Maratona de Programação da SBC 2023

Sub-Regional Brasil do ICPC

02 de Setembro de 2023

Caderno de Problemas

Informações Gerais

Este caderno contém 13 problemas; as páginas estão numeradas de 1 a 19, não contando esta página de rosto. Verifique se o caderno está completo.

Este conjunto de problemas também está sendo utilizado simultaneamente nas seguintes competições: Gran Premio de México 2023, Gran Premio de Centroamérica 2023, The 2023 ICPC Bolivia Preliminary Contest e Torneo Chileno de Programación 2023.

A) Sobre os nomes dos programas

1) Para soluções em C/C++ e Python, o nome do arquivo-fonte não é significativo, pode ser qualquer nome.
2) Se sua solução é em Java, ela deve ser chamada *codigo_de_problema*`.java` onde *codigo_de_problema* é a letra maiúscula que identifica o problema. Lembre que em Java o nome da classe principal deve ser igual ao nome do arquivo.
3) Se sua solução é em Kotlin, ela deve ser chamada *codigo_de_problema*`.kt` onde *codigo_de_problema* é a letra maiúscula que identifica o problema. Lembre que em Kotlin o nome da classe principal deve ser igual ao nome do arquivo.

B) Sobre a entrada

1) A entrada de seu programa deve ser lida da *entrada padrão*.
2) A entrada é composta de um único caso de teste, descrito em um número de linhas que depende do problema.
3) Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
4) Cada linha, incluindo a última, contém exatamente um caractere final-de-linha.
5) O final da entrada coincide com o final do arquivo.

C) Sobre a saída

1) A saída de seu programa deve ser escrita na *saída padrão*.
2) Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
3) Cada linha, incluindo a última, deve conter exatamente um caractere final-de-linha.

Promoção:

Sociedade Brasileira de Computação

v1.0

Problema A
Altura Mínima

Carlitos é um entusiasta de aventuras com um amor insaciável por parques de diversões. Apesar da sua paixão vibrante, Carlitos enfrenta um desafio único: a sua estatura limitada. Enquanto planeja ansiosamente sua aventura de fim de semana, ele percebe que suas limitações verticais podem atrapalhar sua experiência no parque de diversões. Não se trata apenas de escolher um parque; trata-se de encontrar um onde ele possa aproveitar a emoção dos brinquedos.

Imagine o caleidoscópio de cores, as risadas jubilosas e a adrenalina dos passeios. Carlitos sempre foi atraído pela energia dos parques de diversões. Com o fim de semana se aproximando, ele se debruça sobre os folhetos do parque, estudando os requisitos de altura de cada passeio. O objetivo dele é maximizar sua diversão, e é aí que você entra.

Sua tarefa é ajudar Carlitos a determinar o número de passeios que ele pode desfrutar em um parque específico. Considerando sua altura e os requisitos mínimos de altura de cada passeio, oriente-o a aproveitar ao máximo sua aventura no parque de diversões.

Entrada

A primeira linha contém dois números inteiros, N e H ($1 \leq N \leq 6$ e $90 \leq H \leq 200$), que representam a quantidade de brinquedos em um parque e a altura de Carlitos em centímetros, respectivamente.

Na segunda linha da entrada, serão fornecidas as alturas mínimas $A_1, \ldots, A_N$ ($90 \leq A_i \leq 200$) de cada um dos brinquedos do parque.

Saída

Seu programa deve imprimir uma única linha contendo a quantidade de brinquedos nos quais Carlitos pode ir, ou seja, a quantidade de brinquedos para os quais a altura de Carlitos é pelo menos tão grande quanto a altura mínima necessária.

Exemplo de entrada 1	**Exemplo de saída 1**
`1 100` `100`	`1`

Exemplo de entrada 2	**Exemplo de saída 2**
`6 120` `200 90 100 123 120 169`	`3`

Problema B
Baralho Embaralhado

Um embaralhamento justo é uma forma de rearranjar N cartas de um baralho disposto horizontalmente da esquerda para a direita. Nesse embaralhamento, as cartas são inicialmente dividas em duas partes contíguas de tamanhos possivelmente diferentes em que uma delas pode ter até mesmo zero cartas! Denotemos por L e R as partes da esquerda e direita, respectivamente. As cartas de L são, então, combinadas com as cartas de R, de tal forma que a ordem relativa entre as cartas de cada partição seja mantida.

