

Maratona de Programação da SBC 2022

Sub-Regional Brasil do ICPC

08 de Outubro de 2022

Caderno de Problemas

Informações Gerais

Este caderno contém 14 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: Terceira Fecha Gran Premio de México 2022, Gran Premio de Centroamérica 2022, Torneo Argentino de Programación 2022 e The 2022 ICPC Bolivia Preliminary Contest.

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

Problema A

Achando os Monótonos Não-Triviais Maximais

Neste problema iremos lidar com sequências de caracteres, muitas vezes chamadas de *strings*. Uma sequência é *não-trivial* se ela possui ao menos dois elementos.

Dada uma sequência s , dizemos que um trecho s_i, \dots, s_j é *monótono* se todos seus caracteres são iguais, e dizemos que ela é *maximal* se este trecho não pode ser estendido à esquerda e nem à direita sem perder a monotonicidade.

Dada uma sequência composta apenas por caracteres “a” e “b”, determine quantos caracteres “a” ocorrem em trechos monótonos maximais não-triviais.

Entrada

A entrada é composta por duas linhas. A primeira linha contém um único inteiro N , satisfazendo $1 \leq N \leq 10^5$. A segunda linha contém uma string, com exatamente N caracteres, composta apenas pelos caracteres “a” e “b”.

Saída

A saída é composta por uma única linha contendo um inteiro correspondente à quantidade total de vezes que o caractere “a” ocorre em trechos monótonos maximais não-triviais.

Exemplo de entrada 1 7 abababa	Exemplo de saída 1 0
Exemplo de entrada 2 7 bababab	Exemplo de saída 2 0
Exemplo de entrada 3 10 aababaaabb	Exemplo de saída 3 5
Exemplo de entrada 4 10 bbaababaaa	Exemplo de saída 4 5

Problema B

Brincando com Pedras

Alice e Bob irão jogar um jogo com 3 pilhas de pedras. As jogadas são feitas alternadamente, e cada jogador deve escolher uma pilha que ainda tenha pedras e retirar um número positivo de pedras dela. Vence quem remover a última pedra da última pilha que ainda tenha pedras. Alice faz a primeira jogada.

A i -ésima pilha terá um número de pedras aleatório e uniformemente distribuído no intervalo $[L_i, R_i]$. Qual é a probabilidade de Alice ganhar assumindo que ambos jogam de maneira ótima?

Entrada

A entrada consiste de uma linha com 6 inteiros, $L_1, R_1, L_2, R_2, L_3, R_3$, nessa ordem. Para cada i , $1 \leq L_i \leq R_i \leq 10^9$.

Saída

Imprima um inteiro representando a probabilidade de Alice ganhar módulo $10^9 + 7$.

É possível mostrar que a resposta pode ser expressa como uma fração irredutível $\frac{p}{q}$, tal que p e q são inteiros e $q \not\equiv 0 \pmod{10^9 + 7}$, ou seja, estamos interessados no inteiro $p \times q^{-1} \pmod{10^9 + 7}$.

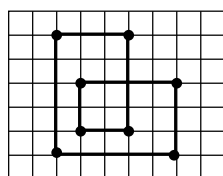
Exemplo de entrada 1 3 3 4 4 5 5	Exemplo de saída 1 1
Exemplo de entrada 2 4 4 8 8 12 12	Exemplo de saída 2 0
Exemplo de entrada 3 1 10 1 10 1 10	Exemplo de saída 3 580000005
Exemplo de entrada 4 5 15 2 9 35 42	Exemplo de saída 4 1

Problema C

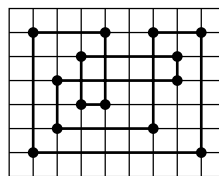
Cortes a Laser

Uma máquina de corte a laser para chapas de madeira possui uma cabeça de laser que pode movimentar-se apenas nas direções horizontal e vertical. Você foi contratado para compor a equipe de testes da máquina.

