V Escola de Inverno da Maratona de Programação

Caderno de Problemas

23 de julho de 2024

Realização

Organização

(Este caderno contém 12 problemas)

Comissão Organizadora:
Alberto Tavares Duarte Neto (UnB)
Bruno César Ribas (UnB/FGA)
Caleb Martim de Oliveira (UnB)
Daniel Saad Nogueira Nunes (IFB)
Edson Alves da Costa Júnior (UnB/FGA)
Eduardo Freire dos Santos (UnB)
Eduardo Schwarz Moreira (UDESC)
Eric Grochowicz (UDESC)
Guilherme Novaes Ramos (UnB)
Gustavo Machado Leal (UFG)
Henrique de Oliveira Ramos (UnB)
Maria Eduarda Carvalho Santos (UnB)
Vinicius Ruela Pereira Borges (UnB)

Patrocínio

Lembretes

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova, entretanto, o mesmo não vale para materiais dispostos eletronicamente.
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída conforme as amostras dos exemplos. Deve-se considerar entradas e saídas padrão;
- Para cada problema, além dos testes públicos, o juiz executará a sua submissão contra uma série de testes secretos para fornecer um parecer sobre a correção do programa.
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. Lembre-se que as soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Utilize a aba *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;

C/C++

- Seu programa deve retornar zero, executando, como último comando, `return 0` ou `exit 0`.

Java

- Não declare `‘package’` no seu programa Java.
- Note que a conveção para o nome do arquivo fonte deve ser obedecida, o que significa que o nome de sua classe pública deve ser uma letra maiúscula igual a letra que identifica o problema.

Python

- Tenha cuidado ao selecionar a versão correta na submissão.

Problema A – Alazão

Limite de tempo: 1s

Limite de memória: 256MB

Eric tem um estábulo cheio de cavalos, dos quais o mais veloz é o Alazão.

Como Eric sempre se atrasa quando vai até a faculdade caminhando, ele decide finalmente aprender a andar no Alazão. Para isso, ele pediu ajuda a Eduardo e Enzo.

Eduardo construiu uma pista de treino usando N marcações no chão, numeradas de 1 a N. Entre duas marcações pode haver desenhadas estradas de mão única. É possível que haja simultaneamente uma estrada que vai de de a até b e uma de b até a. Dessa forma, cada estrada possui uma origem e um destino, as marcações que ela conecta. O desafio consiste em partir da marcação de número 1 e, através das estradas desenhadas, chegar naquela de número N.

Enzo informou a Eric que o trote de Alazão o faz se mover pela pista dando saltos de tamanho exatamente K. Ou seja, a cada passo, ele pode escolher K estradas consecutivas (o destino da primeira é a origem da segunda, e assim por diante) partindo da marcação atual e ir até o destino da última destas. Enzo deixou claro que é possível repetir marcações e estradas em um mesmo salto.

Agora, Eric está em dúvida se, montado no Alazão, é possível concluir o desafio proposto por Eduardo. Por isso, ele pediu a sua ajuda. Dados uma pista e o tamanho do salto de Alazão, determine se é possível ou não concluir o desafio.

Entrada

A primeira linha da entrada contém dois inteiros N ($1 \leq N \leq 500$) e M ($0 \leq M \leq N \cdot (N-1)$), o número de marcações e estradas no circuito.

As próximas M linhas contêm, cada uma, dois inteiros a_i e b_i ($1 \leq a_i, b_i \leq N$), a origem e o destino da i-ésima estrada.

A última linha contém um inteiro K ($1 \leq K \leq 500$), o tamanho do salto de Alazão.

Saída

Imprima "`Sim`" (sem aspas duplas) se é possível que Eric conclua o desafio com aquele valor de K e "`Nao`" (sem aspas duplas) se não for possível.

Exemplo

Entrada	Saída

`5 5`
`1 3`
`3 2`
`3 5`
`2 4`
`4 3`
`1`
	`Sim`

`3 3`
`1 2`
`2 1`
`2 3`
`11`
	`Sim`

`5 3`
`1 2`
`2 3`
`4 5`
`1`
	`Nao`

Problema B – Balanceando Competições

Limite de tempo: 1s

Limite de memória: 1024MB

Preparar competições não é uma tarefa fácil, em geral a dificuldade dos problemas pode ser decomposta em dois fatores muito relevantes, a dificuldade teórica de um problema e a dificuldade pratica.