Você é apresentado à uma disposição final das cartas e deve descobrir qual a quantidade mínima de embaralhamentos justos que devem ser realizados no baralho inicial para que ele chegue a este estado. Inicialmente, as cartas do baralho podem ser vistas como a sequencia `1 2 ... N`.

Por exemplo, começando com a sequência `1 2 3 4 5` e realizando um embaralhamento justo com $L =$ `1 2` e $R =$ `3 4 5`, podemos obter as seguintes permutações:

- `1 3 2 4 5`
- `1 3 4 2 5`
- `3 4 5 1 2`
- `1 2 3 4 5`
- etc

Cada uma das permutações acima representa um possível resultado do embaralhamento justo. Note que `1 3 2 5 4` não é uma embaralhamento possível pois as ordens relativas das cartas `4` e `5` de R não é preservada.

Assuma que o resultado do primeiro embaralhamento é `1 3 2 4 5`. Se fizermos um segundo embaralhamento justo nele, podemos particionar o baralho em $L =$ `1 3 2 4`, $R =$ `5`, e combinar ambas para obter a permutação `1 3 2 5 4`.

Entrada

A primeira linha contém um inteiro $N (1 \leq N \leq 10^6)$, o número de cartas no baralho. A segunda linha contém uma permutação de números inteiros de 1 a N descrevendo a disposição final das cartas.

Saída

Imprima um único inteiro K, que representa o menor número de embaralhamentos justos necessários para obter a permutação dada.

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

Exemplo de entrada 2	Exemplo de saída 2
10 1 6 5 2 10 3 4 8 7 9	3

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

Exemplo de entrada 4	Exemplo de saída 4
8 1 6 2 3 7 8 4 5	1

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

Problema C
Caminhada na Montanha

Finalmente acabaram as provas e é chegada a hora de dar uma pausa na trabalheira da faculdade para umas merecidas férias. Com suas malas arrumadas, você partiu em uma aventura para explorar uma belíssima montanha da sua região – um passeio que você tem sonhado fazer há anos e que finalmente se torna realidade.

Além de bela, a majestosa montanha é gigantesca e oferece diversas trilhas a seus visitantes. Contando com N marcos, cada um unicamente identificado por um número entre 1 e N a montanha possui $N - 1$ caminhos entre os marcos. Essas passagens garantem uma travessia tranquila de um marco a outro, de tal forma que toda a montanha está conectada.

A cada marco i está associado um valor v_i; esse número reflete o número de curtidas que uma foto tirada nele terá na sua rede social favorita. Quase explodindo de entusiasmo, você decidiu adicionar uma nova camada à sua jornada por meio do "desafio do hype", permitindo que seus seguidores vivenciem sua aventura na mesma ordem que você. Nesse desafio, seu objetivo é, no mínimo, incrível: tirar e postar fotos de tal maneira que cada nova foto postada terá mais curtidas que a anterior.

As regras do desafio essencialmente ditam o desenrolar da sua jornada da seguinte maneira:

1. Sua trilha começa no marco de índice 1.
2. Seguindo apenas os caminhos já disponíveis entre os marcos, você se move apenas em frente, não podendo passar por um marco mais de uma vez.
3. A cada marco visitado, você pode tirar uma foto e imediatamente postar em sua rede social, ou não tirar nenhuma foto.

Sendo uma pessoa muito sábia, você já reparou que há muitas possíveis rotas e resultados para o desafio. Especificamente, para cada marco i, você gostaria de determinar o maior número de fotos que podem ser postadas se você iniciar sua jornada no marco 1 e encerrar no marco i (sem necessariamente tirar uma foto no marco i). Lembre-se, uma foto só pode ser postada se ela tiver mais curtidas que a foto anterior!

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 10^5$), que representa o número de marcos na montanha. A segunda linha contém $N - 1$ inteiros $p_2, p_3, \ldots, p_N$ ($1 \leq p_i \leq N$), onde p_i indica que existe um caminho entre os marcos i e p_i. A terceira linha contém N inteiros $v_1, v_2, \ldots, v_N$ ($1 \leq v_i \leq 10^9$), onde v_i representa o número de curtidas da foto do i-ésimo marco.

