

17^o Treino para alunos da UFPR

Julho de 2015

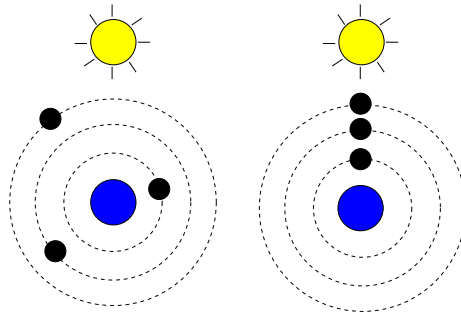


Flávio Zavan
Ricardo Oliveira

A: Alinhamento Lunar

Arquivo: alinhamento.[c|cpp|java|pas]

Em uma galáxia muito, muito distante, existe o planeta Nlôguerrà, habitado predominantemente por dinossauros. Nlôguerrà é orbitado por três luas. A órbita de cada lua tem a forma de uma circunferência cujo centro é Nlôguerrà, como indica a figura abaixo, à esquerda.



Sempre que as três luas se alinham e ficam entre o planeta e o sol, como mostra a figura acima, à direita, uma catástrofe terrível acontece! Na última vez que isto ocorreu, há exatamente M anos, uma grande seca se instaurou em todo o planeta, reduzindo sua população de dinossauros pela metade. A primeira lua leva L_1 anos terrestres para completar uma volta ao redor do planeta, enquanto a segunda leva L_2 anos e a terceira leva L_3 anos. Determine quantos anos irão se passar até o próximo alinhamento lunar entre o planeta e o sol.

Entrada

A primeira linha da entrada contém o inteiro M ($1 \leq M \leq 10^9$), indicando há quantos anos ocorreu o último alinhamento. A segunda linha contém três inteiros L_1 , L_2 e L_3 ($1 \leq L_1, L_2, L_3 \leq 10^3$), o tempo levado, em anos, para as luas completarem uma volta. É garantido que não houve alinhamentos como especificado nos últimos $M - 1$ anos e que não há alinhamento este ano.

Saída

Imprima uma linha contendo um número X , indicando que o próximo alinhamento lunar entre o planeta e o sol ocorrerá daqui a X anos.

Exemplo de entrada	Exemplo de saída
2 1 2 3	4

Exemplo de entrada	Exemplo de saída
3 2 4 8	5

B: Pesquisa

Arquivo: `pesquisa.[c|cpp|java|pas]`

Todo turista que visita a cidade deve conhecer a Avenida Quatro de Maio. A avenida é famosa devido, dentre outros motivos, à *imensa* quantidade de pessoas que moram nela.

Há N casas na avenida, numeradas sequencialmente de 1 a N . Há P_i pessoas morando na casa de número i da avenida.

O instituto de estatística está realizando o censo deste ano. Para tal, o instituto irá realizar diversas consultas sobre a população da avenida. Em cada consulta, são dados dois números A e B , com $A < B$. Para cada consulta, é necessário determinar o número total de pessoas que moram entre as casas A e B , inclusive.

Sua tarefa é, dada a população de cada casa e as consultas que o instituto irá fazer, determinar a resposta para cada consulta dada.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 10^5$). A segunda linha contém N inteiros $P_1, P_2, \dots, P_N$, indicando a população de cada casa da avenida (Para cada $1 \leq i \leq N$, $0 \leq P_i \leq 100$). A terceira linha contém um inteiro Q ($1 \leq Q \leq 10^4$), indicando o número de consultas a serem realizadas. Por fim, cada uma das próximas Q linhas descreve uma consulta. Cada linha contém dois inteiros A e B ($1 \leq A \leq B \leq N$).

Saída

Para cada consulta, imprima uma linha contendo sua resposta.

Exemplo de entrada	Exemplo de saída
4 10 2 8 72 3 4 4 1 3 2 4	72 20 82

Exemplo de entrada	Exemplo de saída
5 1 2 3 4 5 2 1 5 1 1	15 1

C: Cheddar

Arquivo: `cheddar.[c|cpp|java|pas]`

Pobre Petya está com fome.

– Mãe! Mãe! Mããããe! – Parte o grito do porão.

– Que foi, piá? – A doce senhora responde.

– Quero cheddar!