A dificuldade teórica é sobre a dificuldade antes de colocar a mão na massa, o quão difícil é pensar em um devido problema. A dificuldade pratica é sobre a dificuldade ao implementar uma solução já pensada. O desbalanço de um problema é definido como a diferença absoluta entre essas duas dificuldades.

Beralto está preparando uma competição e decidiu q esta terá N problemas. Além disso ele quer que os problemas tenham níveis diversos, logo, ele decidiu que a dificuldade teórica e a dificuldade pratica dos problemas serão uma permutação dos valores entre 1 e N. Assim, se $N = 2$ existem 4 possíveis distribuições:

- $(1, 1), (2, 2)$;
- $(1, 2), (2, 1)$;
- $(2, 1), (1, 2)$;
- $(2, 2), (1, 1)$.

A única preocupação de Beralto é o desbalanço final da competição, isto é, a soma do desbalanço de cada um dos problemas. Assim Beralto decidiu analisar a viabilidade desta ideia, ele definiu um limite T e decidiu verificar quantas distribuições possíveis de dificuldades teóricas e praticas existem, tal que o desbalanço final seja menor ou igual a T. Como Beralto está muito ocupado pensando nos problemas em si, ajude-o a fazer este cálculo.

Entrada

A entrada conterá uma linha com dois inteiros N e T ($1 \leq N \leq 80$, $0 \leq T \leq 6400$), onde N é a quantidade de problemas e T é o limite de desbalanço final.

Saída

Imprima o resto da divisão por $10^9 + 7$ do resultado para a análise de Beralto.

Exemplo

Entrada	Saída
2 2	4
2 1	2
5 0	120
4 10	576
42 1234	444068236

Problema C – Cidades Planejadas

Limite de tempo: 1s

Limite de memória: 256MB

O governo da Nlogônia irá criar diversas cidades em uma região de planalto visando construir cidades planejada como ocorre em Brasília. O mapa de cada cidade será quadrado $N \times N$, sendo cada posição (i, j) uma construção (prédios, casas, etc), ou elementos de paisagismo (rios, jardins, lagoas, etc), ou um terreno vago.

O governante da Nlogônia definiu que o mapa da cidade deverá ser simétrico em relação a um riacho, que será construído na diagonal principal do mapa, isto é, ligando a posição $(1, 1)$ (canto superior esquerdo) à posição (N, N) (canto inferior direito). Os arquitetos e engenheiros ficaram confusos no momento do pedido e elaboraram diversos mapas. Eles pedem sua ajuda! Determine se o mapa fornecido é simétrico em relação a esse riacho artificial.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 100$) indicando as dimensões do mapa quadrado da cidade.

Em seguida, existem N linhas descrevendo o mapa da cidade. Cada uma das N linhas apresenta N caracteres. Assim, a i-ésima e a j-ésima coluna apresentam o caractere '`.`', caso seja um terreno vago, ou o caractere '`*`' para o caso de jardins, ou '`#`' para construção, ou o caractere '`R`', caso seja uma posição do riacho artificial.

É garantido que o riacho existe em todos os casos de teste.

Saída

Imprima "Sim" (sem aspas duplas) caso o mapa da cidade seja simétrico em relação à diagonal principal. Caso contrário, imprima "Nao" (sem aspas duplas).

Exemplo

Entrada	Saída

`2`
`R.`
`.R`
	`Sim`

`3`
`R.*`
`.R.`
`*.R`
	`Sim`

`4`
`R.**`
`.R*.`
`.*R.`
`**.R`
	`Nao`

Problema D – De volta para casa

Limite de tempo: 1s

Limite de memória: 512MB

Joãozinho quer voltar para casa de metrô, mas ele não sabe praticamente nada sobre como funciona a complicadíssima rede de metrôs de Brasília. O que ele sabe é que existem exatamente n estações de metrô enumeradas de 1 até n e existem $n-1$ conexões (todas de mão dupla) entre as estações. Além disso, é possível alcançar qualquer estação a partir de qualquer outra utilizando a rede de metrôs. Ele também sabe que sua casa fica logo ao lado da estação número 1.