Um dos testes consiste em programar a máquina para realizar uma sequência não vazia de cortes consecutivos que inicia e termina no mesmo ponto. Cada corte na sequência, exceto o primeiro, inicia no ponto em que o corte anterior terminou. Nenhum corte toca a borda da chapa a ser cortada. As figuras (a) e (b) abaixo mostram dois exemplos de sequência de cortes, respectivamente com 8 e 14 cortes.

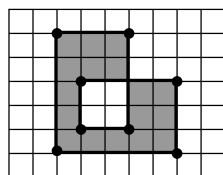


(a)

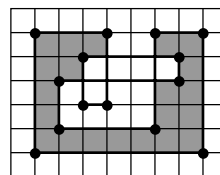


(b)

O seu chefe pediu que você determine a área do pedaço de maior área produzido pela sequência de cortes, desconsiderando o pedaço que contém as bordas da chapa cortada. Isto é, devem ser considerados apenas os pedaços contidos no polígono definido pelas linhas de corte. As figuras (c) e (d) abaixo mostram respectivamente os maiores pedaços produzidos pelos cortes das figuras (a) e (b).

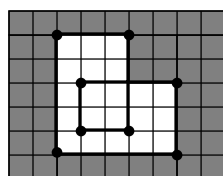


(c)

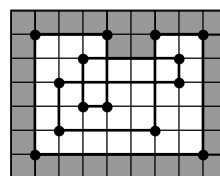


(d)

Para ilustrar, as figuras (e) e (f) abaixo mostram o pedaço descartado (que contém as bordas da chapa cortada) dos cortes das figuras (a) e (b), respectivamente.



(e)



(f)

Entrada

A primeira linha contém um inteiro N , o número de cortes da sequência ($4 \leq N \leq 10^4$). A segunda linha contém dois inteiros X_0 e Y_0 , a posição inicial da cabeça de laser na sequência de cortes ($1 \leq X_0 \leq 10^3$ e $1 \leq Y_0 \leq 10^3$). Cada uma das N linhas seguintes contém dois inteiros X_i e Y_i , a posição final do corte i ($1 \leq X_i \leq 10^3$ e $1 \leq Y_i \leq 10^3$, para $1 \leq i \leq N$, e $(X_N, Y_N) = (X_0, Y_0)$). Todas as posições dadas na entrada são distintas entre si, exceto a primeira e a última.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, a área do maior pedaço produzido pela sequência de cortes.

Exemplo de entrada 1 8 2 1 7 1 7 4 3 4 3 2 5 2 5 6 2 6 2 1	Exemplo de saída 1 17
Exemplo de entrada 2 14 1 1 8 1 8 6 6 6 6 2 2 2 2 4 7 4 7 5 3 5 3 3 4 3 4 6 1 6 1 1	Exemplo de saída 2 21

Problema D

Deslocando Partículas

Um quadrado possui seus vértices nas coordenadas $(0, 0)$, $(0, 2^N)$, $(2^N, 2^N)$, $(2^N, 0)$. Cada vértice possui um atrator. Uma partícula é posicionada inicialmente na posição $(2^{N-1}, 2^{N-1})$. Cada atrator pode ser ativado individualmente, qualquer quantidade de vezes. Quando um atrator na posição (i, j) é ativado, se a partícula encontra-se na posição (p, q) , ela desloca-se para o ponto médio entre (i, j) e (p, q) .

Dados N e um ponto (x, y) , determine o menor número de ativações necessárias para que a partícula seja posicionada na posição (x, y) .

Entrada

A entrada consiste de uma única linha contendo três inteiros N , x e y , satisfazendo $1 \leq N \leq 20$ e $0 < x, y < 2^N$.

Saída

Imprima uma única linha, contendo o menor número de ativações necessárias.