Saída

Imprima uma única linha com $N - 1$ inteiros, onde o i-ésimo inteiro representa o maior número de fotos que você pode postar se você iniciar sua caminhada no marco 1 e terminar no marco $(i + 1)$.

Exemplo de entrada 1	**Exemplo de saída 1**

`5`
`1 1 3 3`
`5 7 7 6 8`
	`2 2 2 3`

Exemplo de entrada 2	**Exemplo de saída 2**

`5`
`3 1 3 1`
`5 4 7 6 5`
	`2 2 2 1`

Problema D

Desvio

Na cidade de Nlogonia, o prefeito finalmente vai cumprir sua promessa de campanha e irá repavimentar alguns trechos de ruas. Contudo, enquanto um trecho estiver sendo repavimentado, os carros não poderão usá-lo e portanto um desvio deve ser utilizado.

Cada trecho de rua conecta duas esquinas na cidade, tem comprimento positivo e pode ser percorrido em ambas as direções.

Um desvio é um caminho alternativo que pode servir como um substituto temporário para o trecho de rua em obras. Mais especificamente, se o trecho que conecta as esquinas U e V estiver interditado, o desvio deve ser uma sequência de trechos de ruas que começa em U, termina em V, e não usa o trecho que conecta U diretamente com V. O objetivo é encontrar o desvio mais curto para cada trecho, de forma a minimizar o impacto enquanto as obras estiverem sendo feitas.

Como Integrante do Centro de Pavimentação e Carros, você deve ajudar o prefeito a calcular qual é o comprimento do desvio mais curto, para cada trecho.

Entrada

A primeira linha contém dois inteiros, N e M ($1 \leq N \leq 300$), que representam, respectivamente, o número de esquinas e o número de trechos de ruas. Cada uma das M linhas seguintes contém três inteiros: U, V, e L ($1 \leq U \leq N$, $1 \leq V \leq N$, $U \neq V$, $1 \leq L \leq 10^6$), que representam um trecho de mão dupla de comprimento L que liga as esquinas U e V. Nenhum trecho de rua é representado mais de uma vez.

Saída

Imprima M linhas, onde cada linha contém um inteiro. O i-ésimo inteiro deve ser o comprimento do desvio mais curto para o i-ésimo trecho, ou -1 se não for possível fazer um desvio. A ordem dos trechos na saída deve ser a mesma ordem fornecida na entrada.

Exemplo de entrada 1	Exemplo de saída 1
4 5 1 2 4 1 3 8 2 3 4 4 1 2 3 4 3	9 5 9 11 10

Exemplo de entrada 2	Exemplo de saída 2
2 1 1 2 1	-1

Problema E
Extraindo Pólen

É chegada a primavera, dando início a mais uma temporada de trabalho intenso na Sociedade das aBelhas de Chapecó (SBC)! No jardim da SBC, N lindas flores floresceram, cada uma com uma certa quantidade de grãos de pólen, que serão coletados pelas árduas trabalhadoras da sociedade. Para manter o ambiente de trabalho seguro, a SBC tem regras muito estritas para a coleta do pólen, sendo elas:

1. Quando uma abelha visita uma flor, ela deve coletar uma quantidade de pólen igual à soma dos dígitos do total de pólen atualmente naquela flor. Por exemplo, se uma flor tem 123 grãos de pólen, a abelha que a visitar deve coletar $1 + 2 + 3 = 6$ grãos, deixando a flor com $123 - 6 = 117$ grãos. Se a flor tem 201 grãos, a abelha coletará $2 + 0 + 1 = 3$ grãos, deixando a flor com $201 - 3 = 198$ grãos de pólen.

2. Todas as abelhas devem formar uma fila no início do dia; aquela que estiver na primera posição da mesma deve coletar pólen de alguma flor com o maior total de pólen. Se a abelha visitar um flor com 0 grãos, ela coleta zero grãos de pólen. Após coletar o pólen de uma flor, a abelha encerra seu turno de trabalho e volta para a colmeia.

A abelha Gertrude achou essas regras muito estranhas e procurou a sua ajuda para saber quanto pólen ela irá coletar no seu turno. Gertrude tem uma visão incrível e descobriu que atualmente está na K-ésima posição da fila.