Como Joãozinho não sabe o que deve fazer para voltar para casa, quando ele está em uma estação de metrô ele escolhe a próxima conexão que irá pegar de forma uniformemente aleatória. Isso significa que se Joãozinho está em uma estação que faz conexões com as estações $a_1, a_2, \ldots, a_m$, então a probabilidade dele ir para uma dada estação a_i (com $1 \leq i \leq m$) é de $\frac{1}{m}$.

Para cada uma das n estações possíveis que Joãozinho pode começar, determine o valor esperado de conexões que Joãozinho pegará até alcançar a estação 1 pela primeira vez. É possível provar que este valor pode ser escrito da forma p/q em que p e q são inteiros não negativos e q é coprimo com 998244353. Imprima $p \cdot q^{-1} \mod 998244353$, em que q^{-1} é o inverso multiplicativo de q módulo 998244353.

Entrada

A primeira linha de entrada contém um único inteiro n ($2 \leq n \leq 2 \cdot 10^5$) — o número de estações de metrô.

Cada uma das próximas $n-1$ linhas contêm dois inteiros a, b ($1 \leq a, b \leq N$, $a \neq b$) indicando uma conexão de mão dupla entre as estações a e b. É garantido que é possível alcançar qualquer estação de metrô a partir de qualquer outra.

Saída

Para cada estação a de 1 até n, imprima exatamente um inteiro: $p \cdot q^{-1} \mod 998244353$, o valor esperado de conexões que Joãozinho deve pegar para chegar na estação 1 começando a partir da estação a.

Exemplo

Entrada	Saída
2 1 2	0 1
4 1 2 2 3 3 4	0 5 8 9
10 8 3 8 4 1 5 1 6 1 8 2 1 7 2 7 9 7 10	0 7 6 6 1 1 12 5 13 13

Notas

No primeiro caso de teste, o valor esperado de conexões para Joãozinho ir da estação 2 para a estação 1 é exatamente 1. De fato, a única conexão que a estação 2 faz é com a estação 1, então ele sempre a alcançará após pegar uma conexão.

Em todos os casos de teste o valor esperado de conexões atravessadas para atingir a estação 1 a partir da estação 1 é 0.

Problema E – Estrutural

Limite de tempo: 2s

Limite de memória: 256MB

A Cidade Estrutural é uma região administrativa do Distrito Federal. Apesar de ter sofrido, por muito tempo, problemas ambientais e de infra-estrutura, apresentou muito progresso nos últimos tempos. Atualmente, essa região conta com diversas iniciativas de reciclagem para solucionar os problemas de outrora.

Essas iniciativas trabalham no regime de coleta seletiva e separam os materiais de acordo com sua natureza para que eles possam ser reaproveitados posteriormente.

Por sua vez, as equipes de coleta seletiva, elaboraram a seguinte heurística para transformar lixo em luxo. Como Brasília é bem divida em setores, deve-se visitar todos eles, efetuando a coleta, e voltar ao setor original. Contudo, para otimizar o tempo, o percurso de um setor a outro só pode ser feito se a sigla do setor for igual a sigla de outro setor realizando a substituição de, no máximo, X letras.

Ajude a equipe a determinar se é possível realizar esse percurso e qual é ele.

Entrada

A primeira linha da entrada possui dois inteiros, separados por um espaço, N e X.

As próximas N linhas possuem, cada, uma palavra S_i indicando a sigla do setor.

Restrições:

- $1 \leq N \leq 20$
- $1 \leq X \leq 10$
- A string S_i é composta por símbolos sobre o alfabeto $\Sigma = \{A, \ldots, Z\}$, $1 \leq i \leq N$
- $|S_i| = |S_j|$, $1 \leq i, j \leq N$
- $S_i \neq S_j$, $1 \leq i, j \leq N$ e $i \neq j$
- $1 \leq |S_i| \leq 10$, $1 \leq i \leq N$

Saída

Dê como saída o trajeto realizado pela equipe de coleta seletiva. Cada setor deve estar separado do outro por “ `->` ”.

