Maratona de Programação

2026.1

Caderno de Problemas: Standard

Organização:

Maratona
</> PPCI

Clube de Programação
(UTFPR Curitiba)

Problema A. Anceloti e Endrick

Tempo limite: 1000 ms
Memória limite: 256 MiB

Desde que passou a integrar a seleção brasileira, Endrick percebeu que suas preferências raramente duram muito tempo. Sempre que ele demonstra gostar de algum tipo de treino, Carlo Ancelotti resolve mudar a programação da equipe.

Existem n tipos de atividades de treinamento, numeradas de 1 a n. Inicialmente, Endrick gosta de todas elas. Em seguida, Ancelotti anuncia q decisões, uma após a outra. Cada decisão é representada por três números l, r e k:

- Se $k = 1$, então Ancelotti faz com que Endrick deixe de gostar de todas as atividades com índices entre l e r (inclusive). Mesmo que alguma dessas atividades tenha voltado a ser do agrado de Endrick após uma decisão anterior, ela passa novamente a não ser do seu gosto.
- Se $k = 2$, então Ancelotti faz com que Endrick volte a gostar de todas as atividades com índices entre l e r (inclusive). Mesmo que alguma dessas atividades tenha deixado de ser do seu gosto por uma decisão anterior, ela passa novamente a ser apreciada.

Após cada decisão de Ancelotti, determine quantas atividades Endrick ainda gosta.

Entrada

A primeira linha contém um inteiro n e um inteiro q ($1 \le n \le 10^9$, $1 \le q \le 3 \cdot 10^5$) o número de atividades e o número de decisões de Ancelotti, respectivamente.

As próximas q linhas contêm três inteiros l_i, r_i e k_i, descrevendo a i-ésima decisão ($1 \le l_i \le r_i \le n$, $1 \le k_i \le 2$).

Saída

Imprima q inteiros. O i-ésimo deles deve ser igual ao número de atividades de que Endrick gosta após a aplicação das primeiras i decisões de Ancelotti.

Exemplos

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

Problema B. Bola Passada

Tempo limite: 500 ms
Memória limite: 256 MiB

Durante um treino da seleção para a Copa do Mundo, N jogadores formam um círculo. Os jogadores são numerados de 1 a N em sentido horário.

Inicialmente, o jogador K está com a bola. Enquanto houver algum jogador que ainda não recebeu a bola nenhuma vez, o jogador que está com ela realiza um passe:

- para o jogador imediatamente à sua esquerda, com probabilidade 50%; ou
- para o jogador imediatamente à sua direita, com probabilidade 50%.

O treino termina no momento em que todos os jogadores tiverem recebido a bola pelo menos uma vez.

O técnico está interessado em um intervalo específico de jogadores. Considere o conjunto de jogadores obtido ao partir do jogador L e seguir no sentido horário até alcançar o jogador R.

Considerando apenas treinos que acabam em algum momento, determine a probabilidade de que o último jogador a receber a bola (imediatamente antes de o treino acabar) pertença à esse intervalo.

Entrada

A entrada consiste em uma linha contendo 4 inteiros N, K, L, R ($1 \leq N \leq 10^8, 1 \leq K, L, R \leq N$) onde:

- N é o número de jogadores;
- K é o jogador que inicia com a bola;
- L é uma extremidade do intervalo fechado.
- R é a outra extremidade do intervalo fechado. O intervalor começa em L e segue no sentido horário até alcançar R.

Saída

Imprima a probabilidade de que o último jogador a receber a bola pertença ao intervalo dado S.

A resposta será considerada correta se o erro absoluto for menor que 10^{-6}.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
`5 1 2 4`	`0.750000000000`

Exemplo de entrada 2	Exemplo de saída 2
`8 3 1 3`	`0.285714285714`

Problema C. Campeões da Copa

Tempo limite: 1000 ms
Memória limite: 256 MiB

Faltam poucos minutos para o jogo do Brasil na Copa e a tensão está no ar (junto com o cheiro de churrasco). Uma passarela estreitíssima — daquelas em que só cabe um torcedor de cada vez — liga a rua às catracas do estádio. Ela é um segmento de reta de comprimento M: o Portão Esquerdo fica na posição 0 e o Portão Direito na posição M. Quem alcança um portão entra no estádio, some no mar de camisas amarelas e não atrapalha mais ninguém.