Exemplo de entrada 1 1 1 1	Exemplo de saída 1 0
Exemplo de entrada 2 4 12 4	Exemplo de saída 2 1
Exemplo de entrada 3 4 3 1	Exemplo de saída 3 3

Problema E

Estourando Balões

Após a cerimônia de encerramento da Maratona um número enorme de balões soltos estão flutuando no espaço do salão. O dono do salão está bravo porque no dia seguinte ocorre outro evento importante e os balões precisam ser removidos. Felizmente este ano Carlinhos veio preparado com seu arco e flecha para estourar os balões.

Felizmente também, devido ao fluxo do ar condicionado, os balões estão todos em um mesmo plano vertical (isto é, um plano paralelo a uma das paredes), embora em alturas e posições distintas.

Carlinhos vai atirar a partir do lado esquerdo do salão, a uma altura de sua escolha, em direção ao lado direito do salão. Cada flecha se move da esquerda para a direita, na altura em que foi lançada, no mesmo plano vertical dos balões. Ao encontrar um balão, este estoura e a flecha continua seu movimento para a direita, com a altura diminuída de 1. Ou seja, se a flecha estava a uma altura H , após atingir um balão ela continua na altura $H - 1$.

Carlinhos quer estourar todos os balões atirando o menor número possível de flechas. Você pode ajudá-lo?

Entrada

A primeira linha da entrada contém um inteiro N , o número de balões ($1 \leq N \leq 5 \times 10^5$). Como todos os balões estão num mesmo plano vertical, vamos definir que a *altura* de um balão é dada em relação ao eixo y e a *posição* de um balão é dada em relação ao eixo x desse plano. Os balões são numerados de 1 a N . Os números dos balões indicam as suas posições, da esquerda (balão número 1) para a direita (balão número N), independentemente das suas alturas. A posição do balão número i é diferente da posição do balão número $i + 1$, para todo i . A segunda linha contém N inteiros H_i , onde H_i indica a altura em que o balão número i está ($1 \leq H_i \leq 10^6$ para $1 \leq i \leq N$).

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o menor número de flechas que Carlinhos precisa atirar para estourar todos os balões.

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

Problema F

Forca Multidimensional

O Jogo da Forca Multidimensional tem regras muito peculiares. De certa forma, é como se você estivesse jogando várias partidas do tradicional Jogo da Forca ao mesmo tempo, com a diferença que as palavras não precisam existir no dicionário. Se você nunca jogou o Jogo da Forca, não se preocupe: toda a informação que você precisa estará abaixo.

Na versão multidimensional do jogo, existem várias palavras no tabuleiro, inicialmente desconhecidas, todas de mesmo comprimento. Em cada momento do jogo, você conhece alguns caracteres de certas posições das palavras (a maneira como estes caracteres foram descobertos não é importante para este problema). Em determinado momento, quando resta apenas um caractere desconhecido em cada palavra do tabuleiro, o jogo parte para a fase de *tudo ou nada*. Neste momento, você deve escolher uma palavra que maximize o número de compatibilidades com as palavras do tabuleiro. Para uma escolha de palavra P , dizemos que ela é compatível com uma palavra T do tabuleiro se todas as letras conhecidas de T ocorrem exatamente nas mesmas posições em P .

Dadas as informações conhecidas sobre as palavras do tabuleiro, você deve determinar qual palavra escolher para o *tudo ou nada*, que maximiza o número de compatibilidades. Se houver mais de uma solução, imprima a menor lexicograficamente. Dizemos que uma palavra P é lexicograficamente menor que uma palavra Q se $P_i < Q_i$ onde P_i é o i -ésimo caractere de P , Q_i é o i -ésimo caractere de Q e i é o menor índice tal que $P_i \neq Q_i$.