Exemplo

Entrada	Saída

`4 1`
`SCS`
`SCN`
`SHN`
`SHS`
	`SCS -> SHS -> SHN -> SCN -> SCS`

`3 1`
`SCN`
`SCS`
`SHN`
	`impossivel`

`3 2`
`SCN`
`SCS`
`SHN`
	`SCN -> SHN -> SCS -> SCN`

Notas

No primeiro exemplo, é possível realizar o percurso na seguinte ordem: Setor Comercial Sul, Setor Hospitalar Sul, Setor Hospitalar Norte, Setor Comercial Norte e Setor Comercial Sul. Outra possibilidade é `SHS -> SHN -> SCN -> SCS -> SHS`.

No segundo exemplo, é impossível cobrir todos os setores.

No terceiro exemplo, como $X = 2$, é possível ir de `SHN` para `SCS` sem problemas.

Problema F – Fila da Padaria

Limite de tempo: 1s

Limite de memória: 256MB

Todo dia pela manhã, *Wawa* vai à Pão Dourado comprar pão para seu café da manhã e hoje você decidiu acompanhá-lo. Uma unidade de pão custa P reais, onde P é um número inteiro, para facilitar as contas da padaria.

Hoje a fila da Pão Dourado possui N pessoas, onde para cada $i = 1, 2, \ldots, N$, a pessoa na i-ésima posição da fila está com A_i reais no bolso.

Você percebeu que cada pessoa na fila está preocupada com as pessoas à sua frente que conseguem comprar uma quantidade de pão maior do que ela mesma pode. Afinal, a quantidade de pão que se come no café da manhã é um símbolo de status.

Você e *Wawa* não tinham nada melhor para fazer enquanto esperavam, então decidiram descobrir: para cada pessoa na fila, quantas pessoas à frente dela conseguem comprar mais pão do que ela.

Entrada

A primeira linha contém dois inteiros N e P $(1 \leq N \leq 2 \cdot 10^5,\ 1 \leq P \leq 10^9)$ — quantas pessoas estão na fila da padaria hoje e quanto uma unidade de pão custa em reais.

A segunda linha contém N inteiros $A_1, A_2, \ldots, A_N$ $(1 \leq A_i \leq 10^9)$ — quantos reais a pessoa na i-ésima posição da fila possui.

Saída

Imprima N inteiros separados por espaço, onde o i-ésimo inteiro representa quantas pessoas à frente da pessoa na i-ésima posição da fila podem comprar mais pão do que ela.

Exemplo

Entrada	Saída
5 1 5 3 4 2 1	0 1 1 3 4
4 2 5 3 4 6	0 1 0 0
6 8 36 42 31 19 36 15	0 0 2 3 1 5

Problema G – Gama, sempre Gama!

Limite de tempo: 1s

Limite de memória: 256MB

Fundada em 1975, a Sociedade Esportiva do Gama é um dos mais tradicionais times de Brasília, tendo como ponto alto de sua história a participação na Série A do Campeonato Brasileiro entre 1999 e 2002.

O time está se preparando para lançar, em 2025, um vídeo comemorativo de seus 50 anos. Foram pré-selecionados, dentre os vídeos do acervo do clube, N clipes que resgatam a momentos marcantes da trajetória do time. Cada clipe tem duração de t segundos e um fator de engajamento e, estimado a partir de dados como o público no dia do jogo, curtidas em redes sociais, etc.

O orçamento de publicidade tem recursos para um vídeo de, no máximo, M minutos. Com o intuito de obter uma boa recepção da torcida, o clube te contratou para selecionar zero ou mais clipes, dentre os N pré-selecionados, de modo que soma das durações dos clipes escolhidos seja menor ou igual a M e que a soma dos fatores de engamento dos clipes escolhidos seja a maior possível. Para que o vídeo não fique repetitivo, cada clipe só pode ser escolhido uma única vez.