Sobre a passarela há n torcedores, todos em posições inteiras distintas e estritamente entre os portões. No instante 0, ao apito inicial, cada um sai marchando a 1 metro por segundo numa direção fixa: para a esquerda (`E`, sentido decrescente) ou para a direita (`D`, sentido crescente).

Acontece que a passarela é apertada e a torcida é fervorosa. Quando dois torcedores se encontram exatamente na mesma posição (sempre vindo um de cara para o outro), em vez de educadamente darem passagem, eles *começam a discutir a escalação*: "Coloca o ENDRICK!", "RAFINHA é bagre". Como ninguém dá o braço a torcer, no mesmo instante ambos giram 180° e saem pisando duro na direção oposta à que vinham — ainda a 1 m/s, e bufando. Tudo é instantâneo e, por sorte, cada encontro envolve exatamente dois torcedores.

Um torcedor que alcança um portão (posição 0 ou M) entra no estádio naquele instante e, livre da confusão, não volta mais para a passarela.

Passados T segundos, a transmissão quer saber, para cada torcedor (na ordem em que foram dados na entrada): a sua posição na passarela, se ainda estiver lá brigando; ou por qual portão ele finalmente conseguiu entrar.

Entrada

A primeira linha contém três inteiros n, M e T ($1 \le n \le 2 \times 10^5$, $2 \le M \le 10^9$, $0 \le T \le 10^9$), o número de torcedores, o comprimento da passarela e o tempo decorrido.

Cada uma das n linhas seguintes contém um inteiro x_i e um caractere c_i ($0 < x_i < M$, $c_i \in \{\texttt{E}, \texttt{D}\}$), a posição e a direção inicial do i-ésimo torcedor. Todas as posições x_i são distintas.

Saída

Imprima n linhas, uma para cada torcedor, na ordem em que foram dados na entrada. Para o i-ésimo torcedor, imprima o inteiro com a sua posição na passarela no instante T, se ele ainda estiver nela; caso contrário, imprima `ESQUERDA` se ele entrou pelo Portão Esquerdo, ou `DIREITA` se entrou pelo Portão Direito.

Exemplos

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

Explicação do exemplo 1

Inicialmente, o torcerdor 1 está na posição 2 indo para a direita, e o torcedor 2 está na posição 5 indo para a esquerda. Após 1,5 segundo eles se encontram, brigam e mudam suas direções. Assim, após 2 segundos, o torcedor 1 está na posição 3 indo para a esquerda, e o torcedor 2 está na posição 4 indo para a direita.

Exemplo de entrada 2	Exemplo de saída 2
4 10 6 3 E 4 D 5 E 8 D	ESQUERDA ESQUERDA DIREITA DIREITA

Problema D. Duplicatas

Tempo limite: 1000 ms
Memória limite: 256 MiB

É clima de Copa do Mundo! E os torcedores estão em busca de completar seus álbuns de figurinhas. Ricardo e Daniel são dois amigos que estão tentando completar seus álbuns, mas eles tem um problema: ambos estão cheios de figurinhas repetidas.

Então, eles decidiram trocar figurinhas entre si. No entanto, eles querem fazer isso de forma justa: cada um quer dar apenas figurinhas que o outro não possui ainda, e também que não seja sua única, isto é, que ele tenha mais de uma unidade. As trocas são feitas uma de cada vez.

Ricardo tem N figurinhas, enquanto Daniel tem M figurinhas. Cada figurinha possui um ID único, que vai de 1 a 1000.

Ambos os amigos querem trocar o máximo possível de figurinhas. Ajude-os a descobrir quantas trocas serão feitas ao todo.

Entrada

A entrada é composta por uma linha contendo dois inteiros N e M ($1 \leq N, M \leq 10^4$), representando o número de figurinhas que Ricardo e Daniel possuem, respectivamente. Em seguida, há uma linha contendo N inteiros A_i ($1 \leq A_i \leq 1000$), representando os IDs das figurinhas que Ricardo possui. Por fim, há uma linha contendo M inteiros B_i ($1 \leq B_i \leq 1000$), representando os IDs das figurinhas que Daniel possui.

Obs: Ambas as sequencias de inteiros serão entregeues em ordem aleatória, e podem conter inteiros repetidos.

Saída

A saída deve conter um único inteiro, o número total de trocas feitas por Ricardo e Daniel.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2 3 12 34 25 17 25	0

Explicação do exemplo 1

Ricardo possui 2 figurinhas, com IDs 12 e 34. Já Daniel possui 3 figurinhas, com IDs 25, 17 e 25.