CHEDDAR! O menino está com fome, quer cheddar. Você, vizinho de Petya e excelente programador, ouviu os gritos e está disposto a ajudar. Como a polícia do queijo não permite a venda de cheddar para qualquer um, você fará um programa para facilitar as compras da querida mãe.

Todos os candidatos à compra de cheddar devem passar por um desafio de sorte. Eles recebem N ($0 < N \leq 1000$) dados (honestos, com 6 lados e valores de 1 a 6) e um número V ($0 \leq V \leq 6000$). Todos os N dados são lançados e o valor S é calculado sendo a soma dos números sorteados. Se S for maior ou igual a V , a compra está permitida, caso contrário, o comprador volta para casa de mãos vazias.

Seu programa deve calcular, dado N e V , a probabilidade da mãe de Petya conseguir comprar cheddar.

Entrada

Uma única linha com dois número inteiros, N e V , respectivamente.

Saída

Imprima uma única linha com um único número com precisão de quatro casas decimais, a probabilidade da mãe de Petya conseguir comprar cheddar.

Exemplo de entrada	Exemplo de saída
1 6	0.1667

Exemplo de entrada	Exemplo de saída
1 5	0.3333

Exemplo de entrada	Exemplo de saída
3 18	0.0046

Exemplo de entrada	Exemplo de saída
3 12	0.3750

D: Final Mundial de 2015

Arquivo: `final.[c|cpp|java|pas]`

Preocupado com a atual situação de crise no transporte aéreo, o diretor regional do concurso do ICPC no Brasil já iniciou seus preparativos para fazer as reservas das passagens aéreas para as finais mundiais em Marrocos em 2015. O primeiro passo foi estudar a malha aérea disponível, em que cada voo tem um certo preço e liga duas cidades (estamos, na verdade, chamando de voo apenas um trecho *non stop* de um voo comercial). O objetivo do diretor é fazer várias consultas nesta malha de voos.

Em geral desejamos fazer voos sem escalas, mas estes podem ser muito caros. Para contornar este fato o diretor deseja permitir algumas escalas possíveis. Assim, ele ordenou as várias cidades da malha em sua ordem de preferência para fazer escala. Ou seja, a cidade de índice 1 é a que ele prefere fazer escala, seguida pela cidade 2, e assim por diante.

As consultas que o diretor fará são então do seguinte tipo: é dada a cidade de partida e de chegada e um número t de cidades em que o diretor permite que sejam feitas escalas. Seu programa deverá encontrar o custo de um voo de custo mínimo entre as cidades que faça, no máximo, escalas nestas cidades. Por exemplo, se $t = 1$ você deverá encontrar o custo de um voo de custo mínimo entre as duas cidades que seja ou *non stop* ou que faça uma escala na primeira cidade.

Entrada

A primeira linha consiste em dois inteiros n ($1 \leq n \leq 100$) e m ($1 \leq m \leq 100000$), indicando o número de cidades e o número de escalas. Nas m linhas seguintes temos três inteiros u , v e w ($1 \leq u, v \leq n$ e $0 \leq w \leq 100$) indicando que existe uma escala que vai de u para v com custo w . Em seguida um inteiro c ($1 \leq c \leq 10000$) indicando o número de consultas, e nas c linhas seguintes temos três inteiros o , d e t ($1 \leq o, d \leq n$ e $0 \leq t \leq n$) onde o é a cidade de origem, d é a cidade de destino e t indica que as cidades $1, 2, \dots, t$ podem ser usadas para escalas.

Saída

Para cada consulta, na ordem da entrada, você deve imprimir uma linha contendo o custo mínimo ou -1 caso não exista caminho entre as duas cidades obedecendo as restrições dadas.

Exemplo de entrada	Exemplo de saída
4 7 4 1 0 2 1 3 1 4 20 2 3 15 4 2 1 3 1 21 1 2 0 3 2 1 0 4 2 2 4 3 1	3 0 -1

Exemplo de entrada	Exemplo de saída
5 10 4 5 2 2 1 4 1 2 7 2 4 7 5 2 1 4 1 2 4 5 12 5 4 4 5 3 7 3 5 9 4 2 5 0 3 4 5 4 5 1 2 3 2	-1 13 2 -1

E: Tesouro

Arquivo: `tesouro.[c|cpp|java|pas]`

Vasya e Petya adoram caça ao tesouro. O processo de diversão da dupla é agilizado por um site dedicado ao esporte, lá os amigos têm acesso às coordenadas do tesouro, direções e dicas de como encontrá-los.