Entrada

A primeira linha da entrada contém os valores dos inteiros N ($1 \leq N \leq 100$) e M ($1 \leq M \leq 5 \times 10^3$), separados por um espaço em branco, onde N é o número de clipes pré-selecionados e M a duração máxima do vídeo, em minutos.

Cada uma das N linhas seguintes contém um par de inteiros d_i ($1 \leq d_i \leq 3 \times 10^3$) e e_i ($1 \leq e_i \leq 10^9$), separados por um espaço em branco, onde t_i e e_i representam a duração, em segundos, e o fator de engajamento do i-ésimo ($1 \leq i \leq N$) clipe, respectivamente.

Saída

Imprima, em uma linha, o engamento máximo que pode ser obtido por meio da seleção de zero ou mais dentre os N clipes pré-selecionados, com as restrições de que cada clipe só pode ser escolhido uma única vez e que a soma das durações dos clipes escolhidos não pode exceder M minutos.

Exemplo

Entrada	Saída
3 10 3 7 5 8 2 10	25
2 10 2000 1 1000 2	0
5 5 300 10 1 1 1 2 2 3 1 4	10

Notas

No primeiro caso, os três clipes podem ser selecionados, obtendo um fator de engajamento igual a $7 + 8 + 10 = 25$.

No segundo caso, nenhum clipe pode ser selecionado.

Problema H – Hortaliças

Limite de tempo: 1s

Limite de memória: 256MB

Embora várias de suas regiões administrativas tenham alta densidade populacional, o Distrito Federal possui grandes áreas rurais, onde são produzidos diversos produtos agrícolas, dentre eles hortaliças.

Em uma das fazendas de produção de hortaliças, a área para plantio é delimitada pelo paralelogramo $ABCD$, cujos lados AD e BC são paralelos ao eixo-y. As arestas do paralelogramo marcam a posição dos canos por onde passa a água que alimenta os N irrigadores posicionados no interior, ou mesmo sob a borda, do paralelogramo (uma vez que a encanação é subterrânea).

Com o intuito de reformar a encanação, o fazendeiro precisa desativá-la temporariamente. Para isso, ele precisa passar uma encanação temporária para alimentar os irrigadores. Ele deseja que esta encanação seja posicionada em um segmento de reta r com as seguintes restrições:

1. os extremos S e T de r deve estar posicionados nos lados AD e BC, um extremo em cada lado, nesta ordem;
2. S e T devem ter coordenadas inteiras;
3. $S \neq A, S \neq D, T \neq B, T \neq C$;
4. r deve ser seja paralelo aos lados AB e CD; e
5. deve ser minimizada a soma das distâncias dos irrigadores para r. Formalmente, se o i-ésimo irrigador está posicionado no ponto $P_i = (x_i, y_i)$, então deve ser minimizado o valor $d(r, N)$, onde

$$d(r, N) = \sum_{i=1}^{N} \text{dist}(r, P_i)$$

Auxilie o fazendeiro determinando os pontos S e T que satisfaçam as restrições impostas.

Entrada

As quatro primeiras linhas da entrada descrevem o paralelogramo, por meio das coordenadas inteiras x e y ($0 \leq x, y \leq 10^6$) dos pontos A, B, C e D, respectivamente, separadas por um espaço em branco. É garantido que $A_x = D_x, B_x = C_x, D_y - A_y > 1$ e que AB e CD são paralelos.

A quinta linha contém o valor do interior N ($1 \leq N \leq 2 \times 10^5$), que indica o número de irrigadores.

As N linhas seguintes contém as coordenadas inteiras x e y ($0 \leq x, y \leq 10^6$) de cada irrigador. É garantido que cada irrigador esteja posicionado ou no interior, ou em um dos lados, do paralelogramo.

Saída

Na primeira linha imprima as coordenadas x e y do ponto S, separadas por um espaço em branco.

Na segunda linha imprima as coordenadas x e y do ponto T, separadas por um espaço em branco.