Perceba que Ricardo possui apenas uma figurinha de cada ID, então ele não pode trocar nenhuma delas. Já Daniel possui duas figurinhas com ID 25, então ele pode trocar uma delas com Ricardo, que não possui essa figurinha. Porém, como Ricardo não tem nenhuma figurinha que Daniel não tenha, o número máximo de trocas é 0.

Exemplo de entrada 2	Exemplo de saída 2
10 10 56 142 627 38 912 255 711 483 56 38 542 12 203 39 721 411 12 203 614 39	2

Problema E. El Classico

Tempo limite: 1000 ms
Memória limite: 256 MiB

A tão aguardada final da Copa UTFPR finalmente chegou! Após uma temporada histórica, os gigantes Real PPCI e BarCdPlona irão se enfrentar em mais um lendário *El Clássico*. De um lado está o renomado treinador Ricardo Ancelotti, comandante do Real PPCI, conhecido por suas estratégias inovadoras e sua obsessão por análise tática. Do outro lado está seu maior rival, o treinador Zatesko Guardiola, líder do BarCdPlona, famoso por montar formações defensivas praticamente impenetráveis. Na véspera da grande decisão, Ricardo Ancelotti decidiu utilizar uma tecnologia experimental para planejar o posicionamento ideal de seus jogadores em campo. Após estudar centenas de partidas anteriores do adversário, ele identificou diversos pontos estratégicos do gramado onde seus jogadores devem atuar para neutralizar as jogadas do BarCdPlona.

Para organizar sua estratégia, Ricardo Ancelotti deseja dividir seus jogadores em grupos táticos (possivelmente apenas um grupo). Cada grupo será composto por jogadores em uma mesma determinada região circular do campo. Por limitações físicas e de comunicação, as regiões de todos os grupos devem possuir o mesmo raio R. Formalmente, um grupo de jogadores pode ser coberto por uma região circular de raio R se todos os jogadores do grupo estiverem dentro dessa região, jogadores que tangenciam a região circular de um grupo também estão considerados dentro dela.

Seu objetivo é ajudar Ricardo Ancelotti a descobrir o menor número de regiões circulares de raio R necessárias para cobrir todos os jogadores em campo.

Entrada

A primeira linha contém dois inteiros N ($1 \leq N \leq 18$) e R ($2 \leq R \leq 10^6$), onde N é o número de jogadores em campo e R é o raio da(s) região(ões) circular(es).

As próximas N linhas contêm dois inteiros X_i e Y_i ($-10^6 \leq X_i, Y_i \leq 10^6$) representando as coordenadas do i-ésimo jogador no campo.

Saída

Imprima um único inteiro representando o menor número de grupos com regiões circulares de raio R necessários para cobrir todos os jogadores.

Note que é possível que exista soluções onde regiões circulares intersectam umas as outras.

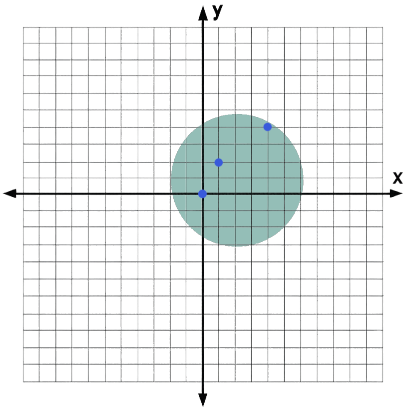
Exemplos

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

`3 4`
`0 0`
`1 2`
`4 4`
	`1`

Explicação do exemplo 1

Os 3 jogadores podem ser incluidos em um único grupo na mesma região circular, conforme a ilustração a seguir:

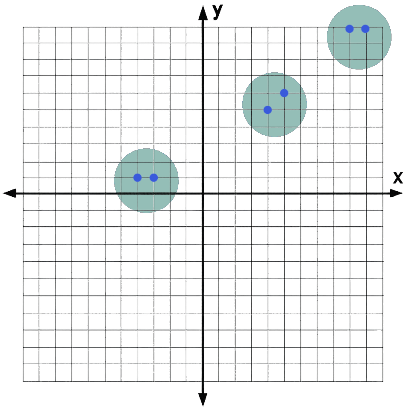


Exemplo de entrada 2	**Exemplo de saída 2**

`6 2`
`-4 1`
`5 6`
`4 5`
`-3 1`
`9 10`
`10 10`
	`3`

Explicação do exemplo 2

É impossível incluir os 6 jogadores em menos que 3 regiões circulares, conforme a ilustração a seguir:



Problema F. Folga Frustrada

Tempo limite: 1000 ms
Memória limite: 256 MiB

Faltam apenas alguns dias para as tão aguardadas quartas de final, e as seleções que avançaram na competição ganharam um dia de folga. Para relaxar e fugir um pouco do estresse do futebol, os jogadores de vários países decidiram organizar um torneio amistoso de Beach Tennis no complexo de hotels onde estão hospedados.