Tendo anos de experiência e mais de 8192 tesouros encontrados, Vasya decidiu não apenas procurá-los, mas também escondê-los. Ambicioso, ele quer gerar uma experiência a ser lembrada o resto da vida para quem for atrás deles. O rapaz decidiu juntar seu conhecimento de geometria com o seu hobby para gerar um desafio formidável.

Os caçadores receberão a descrição de dois retângulos R_0 e R_1 . O tesouro está escondido na região de intersecção de R_0 e R_1 .

Como nem tudo na vida é fácil, existem regras para que o tesouro de Vasya seja listado no site. A área de busca não pode exceder $10m^2$. Por isto, Vasya pediu a Petya que verificasse a qualidade dos desafios. O amigo deve dizer se a área de busca tem tamanho adequado, se não há intersecção, se a área é muito grande, se é apenas uma linha ou se é apenas um ponto.

Tendo se formado em culinária, Petya não é muito bom em geometria. Ele admite não saber resolver o problema de seu amigo e está pedindo sua ajuda.

Entrada

Duas linhas, cada descrevendo um retângulo através quatro inteiros, duas coordenadas (em metros), X_0 , Y_0 , X_1 e Y_1 ($0 \leq X_0 < X_1 \leq 1000$ e $0 \leq Y_0 < Y_1 \leq 1000$), dois vértices opostos.

Saída

Imprima uma única linha com o resultado da avaliação da área de busca: "adequada", se estiver adequada e não for apenas ponto nem linha, "linha", se for uma linha, "ponto", se for apenas um ponto, "inexistente", se não existir, "grande", se for grande de mais.

Exemplo de entrada	Exemplo de saída
0 0 40 40 39 0 40 40	grande

Exemplo de entrada	Exemplo de saída
0 0 40 40 39 39 40 40	adequada

F: Surf Aquático

Arquivo: surf.[c|cpp|java|pas]

Este mês a Nlogônia está sediando a copa mundial de Surf Aquático. Neste esporte, um competidor fica N minutos dentro de um aquário fazendo acrobacias com sua prancha, tentando impressionar os juízes.

Ao contrário do Surf comum, os juízes dão uma nota ao competidor a cada minuto. Assim, um competidor recebe uma sequência de N notas a cada performance.

Uma *bateria* (de notas) é uma sequência contínua de B notas. O *resultado* de uma bateria é dada pela soma de todas as notas nela, exceto pela maior e pela menor nota, que são descartadas. Assim, o resultado da bateria (2, 3, 7, 5), por exemplo, é 8.

O *score* final do competidor é dado pela soma dos resultados de todas as baterias de tamanho B em sua sequência de N notas. Assim, se as notas são indexadas de 1 a N , soma-se o resultado da bateria contendo as notas de 1 a B , da bateria contendo as notas de 2 a $B + 1$, etc., até a bateria contendo as notas de $N - B + 1$ a N .

Dada a sequência de notas de um competidor, determine seu *score*.

Entrada

A primeira linha contém dois inteiros N e B ($3 \leq B \leq N \leq 2 \times 10^5$). A segunda linha contém N inteiros n_i ($1 \leq n_i \leq 100$), indicando, em ordem, a sequência de notas do competidor.

Saída

Imprima uma linha contendo o *score* do competidor.

Exemplo de entrada	Exemplo de saída
4 4 2 3 7 5	8

Exemplo de entrada	Exemplo de saída
7 5 2 8 3 4 8 1 2	39

G: Pegman

Arquivo: `pegman.[c|cpp|java|pas]`

Ao usar o *Google Maps*, você já deve ter usado o *Pegman* – um boneco amarelo que ativa o *Street View* no ponto em que é colocado. Hoje, um usuário irá colocar o *Pegman* em uma matriz com R linhas e C colunas. Cada célula desta matriz pode estar em branco ou conter uma seta apontado para alguma das quatro direções (esquerda, direita, cima, baixo).