Entrada

A primeira linha contém dois inteiros N ($1 \leq N \leq 10^6$) e K ($1 \leq K \leq 10^9$), que representam o número de flores e a posição de Gertrude na fila, respectivamente. A segunda linha contém N inteiros, onde o i-ésimo inteiro F_i ($1 \leq F_i \leq 10^6$ para $1 \leq i \leq N$) denota a quantidade inicial de pólen da i-ésima flor.

Saída

Imprima um único inteiro Q, que representa a quantidade de pólen que será coletada por Gertrude.

Exemplo de entrada 1	Exemplo de saída 1
5 3 22 15 7 2 1	6

Explicação do exemplo 1:

A primeira abelha coletará o pólen da primeira flor, deixando-a com $22 - (2 + 2) = 18$ grãos restantes. A segunda abelha também coletará da primeira flor, deixando-a com $18 - (1 + 8) = 9$ grãos restante. Por fim, Gertrude, a terceira abelha da fila, irá coletar pólen da segunda flor, coletando um total de $1 + 5 = 6$ grãos, que será a resposta para este caso de teste.

Exemplo de entrada 2	Exemplo de saída 2
3 10 21 21 21	0

Exemplo de entrada 3	**Exemplo de saída 3**
`3 9` `21 21 21`	`9`

Problema F
Férias Cansativas

William está planejando suas próximas férias. Um tema recorrente em todas as suas férias é a necessidade de lidar com o cansaço. Alguns dias ele nem aproveita muito, pois depois de várias atividades o cansaço começa a ser maior do que ele consegue suportar.

Desta vez, William teve uma ideia. Ele vai estimar o impacto de cada uma das atividades de turismo em sua disposição. Ele notou que algumas das atividades de férias, como esportes e caminhadas, são cansativas, consumindo sua disposição, enquanto outras atividades, como peças de teatro e musicais, são revigorantes, recuperando sua disposição.

Mais precisamente, William começa com D unidades de disposição e separa suas atividades em dois grupos: C atividades cansativas e R atividades revigorantes. Cada atividade cansativa requer uma certa quantidade de disposição e, portanto, consome tal quantidade, quando realizada. Cada atividade revigorante lhe fornece uma certa quantidade de disposição, quando realizada. Além disso, ele ordena as atividades de cada um dos grupos segundo suas preferências, pois há atividades que ele está mais ansioso para realizar. Note que atividades dos dois conjuntos podem ser intercaladas, mas William nunca fará uma atividade de um grupo sem ter feito todas as anteriores, pois isto violaria suas preferências.

Ao longo de suas férias, ao decidir qual atividade realizar em seguida, ele escolherá a primeira atividade cansativa ainda não realizada, se tiver disposição suficiente para fazê-la. Caso contrário, ele realizará a próxima atividade revigorante ainda não realizada, se ainda houver, recuperando uma certa quantidade de disposição. Naturalmente, caso não haja mais atividades cansativas remanescentes em algum momento, ele poderá simplesmente realizar todas as atividades revigorantes restantes.

Agora, tendo em vista este processo, ele pediu sua ajuda para determinar quantas atividades (incluindo cansativas e revigorantes) ele conseguirá realizar.

Entrada

A primeira linha da entrada contém 3 inteiros, D, C e R, indicando respectivamente a quantidade de disposição inicial, o número de atividades cansativas e o número de atividades revigorantes ($1 \leq D \leq 10^5$, $1 \leq C \leq 10^4$ e $1 \leq R \leq 10^4$). Cada uma das C linhas seguintes contém um inteiro C_i ($1 \leq C_i \leq 10^5$ para $1 \leq i \leq C$), indicando o consumo de disposição para uma atividade cansativa, em ordem de preferência. Finalmente, cada uma das R linhas seguintes contém um inteiro R_i ($1 \leq R_i \leq 10^5$ para $1 \leq i \leq R$), indicando o retorno de disposição para uma atividade revigorante, em ordem de preferência.

Saída

Imprima uma única linha contendo um único inteiro, o número total de atividades (incluindo cansativas e revigorantes) que William conseguirá realizar.