O complexo possui 5 quadras de areia de alto padrão. No entanto, a equipe de esportes do complexo deparou-se com um desastre logístico naquela manhã: devido a uma forte chuva na noite anterior, a rede de proteção e a marcação de 4 das 5 quadras foram completamente destruídas. Apenas uma única quadra está em condições de uso.

Sem saber do incidente, as equipes de vários países já haviam enviado para a recepção os seus pedidos de reserva, indicando o horário exato que queriam começar a jogar e o horário que terminariam.

Para evitar uma crise diplomática e garantir que o dia de folga não se torne um caos, a equipe do resort precisa organizar a agenda da única quadra disponível. A missão é aceitar o maior número possível de reservas, de modo que nenhuma partida se sobreponha à outra (ou seja, um time só pode entrar na quadra depois ou exatamente no mesmo instante em que o time anterior sair).

Entrada

A primeira linha da entrada contém um número inteiro N ($1 \leq N \leq 10^5$), que representa a quantidade de times que solicitaram um horário na quadra.

As N linhas seguintes contêm dois inteiros I e F cada ($0 \leq I < F \leq 24$), representando, respectivamente, o horário de início e o horário de fim solicitados por um time.

Saída

A saída deve conter uma única linha com um número inteiro: o número máximo de times que poderão jogar na quadra sem que haja conflito de horários.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
5 8 10 9 11 10 12 11 13 14 16	3

Explicação do exemplo 1

Existem duas possibilidades que maximizam o número de times que utilizam a quadra: agendar os times 1, 3 e 5, ou agendar os times 2, 4 e 5. Não é possível agendar a quadra de modo que mais de 3 times consigam utilizá-la.

Exemplo de entrada 2	**Exemplo de saída 2**
4 8 20 9 11 12 14 15 17	3

Problema G. Grande Multidão

Tempo limite: 1000 ms
Memória limite: 256 MiB

O dia da grande final da Copa do Mundo de 2026 chegou! A seleção está hospedada em um hotel e precisa chegar ao estádio de ônibus. No entanto, o trânsito da cidade está um caos e várias ruas estão bloqueadas por multidões de torcedores em festa.

A cidade pode ser representada por uma grade (matriz) de N linhas por M colunas. Cada célula da grade representa um quarteirão que pode ter as seguintes características:

- '**.**' (Ponto): Uma rua livre por onde o ônibus pode passar.
- '#' (Cerquilha): Uma rua interditada por torcedores (o ônibus não pode passar).
- '**S**' (Start): O hotel onde a seleção está (início).
- '**E**' (Estádio): O destino final.

O ônibus pode se mover para os quarteirões vizinhos nas direções ortogonais (cima, baixo, esquerda e direita), e cada movimento leva 1 minuto. A comissão técnica precisa da sua ajuda: qual é o tempo mínimo necessário para o ônibus ir do hotel ('S') até o estádio sem passar por ruas bloqueadas?

Entrada

A primeira linha contém dois inteiros, N e M ($1 \leq N, M \leq 1000$), representando as dimensões da cidade.

As próximas N linhas contêm M caracteres cada, descrevendo o mapa da cidade. É garantido que haverá exatamente um caractere 'S' e exatamente um caractere 'E' em todo o mapa. Os demais caracteres serão apenas '.' ou '#'.

Saída

Imprima um único inteiro representando o tempo mínimo (em minutos) para o ônibus chegar ao estádio. Se for impossível chegar ao estádio devido aos bloqueios, imprima **-1**.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1

`4 5`
`S....`
`.###.`
`...#.`
`.#.E.`
	`6`

Explicação do exemplo 1

Neste exemplo o menor caminho para o ônibus chegar ao estádio a partir do ponto de inicio é: (Baixo, Baixo, Direita, Direita, Baixo, Direita), 6 movimentos, logo 6 minutos é o tempo mínimo.