Se o *Pegman* é colocado em uma célula em branco, ele não se move. Entretanto, se o *Pegman* é colocado em uma célula contendo uma seta, ele muda sua direção para a direção da seta e começa a caminhar. Durante a caminhada, se o *Pegman* encontra uma célula em branco, ele a atravessa e continua caminhando na mesma direção em que está atualmente. Se o *Pegman* encontra uma célula com uma seta, ele muda sua direção para a direção da seta, e continua caminhando na nova direção.

Como você pode observar, o *Pegman* pode ficar caminhando infinitamente. Entretanto, ele também pode sair dos limites da matriz. Sua tarefa é determinar se é possível *alterar a direção* de algumas setas ¹ de tal forma que o *Pegman* nunca saia dos limites da matriz, *independentemente* da célula em que é colocado inicialmente. Se isto for possível, determine também o menor número de setas que necessitam ser alteradas.

Entrada

A primeira linha contém dois inteiros R e C ($1 \leq R, C \leq 100$). As próximas R linhas contém C caracteres cada, descrevendo a matriz. Os caracteres `<`, `>`, `^` (circunflexo), `v` e `.` indicam uma seta para esquerda, direita, cima, baixo e uma célula em branco, respectivamente.

Saída

Imprima uma linha contendo o menor número de setas a serem alteradas, ou a palavra `IMPOSSIBLE` se não for possível configurar a matriz como se pede.

Exemplo de entrada	Exemplo de saída
2 1 ^ ^	1

Exemplo de entrada	Exemplo de saída
2 2 >v ^<	0

¹Cada seta pode ter sua direção alterada para as outras três direções conhecidas

Exemplo de entrada	Exemplo de saída
3 3^. ...	IMPOSSIBLE

Exemplo de entrada	Exemplo de saída
1 1 .	0

H: High Five

Arquivo: `high.[c|cpp|java|pas]`

A seleção Nlogonense de Basquetebol está na grande final da Copa do Mundo de Basquete, e o jogo está prestes a começar! Neste momento, os N atletas da seleção se preparam para entrar em quadra.

Os atletas são numerados de 1 a N . Inicialmente, os atletas estão em uma fila na beira do campo. Os atletas entram no campo um por um, na ordem de seus números. Assim, o atleta 1 é o primeiro a sair da fila e entrar na quadra. Em seguida, o atleta 2 sai da fila e entra na quadra, e assim por diante.

Ao sair da fila, cada atleta cumprimenta todos os atletas que estão em sua frente na fila com um *high five*. Por exemplo, considere que $N = 4$ e inicialmente a fila é 3 1 2 4, onde o atleta 4 é o mais próximo da quadra. O atleta 1, ao ir para a quadra, faz um *high five* com os atletas 2 e 4. O atleta 2, por sua vez, faz um *high five* apenas com o atleta 4 ao sair da fila. O atleta 3 também faz um *high five* apenas com o atleta 4 (note que os atletas 1 e 2 já saíram da fila). Por fim, o último atleta não cumprimenta ninguém.

Sua tarefa é determinar o número total de *high fives* que serão feitos.

Entrada

A primeira linha contém o inteiro N ($1 \leq N \leq 10^5$). A segunda linha contém N inteiros, indicando a ordem em que os atletas estão inicialmente na fila. O último inteiro presente na linha indica o atleta mais próximo da quadra.

Saída

Imprima uma linha contendo o número total de *high fives* que serão feitos.

Exemplo de entrada	Exemplo de saída
4 3 1 2 4	4

Exemplo de entrada	Exemplo de saída
5 5 4 3 2 1	0

Exemplo de entrada	Exemplo de saída
5 1 2 3 4 5	10

I: Cacheiro Viajante

Arquivo: `cacheiro.[c|cpp|java|pas]`

Clóvis Eduardo é um cacheiro viajante, ou seja, ele viaja o mundo para praticar *geocaching*. *Geocaching* é um jogo mundial de GPS no qual os jogadores encontram caixas escondidas (conhecidas como *geocaches*) por outros jogadores fazendo uso das coordenadas terrestres.

Devido a sua grande reputação e experiência (mais de 32768 *geocaches* encontrados em todos os continentes), Clóvis Eduardo foi convidado pelo ministério de Lazer e Esporte da Nlogônia a ajudar a esconder o *geocache* mais importante da história do país, uma grande caixa de ouro maciço em forma de cavalo, em comemoração aos 50 anos da instauração da república na Nlogônia.