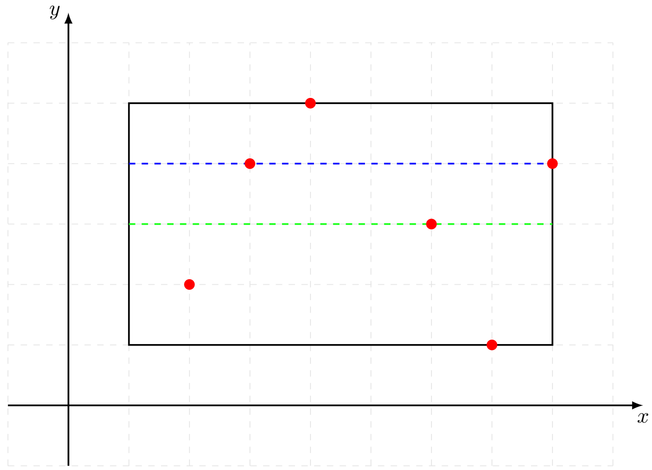
Se existir mais de um par de pontos que satisfaçam as restrições impostas, imprima qualquer um deles.

Exemplo

Entrada	Saída
1 1 8 1 8 5 1 5 6 2 2 3 4 4 5 6 3 7 1 8 4	1 4 8 4
2 1 12 3 12 12 2 10 7 4 3 5 2 7 2 7 3 9 6 10 9 11 5	2 3 12 5

Notas

No primeiro caso, a soma das distâncias dos irrigadores para a reta proposta (em azul na figura) é igual a $2 + 0 + 1 + 1 + 3 + 0 = 7$. Observe que $S = (1, 3)$ e $T = (8, 3)$ (em verde na figura) seria uma resposta igualmente válida. Veja a figura abaixo, onde os irrigadores são os pontos vermelhos:



No segundo caso, observe que os segmentos AB e CD não são, necessariamente, paralelos ao eixo-x.

Problema I – Indo ao Mercado

Limite de tempo: 1s

Limite de memória: 256MB

Guilherme está se preparando para a Maratona de Inverno, e um dos seus afazeres é comprar o coffee-break. Para isso, ele está avaliando os custos com um aplicativo de última geração, que pode prever o custo e o estoque de um certo produto em um certo dia.

Guilherme sabe que o i-ésimo dia do evento irá precisar de uma quantidade de d_i refrigerantes e que neste mesmo dia haverá um estoque de promoção, com quantidade de a_i refrigerantes ao custo de c_i por refrigerante. O estoque da promoção não precisa ser comprado de forma integral, mas a quantidade de refrigerantes demandada em um certo dia tem que ser completamente cumprida.

Há também um custo para guardar os refrigerantes, de modo que, se Guilherme guardar um refrigerante do dia i para o dia $i+1$, o custo para mantê-lo na geladeira é l_i. Usando esta estratégia, ele pode, por exemplo, comprar um refrigerante no dia 1, mantê-lo na geladeira por 3 dias e abri-lo somente no dia 4.

Como dinheiro não cai do céu, calcule o menor custo para cumprir as demandas de refrigerante de todos os N dias de evento. Guilherme já verificou previamente e tem certeza que é possível cumprir as demandas.

Entrada

A primeira linha contém um inteiro N representando a duração do evento em dias ($1 \leq N \leq 2 \cdot 10^5$).

A segunda linha contém N inteiros, representando a quantidade d_i de refrigerantes necessários no i-ésimo dia do evento ($1 \leq d_i \leq 10^5$).

A terceira linha contém N inteiros, representando a quantidade a_i de refrigerantes no estoque de promoção no i-ésimo dia ($1 \leq a_i \leq 10^5$).

A quarta linha contém N inteiros, representando o custo c_i de cada refrigerante no estoque do i-ésimo dia ($1 \leq c_i \leq 10^5$).

A quinta e última linha contém $N-1$ inteiros, representando o custo l_i para manter os refrigerantes guardados do dia i para o dia $i+1$ ($1 \leq l_i \leq 10^5$).

Saída

Imprima uma única linha contendo um inteiro representando o custo mínimo para cumprir as necessidades do Guilherme.