Exemplo de entrada 1	Exemplo de saída 1
40 3 3 30 20 10 5 5 5	5

Exemplo de entrada 2	**Exemplo de saída 2**
40 2 2 60 80 5 10	2

Exemplo de entrada 3	**Exemplo de saída 3**
100 3 1 60 60 50 10	2

Problema G
Grande Tratado da Bytelândia

A Grande Guerra da Bytelândia chegou ao fim. Os reinos restantes agora estão discutindo o Tratado de Divisão, que dividirá todas as terras do mundo entre eles. Este tratado se refere não apenas ao mundo conhecido, mas também a quaisquer territórios ainda não descobertos ou habitados, incluindo terra ou mar. Podemos assumir que o mundo é um plano infinito.

Cada reino no continente da Bytelândia tem uma única capital, e o Tratado de Divisão será baseado em suas localizações: ele declara que cada pedaço de terra pertence ao reino cuja capital é a mais próxima em um voo de pássaro (ou em linha reta). Em outras palavras: onde quer que você esteja no mundo, se C é a capital mais próxima de você, você estará no território do reino de C. Se houver um empate entre as distâncias de duas ou mais capitais, esse lugar estará na fronteira entre seus reinos.

Sob este tratado, alguns reinos podem ficar cercados por outros, enquanto outros reinos podem ficar com território ilimitado. Por isso, alguns monarcas estão contestando o tratado. Para informar essa discussão, eles exigem sua ajuda. Dadas as coordenadas das localizações de cada capital no continente da Bytelândia, você deve descobrir quais reinos teriam territórios infinitos sob o Tratado de Divisão.

Entrada

A primeira linha da entrada contém um único inteiro N ($2 \leq N \leq 10^5$), o número de reinos. Cada reino é identificado por um número inteiro único entre 1 e N. Cada uma das N linhas seguintes contém dois inteiros X e Y ($0 \leq X, Y \leq 10^4$), as coordenadas 2D da localização da capital de um reino. As capitais são dadas em ordem crescente de identificador do reino, não há duas capitais com a mesma localização, e você pode assumir que toda capital tem tamanho insignificante.

Saída

Imprima uma única linha com uma lista de inteiros separados por espaço em ordem crescente: os identificadores dos reinos que teriam territórios infinitos sob o Tratado de Divisão descrito. É garantido que sempre haverá pelo menos um reino assim.

Exemplo de entrada 1	**Exemplo de saída 1**

`4`
`3 2`
`1 5`
`3 6`
`3 5`
	`1 2 3 4`

Exemplo de entrada 2	**Exemplo de saída 2**

`6`
`2 1`
`3 3`
`1 4`
`4 5`
`6 3`
`4 3`
	`1 3 4 5`

Problema H

Honroso Trabalhador

Rafael vive em uma sociedade igualitária ideal: todos os trabalhos oferecem um salário diário igual e você só precisa trabalhar em um emprego que seja de seu interesse. Infelizmente, nem essa utopia estava preparada para o fato de que Rafael não tem nenhuma habilidade.

Para compensar essa situação, Rafael irá trabalhar apenas em empreitadas, onde empregadores em potencial não terão tempo de reparar no quão pouco qualificado Rafael é para o trabalho. Mesmo assim, dada sua desqualificação para esses trabalhos, ele precisa comprar cartas de recomendação falsas para ser empregado.

Rafael tem N trabalhos à sua disposição nos próximos dias. O i-ésimo trabalho começa no dia ℓ_i, termina no final do dia r_i e paga exatamente S moedas de ouro por dia trabalhado. Rafael é realmente ruim de serviço, e não consegue trabalhar em dois empregos ao mesmo tempo; além disso, ele pode começar no emprego i apenas no dia ℓ_i mas, uma vez empregado, ele pode se demitir no final de cada dia, manténdo o dinheiro dos dias trabalhados (inclusive o último), sendo assim capaz de começar outro trabalho a partir do dia seguinte (mas não no mesmo dia de sua demissão). Rafael também sabe que ele precisa de c_i moedas de ouro para comprar a carta de recomendação falsa para o trabalho i. Ciente de suas inabilidades e da necessidade das cartas falsas, Rafael reservou um dinheiro e é sempre capaz de comprar quantas cartas forem necessárias, mesmo antes de começar em qualquer um dos N trabalhos.

Dadas as descrições dos trabalhos disponíveis, qual o lucro máximo que Rafael pode conseguir, levando em conta os custos para a compra das cartas falsas?

Entrada

A primeira linha contém dois inteiros N ($1 \leq N \leq 10^6$) e S ($1 \leq S \leq 10^9$), que representam o número de trabalhos disponíveis e o quanto cada um deles paga por dia trabalhado.