Entretanto, não basta apenas encontrar o local perfeito. Novos *geocaches* devem seguir as regras nacionais de colocação atualizadas este ano:

- Nenhum *geocache* novo pode ser colocado no mesmo lugar de outro já existente;
- A soma das distâncias aos K *geocaches* já existentes mais próximos deve ser igual ou superior a D ;
- Se existirem menos de K *geocaches*, deverão ser considerados todos os existentes na soma das distâncias.

A Nlogônia é representada por um plano e, atualmente, possui N *geocaches* ativos. O ministério de Lazer e Esporte escolheu C possíveis localizações para a colocação do *geocache* comemorativo e pediu para que Clóvis Eduardo os informasse quantas destas localizações são viáveis de acordo as regras nacionais de colocação.

Como Clóvis Eduardo é preguiçoso, ele pediu para que você escrevesse um programa para resolver o seu problema.

Entrada

A primeira linha contém quatro inteiros separados por espaços N , C ($1 \leq N, C \leq 10^4$), K ($1 \leq K \leq 8$) e D ($1 \leq D \leq 10^5$).

As próximas N linhas contém dois inteiros separados por um espaço, x_i e y_i ($0 \leq x_i, y_i \leq 10^4$), a posição (em metros a partir da origem) do i -ésimo *geocache* já existente na Nlogônia. Não há posições repetidas.

As próximas C linhas contém dois inteiros separados por um espaço, x_j e y_j ($0 \leq x_j, y_j \leq 10^4$), a j -ésima possível posição (em metros a partir da origem) do novo *geocache*.

Saída

Imprima exatamente uma linha com o número de posições válidas sugeridas pelo Ministério de Lazer e Esporte.

Exemplo de entrada	Exemplo de saída
10 3 3 10 0 0 10 10 0 10 5 9 15 18 3 10 2 2 2 4 3 8 4 5 1 1 0 0 100 100	1

J: Feira de Adoção

Arquivo: `adocao.[c|cpp|java|pas]`

Carmem Lucrécia é uma pessoa muito solitária, todo sábado de manhã ela passeia sozinha pelas ruas do centro da capital da Nlogônia. A ocasião sempre é aproveitada para pensar na vida e no quão frio os Nlogonenses são.

Desta vez a caminhada não foi como sempre, haviam carros de som e grandes faixas anunciando a XVIII Feira Nacial de Adoção de Filhotes da Nlogônia, evento o qual mudaria a vida de Carmem Lucrécia para sempre.

Decidida a encontrar um companheiro canino, Carmem Lucrécia logo entrou na feira e começou a avaliar as opções.

Há exatamente N filhotes para adoção. Entretanto, Carmem Lucrécia é muito chata e tem uma série de exigências para seu novo companheiro:

- O animal deve ser da espécie **cachorro**;
- O animal deve ter nome composto;
- Pelo menos uma das palavras do nome do animal deve começar com o mesmo caractere do nome de sua raça.

O atendente da feira ficou tão confuso com as exigências de Carmem Lucrécia e pediu para que você fizesse um programa para ajudá-lo.

Entrada

A primeira linha contém o inteiro N ($1 \leq N \leq 100$), o número de filhotes para adoção. As $4N$ linhas seguintes descrevem os filhotes.

Cada filhote é descrito por três linhas. A primeira linha contém a espécie do animal, a segunda, a raça e a terceira, o nome. Cada descrição é seguida de uma linha em branco.

Todas as linhas contém apenas letras minúsculas, número e espaços. Nenhuma linha tem mais de 100 caracteres. Nenhuma começa nem termina com espaços e não há espaços consecutivos.

Saída

Imprima uma linha contendo o número total de filhotes que cumprem as exigências de Carmem Lucrécia.

Exemplo de entrada	Exemplo de saída
7 lobo guara do mato robson cachorro schnauzer schnepps schnitzel cachorro vira latas valdecir cachorro vira latas wanderson luiz gato siames soraia maria cachorro pastor alemao luan paulo cachorro bulldog luciano fabio romario mateus astrogildo bruno bianca	3