Exemplo

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

Notas

No exemplo dado, Guilherme pode ter chegado no custo 5 da seguinte forma:

- No primeiro dia, Guilherme comprou 2 refrigerantes de custo 1 cada. Um refrigerante é usado no evento e o outro é guardado na geladeira com custo 1.
- No segundo dia, Guilherme usa o refrigerante que estava guardado na geladeira para o evento.
- No terceiro dia, Guilherme compra 1 refrigerante de custo 2 e o usa no evento.

O gasto total foi então $(2 \cdot 1) + 1 + (1 \cdot 2) = 5$.

Problema J – Jogo da Moeda

Limite de tempo: 1s

Limite de memória: 256MB

Joãozão inventou um novo jogo para jogar com os seus amigos, o Jogo da Moeda.
O jogo da moeda com N jogadores funciona da seguinte maneira:

- Um dos jogadores é sorteado para receber uma moeda.
- Todos os N jogadores se organizam em um círculo (todos olhando para o centro do círculo), de forma que o i-ésimo jogador está à esquerda do $(i-1)$-ésimo jogador, e o primeiro jogador está à esquerda do N-ésimo jogador.
- Um jogador é escolhido para ser o caçador da moeda.
- A posição do círculo em que o i-ésimo jogador está possui um valor a_i, sempre que o caçador está nessa posição, ele pode andar algum múltiplo de a_i no sentido horário ou anti-horário (podendo fazer quantas voltas no círculo ele quiser). Ou seja, ele pode andar de a_i em a_i até chegar na posição em que deseja. Isso é chamado de um "movimento".
- Quando o caçador chega na posição do jogador que possui a moeda, o jogo acaba.

Joãozão é o X-ésimo jogador na ordem do círculo, e ele vai ser o caçador dessa próxima rodada. Ele quer bolar uma estratégia para achar a moeda o mais rápido possível.

Para ajudar Joãozão, você deve dizer: Para todo Y $(1 \leq Y \leq N)$, é possível achar a moeda se ela estiver com o jogador Y? Se sim, qual é o menor número de movimentos para Joãozão achar a moeda se ela estiver com o jogador Y?

Entrada

A primeira linha da entrada consiste de dois inteiros N e X $(1 \leq X \leq N \leq 10^5)$, a quantidade de jogadores, e quem é Joãozão no círculo, respectivamente.

A segunda linha da entrada consiste de N inteiros $a_1, a_2, \cdots, a_N$ $(1 \leq a_i \leq 10^9)$, onde a_i é o valor da posição em que o i-ésimo jogador está.

Saída

A saída deve conter N inteiros, o i-ésimo número deve conter o menor número de movimentos para achar a moeda se ela estiver com o i-ésimo jogador, ou -1 se for impossível achar a moeda.

Exemplo

Entrada	Saída
6 3 1 2 2 4 3 8	1 2 0 2 1 2
1 1 3	0
8 2 1 4 2 2 4 2 4 6	-1 0 -1 2 -1 1 -1 2

Notas

No primeiro caso de teste, para sair da posição 3 e chegar na 1 com passos múltiplos de 2, é possível andar 2 para a esquerda ou 4 para a direita. Não é possível sair da posição 3 para a 2 com apenas 1 passo, mas é possível ir para a posição 1 e logo em seguida para a 2 (um passo de tamanho 2 para a esquerda seguida de um passo de tamanho 1 para a direita, por exemplo).

Problema K – Kart Indoor

Limite de tempo: 1s

Limite de memória: 256MB

Uma turma de amigos decidiu passar o último sábado no Kart Indoor Brasília para andarem de kart e se divertirem. Os amigos fizeram diversas baterias, sendo que algumas foram mais emocionantes, enquanto que outras foram mais mornas. Vendo a situação, o dono do Kart Indoor disse que daria uma bateria gratuita quando ocorressem muitas ultrapassagens em algumas delas.

Cada kart é representado por um único número inteiro de 1 a N, que está associado à posição de largada. Isso significa que o kart 1 larga na posição 1, o kart 2 larga na segunda posição, e assim por diante. Como os amigos estão se divertindo no kartódromo, eles pedem sua ajuda. Dada a ordem de chegada dos karts, determina a quantidade mínima de ultrapassagens que ocorreram em uma bateria.