Cada uma das N linhas seguintes contém três inteiros: ℓ_i, r_i e c_i ($1 \leq \ell_i \leq r_i \leq 10^9$, $1 \leq c_i \leq 10^9$), que representam a data de início, a data de fim, e o custo de se comprar uma carta de recomendação falsa para o trabalho i, respectivamente.

Saída

Imprima um único inteiro que corresponde ao lucro máximo de Rafael após o fim de todos os trabalhos.

Exemplo de entrada 1	Exemplo de saída 1
3 3 1 5 10 2 10 4 5 15 1	37

Explicação do exemplo 1:

O ideal é começar o segundo trabalho e depois passar para o terceiro, totalizando $14 \times 3 - 4 - 1 = 37$ moedas de ouro.

Exemplo de entrada 2	Exemplo de saída 2

`3 5`
`1 1 3`
`2 3 4`
`3 3 1`
	`8`

Exemplo de entrada 3	Exemplo de saída 3
`1 1000` `1 1 654`	`346`

Explicação do exemplo 3:

Mesmo que ele tenha que tirar dinheiro de sua reserva para comprar a carta de recomendação falsa, ele ainda pode obter algum lucro.

Exemplo de entrada 4	Exemplo de saída 4
`1 5` `1 3 20`	`0`

Explicação do exemplo 4:

Não vale a pena gastar dinheiro com uma carta de recomendação falsa para este trabalho.

Problema I

Investigando Zeros e Uns

Você se encontra em um misterioso mundo binário, onde um vetor de N dígitos binários aguarda pelo seu exame minucioso. Cada dígito é zero ou um, criando um padrão único em toda a paisagem. Sua missão é descobrir os padrões ocultos deste reino binário, desvendando o significado de subvetores com um número ímpar de uns.

O vetor de dígitos é denotado como $b_1, b_2, \ldots, b_N$. Sua tarefa é embarcar em uma jornada para descobrir os subvetores enigmáticos – segmentos de dígitos consecutivos – e determinar a contagem de subvetores que abrigam um número ímpar de uns.

Ao percorrer essa paisagem binária, lembre-se de que um subvetor é definido por seus dígitos iniciais e finais. Por exemplo, na sequência $[b_1, b_2, b_3]$, os subvetores incluem $[b_1]$, $[b_2]$, $[b_3]$, $[b_1, b_2]$, $[b_2, b_3]$, e $[b_1, b_2, b_3]$.

Sua missão é projetar um algoritmo que determine o número total de subvetores contendo um número ímpar de uns nesta sequência binária. Não se esqueça de que a resposta pode não caber em um número inteiro de 32 bits.

Entrada

A primeira linha da entrada contém o inteiro N ($1 \leq N \leq 10^5$), representando o comprimento da sequência binária.

A segunda linha contém os N dígitos binários $b_1, b_2, \ldots, b_N$ ($b_i \in \{0, 1\}$), representando os elementos da sequência.

Saída

Seu programa deve imprimir uma linha contendo a quantidade de subvetores contendo uma quantidade ímpar de uns.

Exemplo de entrada 1	**Exemplo de saída 1**
`3` `0 1 0`	`4`

Exemplo de entrada 2	**Exemplo de saída 2**
`10` `1 0 0 1 1 0 1 1 1 0`	`30`

Problema J
Jogo de Vôlei

Está tudo pronto para a grande final do super voleibol nos Jogos Olímpicos, e a tensão é tamanha que pode ser cortada com uma faca! Ricardão, o treinador de um dos times finalistas, está cuidadosamente posicionando seus jogadores para ter uma vantagem estratégica. Porém, ainda resta uma preocupação crescente: a de que algumas regiões da quadra não estejam cobertas por seus jogadores, o que necessitaria de pulos laterais muito longos e cansativos para que um jogador chegasse na bola antes que ela atingisse o chão.

Vamos ajudar Ricardão nessa partida decisiva! Dados uma quadra e o posicionamento dos jogadores, o treinador gostaria de saber qual o maior pulo que algum jogador deve fazer para interceptar um ataque adversário, garantindo assim que nenhum ponto da quadra fique vulnerável. Times profissionais de super vôlei são muito bons, podendo facilmente determinar qual é o jogador mais próximo de onde a bola vai cair; apenas esse jogador tentará pular até lá, enquanto todos os outros ficarão imóveis.