Exemplo de entrada 2	Exemplo de saída 2

`2 2`
`S#`
`#E`
	`-1`

Problema H. Habilidades Futebolísticas

Tempo limite: 1000 ms
Memória limite: 256 MiB

Com a copa chegando, o técnico de um time de futebol decidiu criar um programa para ajudar os jogadores a melhorarem suas habilidades, para que eles obtenham o melhor desempenho possível em jogo. Porém, ele percebeu que certas habilidades exigem que o jogador já tenha treinado uma outra habilidade para ser aprendida.

Assim, ele precisa montar uma lista de habilidades que os jogadores devem treinar, em ordem, de forma que todas as N habilidades são treinadas, e que a ordem respeite as dependências entre as habilidades.

Entrada

A primeira linha da encontra contém o inteiro N ($1 \leq N \leq 10^6$). As próximas N linhas contêm os pré-requisitos das habilidades.

Considere que as próximas N linhas são numeradas de 1 a N, onde a linha i corresponde à habilidade i. A linha i contém um inteiro P ($0 \leq P \leq N$), indicando que a habilidade P é pré-requisito para a habilidade i. Se P for 0, isso indica que a habilidade i não tem pré-requisitos.

Os pré-requisitos não contêm contradições (ou seja, é sempre possível aprender todas as habilidades de pelo menos uma forma). Além disso, é garantido que pelo menos uma habilidade não possui pré-requisitos.

Saída

Imprima em uma linha um único inteiro, o número de formas diferentes que ele pode montar a lista de habilidades. Como este número pode ser grande, imprima o resultado módulo $10^9 + 7$.

Exemplos

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

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

Explicação do exemplo 2

A lista pode ser montada de três formas: $(1, 2, 3, 4)$, $(1, 3, 2, 4)$ e $(1, 2, 4, 3)$.

Exemplo de entrada 3	**Exemplo de saída 3**
5 0 1 1 1 1	24

Problema I. Igualdade de Figurinhas

Tempo limite: 1000 ms
Memória limite: 256 MiB

Carlos conseguiu comprar um lote raro de figurinhas da Copa do Mundo para seus dois filhos, Lucas e Matheus. Cada figurinha possui um "score" correspondente à habilidade do jogador (um valor numérico), e algumas figurinhas podem ter o mesmo score. Para evitar qualquer tipo de briga na hora de colar no álbum, Carlos decidiu dividir o monte de forma justa, garantindo que cada filho receba um grupo de figurinhas cuja soma total dos scores seja exatamente igual. Dado um número inteiro N e uma lista com os scores das figurinhas que Carlos possui, determine se é possível dividir essas figurinhas em dois grupos com a mesma soma de valores.

Entrada

A entrada é um número inteiro N $(1 \leq N \leq 200)$, em seguida uma linha com N inteiros v_i $(1 \leq v_i \leq 1000)$ representando os scores de cada figurinha, separados por espaços.

Saída

A saída deve ser a palavra "`Sim`" se for possível dividir as figurinhas em dois grupos com igual soma de scores e a palavra "`Nao`" caso contrário.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
3 5 15 10	Sim

Explicação do exemplo 1

Carlos pode dar a figurinha de valor 15 para um filho e as figurinhas de valores 10 e 5 para o outro. Dessa forma, ambos ficarão com um conjunto de figurinhas cuja soma total dos valores é igual a 15.

Exemplo de entrada 2	Exemplo de saída 2
5 10 10 10 10 10	Nao

Explicação do exemplo 2

Não importa como Carlos rearranje as figurinhas, a soma dos valores das figurinhas de um filho sempre terá uma diferença que é múltipla de 10 em relação à soma dos valores das figurinhas do outro filho.

Exemplo de entrada 3	Exemplo de saída 3
2 10 10	Sim

Exemplo de entrada 4	Exemplo de saída 4
4 10 10 10 9	Nao

Problema J. Jornada até a Taça

Tempo limite: 2000 ms
Memória limite: 256 MiB

A Copa do Mundo finalmente chegou, e ela está espalhada por N cidades-sede ligadas por M rodovias. No centro de toda a festa está a estrela maior: a lendária Taça, de ouro maciço, que precisa percorrer o país aparecendo em cada estádio antes das partidas.

