

# Problema 1

## Confederação

*Nome do arquivo fonte:* confederacao.c, confederacao.cpp, confederacao.py ou confederacao.java

A Confederação Galática resolveu fazer uma reforma administrativa, para melhor distribuir os recursos de sua frota. Para isso, ela dividiu todo o espaço em regiões. Para definir as regiões, inicialmente um conjunto de planos infinitos foi especificado, e as regiões foram definidas pelos cortes desses planos. Note que algumas regiões são ilimitadas, mas que também podem existir regiões limitadas. O conjunto de planos foi escolhido de tal maneira que nenhum dos planos intercepta a órbita de um planeta, e portanto cada planeta transita por apenas uma região durante sua órbita (ou seja, um planeta dentro de uma região nunca cruzará um plano para outra região).

Sua tarefa consiste em determinar, dadas as equações dos planos e as posições dos planetas, quantos planetas existem na região com o maior número de planetas (em outras palavras, qual o número máximo de planetas dentro de uma região).

### Entrada

A primeira linha da entrada contém dois inteiros  $M$  ( $1 \leq M \leq 500$ ) e  $N$  ( $1 \leq N \leq 10000$ ), indicando respectivamente o número de planos e número de planetas. As  $M$  linhas seguintes contém cada uma quatro inteiros  $A, B, C$  e  $D$  ( $-10000 \leq A, B, C, D \leq 10000$ ), os coeficientes e o termo livre da equação  $Ax + By + Cz = D$  que define cada um dos planos. A seguir, cada uma das  $N$  linhas seguintes contém três inteiros  $X, Y$  e  $Z$  ( $-10000 \leq X, Y, Z \leq 10000$ ), indicando a posição  $(X, Y, Z)$  de um planeta.

### Saída

Seu programa deve produzir uma única linha contendo apenas um número inteiro, o número de planetas na região que contém o maior número de planetas.

### Exemplos

Entrada	Saída
2 5 1 0 0 1 2 0 0 8 0 1 0 2 2 2 3 3 3 5 5 5 2 18 4	3

Entrada	Saída
4 8 0 0 1 1 1 0 1 2 -1 1 1 3 -1 -1 1 3 0 0 5 0 0 4 0 0 -2 1 0 5 40 19 104 13 26 84 89 -45 18 3 1 0	5

## Problema 2

### ECOLOGIA

*Nome do arquivo fonte:* ecologia.c, ecologia.cpp, ecologia.py ou ecologia.java

O reino da Poliminogônia passou recentemente uma lei ecológica que obriga todas as fazendas a preservar o máximo de árvores possível em uma porcentagem fixa da área da fazenda. Além disso, para que os animais silvestres possam se movimentar livremente, a área preservada deve ser conexa.

As fazendas na Poliminogônia são sempre um reticulado de  $N \times N$  quadrados de um hectare cada. A figura ao lado ilustra uma fazenda com  $N = 5$ . A área preservada deve cobrir exatamente  $M$  quadrados. No exemplo da figura,  $M = 6$ . Ela deve ser conexa ortogonalmente; quer dizer, tem que ser possível se movimentar entre quaisquer dois quadrados preservados apenas com movimentos ortogonais entre quadrados preservados. A área não preservada, entretanto, pode ser desconexa.

31	12	7	1	14
23	98	3	87	1
5	31	8	2	99
12	3	42	17	88
120	2	7	5	7

Os fazendeiros sabem o número de árvores que há dentro de cada quadrado e você deve escrever um programa que calcule o número máximo possível de árvores que podem ser preservadas com uma área de  $M$  quadrados. No exemplo, é possível preservar 377 árvores!

#### Entrada

A primeira linha da entrada contém dois inteiros  $N$  e  $M$  ( $2 \leq N \leq 50$ ,  $1 \leq M \leq 10$ ). As  $N$  linhas seguintes contêm, cada uma,  $N$  inteiros de valor entre 1 e 1000, representando o número de árvores dentro de cada quadrado da fazenda.

#### Saída

Seu programa deve imprimir uma linha contendo um número inteiro, o número máximo de árvores que podem ser preservadas, com as restrições dadas.

#### Exemplos

Entrada	Saída
5 6 31 12 7 1 14 23 98 3 87 1 5 31 8 2 99 12 3 42 17 88 120 2 7 5 7	377

Entrada	Saída
4 8 1 1 1 1 9 9 9 1 9 1 9 1 9 9 9 1	72

## Problema 3

### HANDEBOL

*Nome do arquivo fonte:* handebol.c, handebol.cpp, handebol.py ou handebol.java

Frustrado e desanimado com os resultados de sua equipe de futebol, o Super Brasileiro Clube (SBC) resolveu investir na equipe de handebol. Para melhor avaliar os atletas, os técnicos identificaram que seria útil analisar a regularidade dos jogadores. Especificamente, eles estão interessados em saber quantos jogadores fizeram gols em todas as partidas.

Como o volume de dados é muito grande, eles gostariam de ter um programa de computador para realizar essa contagem.

#### Entrada

A primeira linha da entrada contém dois inteiros  $N$  e  $M$  ( $1 \leq N \leq 100$  e  $1 \leq M \leq 100$ ), indicando respectivamente o número de jogadores e o número de partidas. Cada uma das  $N$  linhas seguintes descreve o desempenho de um jogador: a  $i$ -ésima linha contém  $M$  inteiros  $X_j$  ( $0 \leq X_j \leq 100$ , para  $1 \leq j \leq M$ ), informando o número de gols do  $i$ -ésimo jogador em cada partida.

#### Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o número de jogadores que fizeram gols em todas as partidas.

## Exemplos

Entrada	Saída
5 3 0 0 0 1 0 5 0 0 0 0 1 2 1 1 0	0

Entrada	Saída
12 5 4 4 2 3 7 0 0 0 1 0 7 4 7 0 6 1 2 3 3 2 0 0 0 0 0 4 0 9 10 10 0 1 0 0 0 1 2 0 2 3 10 10 10 1 0 0 3 3 3 4 10 10 0 10 10 1 1 2 0 9	2