Entrada

A primeira linha da entrada contém dois inteiros N e C satisfazendo $1 \leq N \leq 10^4$ e $1 \leq C \leq 12$, indicando número de palavras no tabuleiro e comprimento das palavras que ele contém. Em seguida, N linhas descrevem, cada uma, uma palavra de comprimento C composta apenas de caracteres de “a” a “z” exceto por uma das posições, que conterá um caractere “*”, indicando que o caractere daquela posição ainda é desconhecido.

Saída

Você deve imprimir uma única linha, contendo, respectivamente, uma palavra T , de comprimento C , e um inteiro M , onde M é o maior número de compatibilidades que uma palavra pode ter com as palavras da entrada e T é a palavra lexicograficamente menor dentre aquelas com compatibilidade M .

Exemplo de entrada 1 5 4 rat* ru*d rot* r*ta r*ta	Exemplo de saída 1 rata 3
Exemplo de entrada 2 5 4 bon* fon* n*no *eto *ano	Exemplo de saída 2 nano 2

Problema G

Geometria Triangular

Todo polígono pode ser construído juntando-se triângulos. Em particular, podemos fazer isso iterativamente: começamos com um triângulo, adicionamos um segundo triângulo identificando um de seus lados a um dos lados do triângulo inicial, adicionamos um terceiro triângulo identificando um de seus lados a um dos lados livres de um dos triângulos originais e assim por diante. Consideraremos apenas polígonos que podem ser construídos dessa maneira, onde cada triângulo adicionado toca (e é identificado com) exatamente um lado de um triângulo posicionado anteriormente.

Dado um polígono P , seja T o conjunto de triângulos que foi utilizado para formá-lo. Os lados de cada triângulo são segmentos de reta. Seja L o conjunto de segmentos que são lados de algum triângulo em T . Note que cada elemento de L é um lado de um ou dois elementos de T .

Uma vez que tenhamos um polígono posicionado no plano, em alguns casos podemos remover alguns dos triângulos que o compõem, sem alterar o conjunto L . Queremos remover triângulos de forma que o conjunto L seja mantido e que a área total dos triângulos remanescentes seja mínima. Equivalentemente, queremos selecionar um subconjunto S de triângulos de T de forma que:

1. Todo elemento de L seja o lado de pelo menos um triângulo em S ; e
2. A soma das áreas dos elementos de S seja a menor possível.

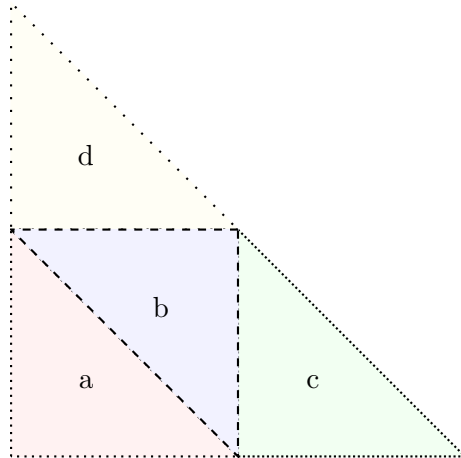
Entrada

A primeira linha da entrada contém um inteiro N , $1 \leq N \leq 10^5$ correspondendo ao número de triângulos na triangulação de P . Cada uma das N linhas seguintes contém 6 números, x_1, y_1, x_2, y_2, x_3 e y_3 , indicando a existência de um triângulo de coordenadas (x_1, y_1) , (x_2, y_2) e (x_3, y_3) . Os triângulos são dados em ordem arbitrária. Todas as coordenadas serão números inteiros de valor absoluto no máximo 10^6 .

Saída

Você deve imprimir a área mínima possível respeitando as condições do problema, com exatamente uma casa decimal.