A quadra é um retângulo $\mathcal{R}$ de lados paralelos aos eixos do plano e definido por seus quatro vértices; por outro lado, os jogadores são modelados como um conjunto de N pontos dentro de $\mathcal{R}$. Sua tarefa é determinar qual a menor distância d tal que, para qualquer ponto da quadra, exista pelo menos um jogador a distância no máximo d daquele ponto. Lembre-se: a borda da quadra também faz parte da quadra e deve ser coberta!

Entrada

As primeiras quatro linhas definem os quatro vértices de $\mathcal{R}$; ou seja, a i-ésima dessas linhas possui dois inteiros x_i e y_i ($-10^5 \leq x_i, y_i \leq 10^5$), que representam as coordenadas do i-ésimo vértice de $\mathcal{R}$. A quinta linha contém um único inteiro N ($1 \leq N \leq 10^5$), que representa o número de jogadores. As N linhas seguintes representam os jogadores, com a i-ésima linha tendo dois inteiros x_i e y_i ($-10^5 \leq x_i, y_i \leq 10^5$) que nos dão as coordenadas do i-ésimo jogador. É garantido que todo jogador está dentro da quadra.

Saída

Imprima um único número d, que representa o pulo máximo que deve ser dado por algum jogador para que o time cubra toda a quadra. A saída será considerada correta se estiver com erro absoluto ou relativo de no máximo 10^{-5} da resposta correta.

Exemplo de entrada 1	Exemplo de saída 1

`-1 -1`
`1 -1`
`1 1`
`-1 1`
`1`
`0 0`
	`1.41421356237`

Exemplo de entrada 2	**Exemplo de saída 2**

`1 -1`
`-1 3`
`1 3`
`-1 -1`
`3`
`0 0`
`1 3`
`-1 3`
	`1.666666666667`

Problema K
k pra Mais, k pra Menos

A vida daquele que estuda ciência da computação nem sempre é tão fácil quanto parece. Alguns dias você pode estar implementando um algoritmo revolucionário, mas em outros você acaba relendo o mesmo livro pela décima vez. Mas a todo momento estamos buscando a mesma coisa: otimizar e automatizar tarefas. Neste caso, um professor está precisando da sua ajuda para orientar seus alunos para a próxima prova. Na opinião do professor, não é fácil decidir quanto tempo eles devem passar estudando tópicos teóricos e quanto tempo eles devem passar implementando algoritmos.

Essa não é a primeira vez que o professor leciona essa matéria, então a quantidade de dados disponíveis é tão grande que ele foi capaz de criar dois polinômios para descrever o desempenho final de cada aluno. Se o aluno gastar x unidades do seu tempo estudando teoria, sua nota aumentará em $t(x)$. Se o aluno gastar x unidades do seu tempo implementando algoritmos, sua nota aumentará em $p(x)$. De tal modo que o aluno que gastar a mesma quantidade x de tempo em cada uma das áreas terá a nota total $t(x) + p(x)$.

Acontece que recentemente um dos estudantes vem se destacando de forma imprevisível. E ele não esconde sua técnica de ninguém: "eu estudo muito mais teoria do que prática!". O professor em questão acha que isso é uma grande mentira e, para confirmar sua suspeita, ele decidiu estimar as notas dos alunos se eles sempre estudassem mais teoria do que prática (ou mais prática do que teoria). Você pode computar o polinômio $q(x) = t(x + K) + p(x - K)$? Ele será capaz de descrever a nota de todos os alunos se eles mudarem sua estratégia de estudo.

Entrada

A entrada consiste em três linhas. Na primeira, estão dois inteiros: N, representando o grau dos polinômios t e p ($1 \leq N \leq 10^5$) e K ($-10^5 \leq K \leq 10^5$). Já a segunda linha contém os $N + 1$ coeficientes de t e a terceira linha contém os $N + 1$ coeficientes de p. Os coeficientes são dados em ordem crescente de grau, com o último coeficiente da linha correspondendo ao termo de grau N, sendo todos não negativos de valor no máximo 10^6.

Saída

Seu programa deve escrever $N + 1$ inteiros, os coeficientes do polinômio $q(x)$ em ordem crescente de grau, módulo 998244353.

