa deve então escrever na tela a transposta dessa matriz ($a_{ij} = a_{ji}$). Não é necessário armazenar a matriz transposta, apenas escrevê-la!

Exemplo:

Input	Output
3 4	1 5 2
1 2 3 4	2 6 4
5 6 7 8	3 7 6
2 4 6 8	4 8 8

7. \triangleright Você foi contratado para ajudar na implementação de uma rede social que conta com M usuários cadastrados. Um recurso que você quer implementar é a sugestão de amigos. Um usuário B deverá ser sugerido para A se eles não forem amigos, mas ambos possuírem pelo menos um amigo em comum.

As amizades estão armazenadas em uma matriz $M \times M$ de inteiros, onde o valor a_{ij} é igual a 1 se o usuário i ($0 \leq i \leq M - 1$) for amigo do usuário j ($0 \leq j \leq M - 1$) e 0 caso contrário. Assuma que a matriz sempre é simétrica, então $a_{ij} = a_{ji}$. Além disso não é possível ser amigo de si, então $a_{ii} = 0$.

Implemente um programa que leia um inteiro M ($M \leq 100$), seguidos de $M \times M$ inteiros e um inteiro x . O programa deve escrever todas as sugestões de amizades para o usuário x em ordem numérica crescente.

Input	Output
6	1 4
0 0 0 1 0 1	
0 0 0 1 1 0	
0 0 0 0 1 0	
1 1 0 0 1 1	
0 1 1 1 0 1	
1 0 0 1 1 0	
0	

8. Você está planejando uma viagem (somente ida) de uma cidade X para uma cidade Z . Para isso você está pesquisando passagens de avião que sejam as mais baratas possível, nem que para isso seja necessário passar por uma cidade Y .

Foi possível coletar preços de passagens com diversas origens e destinos que foram armazenados em uma matriz $M \times M$ de inteiros, onde o valor a_{ij} representa o preço em reais para ir da cidade i para a cidade j .

Implemente um programa que leia um inteiro M ($M \leq 10$), seguidos de $M \times M$ inteiros, um inteiro X e um inteiro Z . O programa deve escrever o custo da viagem de X para Z , passando por **no máximo** 1 cidade intermediária, com menor custo total. Esse custo deve ser precedido pelas cidades separadas por traço (vide exemplo abaixo).

Exemplos

Input	Output
3	0-1-2 R\$7
0 5 9	
0 0 2	
0 0 0	
0 2	
3	0-2 R\$9
0 5 9	
0 0 5	
0 0 0	
0 2	