Exemplo de entrada 1 4 0 0 0 10 10 0 10 10 0 10 10 0 10 10 0 10 0 20 10 10 20 0 10 0	Exemplo de saída 1 150.0
Exemplo de entrada 2 3 0 0 0 10 10 0 10 10 0 10 10 0 10 10 20 0 10 0	Exemplo de saída 2 150.0
Exemplo de entrada 3 1 0 0 1 0 0 1	Exemplo de saída 3 0.5



Na figura acima as triangulações $T_1 = \{a, b, c, d\}$ e $T_2 = \{a, b, c\}$ representam, respectivamente, o primeiro e o segundo exemplo. Note como $S_1 = \{a, c, d\}$ é um subconjunto válido para o primeiro caso. O triângulo b é deixado de fora, mas todos os seus lados estão presentes nos triângulos selecionados.

Problema H

Habilitando a Movimentação

O presidente da Nlogônia decidiu, por decreto, que todas as ruas da Nlogônia deveriam ser de mão única. Devido a certo desconhecimento de ciência básica, não houve um planejamento adequado para as mudanças. Depois que o novo sistema entrou em vigor, as pessoas não eram capazes de ir trabalhar, ou não conseguiam voltar para casa do trabalho, por exemplo. Como resultado, houve caos e rebeliões em muitas cidades.

O presidente sofreu impeachment e a nova administração do país contratou uma equipe de cientistas para resolver o problema. Por sua vez, essa equipe contratou você, especialista em complexidade de algoritmos, para ajudá-los no cálculo eficiente de soluções.

Assim, para cada cidade, você recebe os *pontos de referência* da cidade e as ruas de mão única, cada uma conectando dois pontos de referência. Sua tarefa é determinar o número mínimo de pontes de mão única que devem ser construídas para ter conectividade total na cidade. Cada ponte também deve conectar dois pontos de referência.

Entrada

A primeira linha da entrada contém dois inteiros, N e M ($1 \leq N \leq 10^4, 1 \leq M \leq 10^6$), onde N é o número de pontos de referência e M é o número de ruas. Cada uma das próximas M linhas contém dois inteiros, R e S , $1 \leq R, S \leq N$, $R \neq S$, que corresponde a uma rua que liga R a S , tal que todos os veículos daquela rua se deslocam de R em direção a S .

Saída

Seu programa deve imprimir uma única linha contendo o número mínimo de pontes necessárias para deixar os habitantes felizes.

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

Exemplo de entrada 3 2 1 1 2	Exemplo de saída 3 1
Exemplo de entrada 4 3 3 1 2 2 3 3 1	Exemplo de saída 4 0
Exemplo de entrada 5 2 0	Exemplo de saída 5 2
Exemplo de entrada 6 6 4 1 2 1 3 4 6 5 6	Exemplo de saída 6 3

Problema I

Interceptando Informações

A Spies Breaching Computers (SBC), uma agência privada de espões digitais, está desenvolvendo um novo dispositivo para interceptação de informações que, através de ondas eletromagnéticas, permite a espionagem mesmo sem contato físico com o alvo.

O dispositivo tenta coletar informações de um byte por vez, isto é, uma sequência de 8 bits onde cada um deles, naturalmente, pode ter valor 0 ou 1. Em determinadas situações, devido a interferências de outros dispositivos, a leitura não pode ser feita com sucesso. Neste caso, o dispositivo retorna o valor 9 para o bit correspondente, informando que não foi possível efetuar a leitura.

De forma a automatizar o reconhecimento das informações lidas, foi feita uma solicitação de um programa que, a partir das informações lidas pelo dispositivo, informe se todos os bits foram lidos com sucesso ou não. Sua tarefa é escrever este programa.

Entrada

A entrada consiste de uma única linha, contendo 8 números inteiros $N_1, N_2, N_3, N_4, N_5, N_6, N_7$ e N_8 , indicando os valores lidos pelo dispositivo (N_i é 0, 1 ou 9 para $1 \leq i \leq 8$).

Saída

Imprima uma única linha contendo a letra maiúscula “S” caso todos os bits sejam lidos com sucesso; caso contrário imprima uma única linha contendo a letra maiúscula “F”, correspondendo a uma falha.