Entrada

A primeira linha contém um número inteiro N ($1 \leq N \leq 5 \cdot 10^3$), indicando a quantidade total de karts que participaram da corrida de kart indoor.

A segunda linha contém N números inteiros separados por espaço em branco $a_1, a_2, \ldots, a_N$ ($1 \leq a_i \leq N$), indicando o kart que chegou na i-ésima posição. É garantido que $a_i \neq a_j$ para todo $i \neq j$.

Saída

Imprima um número inteiro contendo a quantidade mínima de ultrapassagens ocorridas na corrida de kart.

Exemplo

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

Problema L – Luka, Miku e Chocolate

Limite de tempo: 1s

Limite de memória: 256MB

É conhecido que, em Brasília, quando se gosta de alguém, você demonstra isso comprando uma barra de chocolate para ela. O problema é que uma pessoa pode não gostar de quem gosta dela e, por isso, ela irá repassar a barra de chocolate que ganhou para quem ela realmente gosta, se este alguém existir.

Megurine Luka e *Hatsune Miku* estão interessadas no fluxo de barras de chocolate, então fizeram uma pesquisa que envolveu N pessoas de Brasília e descobriram de quem cada uma gosta. Por praticidade, cada pessoa foi unicamente identificada com um número inteiro entre 1 a N. Uma pessoa pode ser gostada por zero ou mais pessoas, mas, aqui, cada pessoa gosta de **no máximo** uma outra pessoa.

Para observar os resultados da pesquisa, *Miku* e *Luka* fizeram Q consultas do tipo "se a pessoa identificada pelo número a comprar uma barra de chocolate para quem ela gosta, *em algum momento*, a pessoa identificada pelo número b irá receber esta barra de chocolate?", e ficou para você responder cada uma dessas consultas.

Entrada

A primeira linha contém dois inteiros N e M $(2 \leq N \leq 2 \cdot 10^5,\ 1 \leq M \leq N)$ — quantas pessoas foram envolvidas na pesquisa e quantas entre elas gostam de alguém.

Cada uma das próximas M linhas contém dois inteiros a e b $(1 \leq a, b \leq N,\ a \neq b)$ — indicando que a pessoa identificada pelo número a gosta da pessoa identificada pelo número b. É garantido que a pessoa identificada pelo número a gosta de **no máximo** uma pessoa.

A linha seguinte contém um inteiro Q $(1 \leq Q \leq 2 \cdot 10^5)$ — quantas consultas foram feitas por *Hatsune Miku* e *Megurine Luka*.

Cada uma das próximas Q linhas contém dois inteiros a e b $(1 \leq a, b \leq N,\ a \neq b)$ — indicando a consulta "se a pessoa identificada pelo número a comprar uma barra de chocolate para quem ela gosta, *em algum momento*, a pessoa identificada pelo número b irá receber esta barra de chocolate?"

Saída

Para cada uma das Q consultas imprima uma linha contendo a string "Sim" (sem aspas), no caso afirmativo da consulta ou "Nao" (sem aspas), caso contrário.

Exemplo

Entrada	Saída
3 2 1 2 2 3 2 1 3 3 1	Sim Nao
4 4 1 2 2 1 3 4 4 3 2 3 1 3 4	Nao Sim
5 5 1 2 2 3 3 1 4 2 5 2 3 2 1 4 5 5 4	Sim Nao Nao

Notas

No primeiro exemplo: se a pessoa 1 comprar um chocolate, ela irá dar ele para a pessoa 2, a pessoa 2 irá dar esse chocolate para a pessoa 3. Logo, a resposta para a primeira consulta é "Sim". A pessoa 3 não gosta de ninguém, então ela não irá comprar nenhum chocolate. Por causa disso, a resposta da segunda consulta é "Nao".

No segundo exemplo: se a pessoa 3 comprar um chocolate, ela irá dar esse chocolate para a pessoa 4, como a pessoa 4 gosta da pessoa 3, ela irá dar esse chocolate para ela de volta. Essas ações irão se repetir infinitamente. A pessoa 1 nunca irá receber o chocolate. Então, a resposta para a primeira consulta é "Nao" e para a segunda é "Sim".