Exemplo de entrada 1	Exemplo de saída 1

`1 2`
`1 2`
`0 1`
	`3 3`

Exemplo de entrada 2	Exemplo de saída 2

`2 0`
`1 2 3`
`4 5 6`
	`5 7 9`

Exemplo de entrada 3	Exemplo de saída 3

`2 -1`
`3 3 3`
`1 0 0`
	`4 998244350 3`

Problema L
Lexicograficamente Agradável

Na pitoresca vila da Lexiconia, viviam dois amigos, Lily e Ethan. Um dia, uma misteriosa carta chegou na casa dos amigos, fechada com uma encantadora insignia. Dentro do envelope, eles encontraram uma charada complexa demais até mesmo para as mentes mais sábias da vila.

Nessa charada, foram dados um inteiro K e uma string S contendo apenas letras minúsculas, que pode ser alterada de acordo com uma curiosa regra. A cada momento, os aldeões tem a liberdade de escolher um índice i e, magicamente, os caracteres S_i e S_{i+K} trocam de posição. A charada é resolvida quando a string lexicograficamente mínima, usando apenas operações que respeitem a regra imposta pela charada, for encontrada.

A vila ficou muito curiosa e empolgada com a charada. Lily e Ethan, sempre vorazes por aventura, decidiram mergulhar de cabeça nesse desafio. Porém, à medida que os dois observavam a string, eles notaram que, no caminho para o sucesso, haviam inúmeras trocas possíves.

Com a string da charada vívida em suas mentes, os amigos se perguntaram: como navegar nesse gigantesco mar de possibilidades e determinar a string lexicograficamente mínima? Cada troca realizada parecia como folhear um livro mágico, que revelaria novos segredos e mistérios.

O tempo passou e até agora Lily e Ethan não terminaram de resolver essa charada. Será que você consegue ajudá-los?

Entrada

A primeira linha da entrada contém a string S ($1 \le |S| \le 10^5$). A segunda linha da entrada contém o inteiro K ($1 \le K < |S|$).

Saída

Imprima uma única linha contendo a string lexicograficamente mínima possível de ser feita utilizando essas operações.

Exemplo de entrada 1	**Exemplo de saída 1**
`zaaab` `4`	`baaaz`

Exemplo de entrada 2	**Exemplo de saída 2**
`njoab` `2`	`banjo`

Problema M
Malha Aérea

No reino de Quadradônia, o monarca quer rever todas as tarifas aéreas. Para isso, pediu ao seu contador uma tabela com as propostas de novos preços.

Todavia, o monarca estudou no Instituto de Computação e Programação de Chapecó (ICPC) e tem conhecimento suficiente para exigir coerência na tabela. A tabela é *coerente* se nenhuma rota com escalas é mais barata do que o voo direto.

Verificada a coerência da tabela, o monarca gostaria de diminuir o número de voos diretos, sem contudo aumentar os custos das viagens.

Seu problema é verificar a coerência da tabela e, sendo esta coerente, informar ao monarca quantas voos diretos podem ser eliminados sem encarecer o custo de qualquer viagem.

Entrada

A primeira linha contém N ($1 \le N \le 100$), que é o número de cidades da Quadradônia servidas por voos. Existem então mais N linhas, $L_1, L_2, \ldots, L_N$. A linha L_i contém N inteiros, $C_{i1}, C_{i2}, \ldots, C_{iN}$, onde C_{ij} é o custo do voo direto entre as cidades i e j. O custo de ida e de volta entre duas cidades é sempre igual, ou seja, $C_{ij} = C_{ji}$, para todos os pares $\{i, j\}$ tais que $1 \le i \le N$ e $1 \le j \le N$. Quando $i = j$, $C_{ij} = 0$. Quando $i \ne j$, $1 \le C_{ij} \le 10^3$.

Saída

Imprima uma linha contendo um inteiro. Se a tabela for incoerente, o inteiro deve ser igual a `-1`. Se a tabela for coerente, o inteiro deve ser igual ao maior número de voos diretos que podem ser removidos sem aumento nos custos das viagens para os passageiros.

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

Exemplo de entrada 2	Exemplo de saída 2
3 0 2 2 2 0 2 2 2 0	0

Exemplo de entrada 3	Exemplo de saída 3
3 0 2 9 2 0 2 9 2 0	-1