Exemplo de entrada 1 0 0 1 1 0 1 0 1	Exemplo de saída 1 S
Exemplo de entrada 2 0 0 1 9 0 1 0 1	Exemplo de saída 2 F

Problema J

Jogando 23

Vinte e três é um jogo de cartas simples, jogado por crianças. Como o nome sugere, ele é uma variação do jogo vinte e um (*blackjack* em inglês), que é um dos jogos mais jogados em cassinos e sites de jogos.

O jogo utiliza um baralho de 52 cartas, com quatro naipes, cada naipe com 13 cartas (ás, 2, 3, 4, 5, 6, 7, 8, 9, 10, valete, dama e rei). Os naipes das cartas não são relevantes. As cartas com figuras (valete, dama e rei) valem dez pontos, as cartas com números valem o seu número em pontos (por exemplo, a carta 4 vale quatro pontos) e o ás vale um ponto.

Ganha o jogo o jogador que tiver o maior número de pontos, desde que não exceda 23. Se um jogador tem um número de pontos maior do que 23 dizemos que o jogador *estourou*.

As regras do jogo são simples: a cada partida, inicialmente o baralho é embaralhado, as cartas são colocadas em um monte e cada jogador recebe duas cartas do monte. Todas as cartas são distribuídas com a face para cima (todos os jogadores vêem as cartas de todos os jogadores). O passo seguinte, chamado de *rodada*, é repetido enquanto houver jogadores ativos: uma carta é retirada do monte e colocada na mesa com a face para cima. Essa carta, denominada *carta comum*, vale para todos os jogadores. Se um jogador estourar, ele sai do jogo. Vence a partida o jogador que numa determinada rodada somar 23 (somando suas duas cartas iniciais mais as cartas comuns), ou se o jogador for o único jogador ativo ao final da rodada. Note que pode haver mais de um vencedor (cuja cartas somam 23) e que pode não haver vencedor em uma partida.

João e Maria estão jogando vinte e três. Os dois são os únicos jogadores, nenhum dos dois estourou e nenhum dos dois tem 23 pontos. Além disso, a pontuação dos jogadores é tal que a próxima carta comum pode fazer com que a partida termine.

Dadas as cartas iniciais de João e Maria e as cartas comuns, determine qual é o valor da carta de menor valor que deve ser retirada do monte na próxima rodada para que Maria vença a partida.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 8$), o número de rodadas do jogo até o momento. Cada carta é descrita por um inteiro I ($1 \leq I \leq 13$). Note que as cartas com figuras (valete, dama e rei) são representadas na entrada pelos valores 11, 12 e 13 e não por quantos pontos elas valem. A segunda linha contém dois inteiros, descrevendo as duas cartas iniciais de João. A terceira linha contém dois inteiros, descrevendo as duas cartas iniciais de Maria. A quarta e última linha contém N inteiros, descrevendo as cartas comuns, na ordem em que são retiradas do monte.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o valor da carta de menor valor que deve ser retirada do monte na próxima rodada para Maria vencer a partida, ou -1 se não for possível Maria vencer a partida nessa próxima rodada.

Exemplo de entrada 1	Exemplo de saída 1
1 10 5 9 10 1	3

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

Problema K

Kalel, o Sapo Saltador

Kalel é um sapo que gosta de pular sobre pedras.

Há N pedras em uma linha, numeradas de 1 a N da esquerda para a direita. Kalel começa na pedra 1 e gostaria de chegar na pedra N .

Em cada movimento, Kalel pode escolher um dentre M tipos de pulo. O j -ésimo pulo permite que ele pule da pedra x até a pedra $x + d_j$, e custa p_j pontos de energia. Pode acontecer de p_j ser igual a 0 para alguns valores de j . Você pode assumir que Kalel nunca fica sem energia.

Dados N e K , calcule de quantas maneiras diferentes Kalel consegue chegar na pedra N gastando no máximo K pontos de energia no total. Duas maneiras são consideradas diferentes caso a sequência de escolha de pulos seja diferente. Como este número pode ser muito grande, estamos interessados apenas no resto da divisão do mesmo por 10^9 (um bilhão).

Entrada

A primeira linha contém três inteiros, N , M e K ($1 \leq N \leq 10^9$, $1 \leq M \leq 10^5$, $0 \leq K \leq 400$). As próximas M linhas contêm dois inteiros cada, os números d_j e p_j ($1 \leq d_j \leq 10$, $0 \leq p_j \leq K$).

Saída

Imprima uma única linha, contendo o número de maneiras diferentes de chegar na pedra N gastando no máximo K pontos de energia, módulo 10^9 (um bilhão).

Exemplo de entrada 1 5 3 10 1 3 2 0 3 1	Exemplo de saída 1 6
Exemplo de entrada 2 100000 3 10 1 9 2 0 7 3	Exemplo de saída 2 85449877

Problema L

Listando Caminhos Tediosos

Uma árvore é um grafo conexo (existe um caminho entre quaisquer dois de seus vértices), não direcionado (as arestas do grafo não possuem orientação), e acíclico (não possui ciclos).

Uma árvore colorida é aquela em que cada um de seus vértices tem uma determinada cor.

Um caminho tedioso é um caminho na árvore tal que tanto o vértice inicial quanto o vértice final tem a mesma cor, e não existe vértice ou aresta que apareça mais de uma vez no caminho. Note que as cores dos vértices intermediários do caminho, caso existam, não são relevantes.

Dada uma árvore colorida, com N vértices, sua tarefa é calcular, para cada aresta da árvore, a quantidade de caminhos tediosos que passam por aquela aresta.

Entrada

A primeira linha contém o número de vértices N ($1 \leq N \leq 10^5$). A segunda linha contém N inteiros C_1, \dots, C_N , onde C_i ($1 \leq C_i \leq N$) representa a cor do vértice i . Nas próximas $N - 1$ linhas temos dois inteiros cada, u e v , representando uma aresta ($1 \leq u, v \leq N$ e $u \neq v$). É garantido que o grafo dado será uma árvore.

Saída

Imprima $N - 1$ inteiros, representando o número de caminhos tediosos que passam por cada aresta, seguindo a mesma ordem em que as arestas aparecem na entrada.

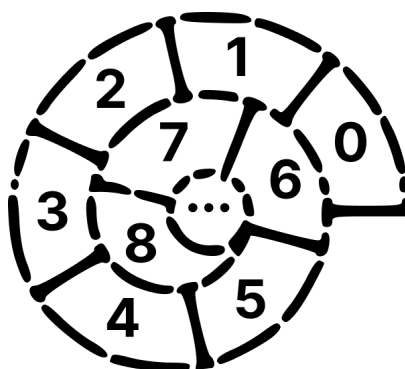
Exemplo de entrada 1 6 1 1 1 2 2 1 1 2 2 3 4 6 2 4 1 5	Exemplo de saída 1 4 3 3 4 1
Exemplo de entrada 2 12 1 2 3 1 2 2 1 3 2 3 1 2 1 2 2 3 2 4 4 5 4 6 1 7 7 8 7 9 9 10 6 11 6 12	Exemplo de saída 2 10 2 10 4 9 9 2 6 2 3 4

Problema M

Maratona de Amarelinha

Nesta data, 8 de outubro de 2022, ocorre o mais aclamado evento por estudantes de computação de todo o país. Não, não estamos falando da maratona de programação.

Estamos falando, é claro, da Amarelinha! Para quem não conhece, a Amarelinha é uma competição anual tradicionalmente organizada como evento paralelo da maratona de programação. Transmitida ao vivo para espectadores de todos os continentes, essa variante exótica do popular jogo infantil ocorre em um campo infinito, em formato de espiral, dividido em casas sequencialmente numeradas a partir de zero, conforme a ilustração abaixo.



Neste ano, a Amarelinha atraiu um número recorde de N participantes, numerados sequencialmente de 1 a N . Sabe-se que o i -ésimo participante inicia na casa de número A_i .

A Amarelinha consiste de Q rodadas. Na q -ésima rodada, o Carlão, fenomenal organizador da Amarelinha praticamente desde a sua concepção, informará dois números inteiros aos participantes: c_q e d_q . Essa é uma ordem para que todos os participantes de número i tal que i e c_q compartilhem um fator inteiro comum maior que 1 retrocedam d_q posições nas casas da Amarelinha, uma por uma, sem nunca retroceder além da casa 0. (Se um participante eventualmente retornar à casa 0, ele deve permanecer lá, ignorando quaisquer comandos adicionais de retrocesso, para não acabar saindo do campo.)

Assumindo que todos os participantes executaram as instruções perfeitamente (nenhum deles gostaria de decepcionar o Carlão), a sua tarefa é determinar, para cada um dos participantes, o número da rodada em que ele retorna à casa 0 (ou indicar que isso nunca acontece).

Entrada

A primeira linha da entrada contém os inteiros N e Q ($1 \leq N, Q \leq 10^5$). A segunda linha contém N inteiros, respectivamente, A_1, A_2, \dots, A_N ($1 \leq A_i \leq 10^9$). Cada uma das próximas Q linhas contém dois inteiros, c_q e d_q ($1 \leq c_q \leq 10^5, 1 \leq d_q \leq 10^9$).

Saída

A saída deverá conter exatamente N linhas. A i -ésima linha deverá conter um único inteiro, indicando o número da rodada em que o i -ésimo participante retorna à casa 0 (ou, se isso nunca acontece, o valor -1).

Exemplo de entrada 1 7 6 10 20 30 40 50 60 70 2 25 3 36 100 42 5 10 7 70 1 1000	Exemplo de saída 1 -1 1 2 3 4 2 5
Exemplo de entrada 2 6 4 100 100 100 100 100 100 2 50 3 50 5 99 5 1	Exemplo de saída 2 -1 -1 -1 -1 4 2

Problema N

Números Virados

Você acaba de ganhar um baralho com N cartas. Cada uma dessas cartas tem dois números escritos: um no lado da frente, e outro no lado de trás.

Seu amigo te desafiou para um jogo. Ele embaralhou as cartas do baralho e as colocou numa mesa. As cartas estão dispostas em uma linha, uma ao lado da outra, todas com a face da frente virada para cima.

Da esquerda para a direita, você sabe que o número escrito na face da frente da i -ésima carta é A_i , e que o número escrito na face de trás da i -ésima carta é B_i .

O jogo é dividido em duas partes.

Na primeira parte, você deve escolher K cartas do baralho. Para escolher uma carta você deve seleccionar ou a carta mais à esquerda ou a carta mais à direita da mesa e pegá-la para si.

Em seguida você deve escolher L das cartas que você pegou e virá-las.

A sua pontuação será igual à soma do número na face da frente de todas as K cartas escolhidas, mais a soma do número na face de trás de todas as L cartas viradas.

O desafio? Conseguir a maior pontuação possível, é claro!

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 10^5$). A segunda linha contém N inteiros A_1, A_2, \dots, A_N , ($1 \leq A_i \leq 10^9$). A terceira linha contém N inteiros B_1, B_2, \dots, B_N , ($1 \leq B_i \leq 10^9$). A quarta linha contém dois inteiros K e L ($1 \leq L \leq K \leq N$).

Saída

Imprima uma linha contendo um inteiro, representando a maior pontuação possível.

Exemplo de entrada 1 5 9 7 2 2 9 5 2 2 3 1 2 1	Exemplo de saída 1 23
Exemplo de entrada 2 5 9 7 2 2 9 5 9 2 3 1 2 1	Exemplo de saída 2 25