Acontece que transportar a Taça não é nada simples. Em cada rodovia há uma quantidade absurda de torcedores: trânsito, fogos e gente fantasiada por todo lado. A organização mediu, para cada rodovia, um índice de caos — quanto maior, mais tumultuada (e arriscada) é a travessia.

O caos de uma rota é o maior índice de caos entre todas as rodovias que ela utiliza, afinal basta um único trecho caótico para colocar a Taça em apuros. Antes de cada jogo, a comissão precisa levar a Taça de uma cidade a outra e quer escolher a rota cujo caos (o pior trecho) seja o menor possível.

São Q transportes a planejar. Para cada um, descubra o menor caos possível de uma rota entre as duas cidades. Se for impossível chegar de uma à outra pelas rodovias, a Taça irá de helicóptero: nesse caso, responda -1.

Entrada

A primeira linha contém três inteiros N, M e Q ($1 \le N \le 2 \cdot 10^5$; $0 \le M \le 2 \cdot 10^5$; $1 \le Q \le 2 \cdot 10^5$): o número de cidades-sede, o número de rodovias e o número de transportes a planejar.

Cada uma das próximas M linhas contém três inteiros u, v e c ($1 \le u, v \le N$; $u \neq v$; $1 \le c \le 10^9$), indicando uma rodovia de mão dupla entre as cidades u e v com índice de caos c. Pode haver mais de uma rodovia ligando o mesmo par de cidades.

Cada uma das próximas Q linhas contém dois inteiros s e t ($1 \le s, t \le N$): a cidade de origem e a de destino de um transporte da Taça.

Saída

Para cada transporte, imprima uma linha com o menor caos possível de uma rota da cidade s até a cidade t, ou -1 caso não exista rota entre elas.

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
3 3 3 1 2 10 2 3 20 1 3 15 1 3 1 2 2 3	15 10 15

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

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

Exemplo de entrada 3	**Exemplo de saída 3**
`2 0 1` `1 2`	`-1`

Problema K. Kurt e o maldito VAR

Tempo limite: 1000 ms
Memória limite: 256 MiB

Durante a final da Copa do Mundo de 2026, o árbitro Kurt, cameleão do PPCI, resolveu utilizar o VAR em excesso, de forma a interromper o jogo várias e várias vezes. Cada vez que o VAR é acionado em um minuto x_i, *todo* aquele minuto fica paralisado, dividindo o jogo em intervalos de tempos jogados sem pausas.

Um analista do jogo quer saber, após cada paralisação ocorrida: qual é o maior tempo que os jogadores jogaram sem que a partida fosse interrompida após aquela paralisação?

Entrada

A primeira linha contém dois inteiros: N $(1 \leq N \leq 10^9)$, a duração total da partida em minutos, e Q $(1 \leq Q \leq 2 \times 10^5)$, a quantidade de paralisações ocorridas.

A segunda linha contém Q inteiros distintos x_1, x_2, ..., x_Q (com $1 \leq x_i \leq N$), indicando os minutos nos quais o VAR foi acionado, em qualquer ordem.

Saída

Para cada paralisação, imprima um inteiro contendo o tamanho do maior intervalo contínuo de minutos sem paralisações após aquela paralisação.

Imprima a resposta referente às paralisações na mesma ordem dada na entrada.

Exemplos

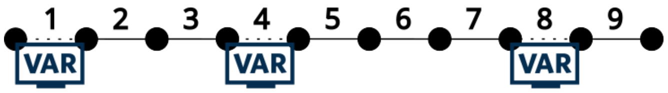
Exemplo de entrada 1	Exemplo de saída 1
9 3 8 1 4	1 3 3

Exemplo de entrada 2	Exemplo de saída 2
8 4 4 2 6 3	2 2 2 2

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

Observações

A figura abaixo apresenta a minutagem do jogo do primeiro exemplo de entrada:



Após a paralisação no minuto 8, o maior intervalo de tempo contínuo jogado é de 1 minuto (minuto 9); Após a paralisação no minuto 1, o maior intervalo de tempo contínuo jogado é de 3 minutos (minutos 5 a 7); e após a paralisação no minuto 4, o maior intervalo de tempo contínuo jogado também é de 3 minutos (idem).

Problema L. Lado ou Bola

Tempo limite: 1000 ms
Memória limite: 256 MiB

Este é um problema interativo.

Chegou a hora da estreia da seleção da Nlogônia na Copa do Mundo! Você, que é o capitão da equipe, não poderia estar mais ansioso para este clássico mundial: Nlogônia vs. Nfatorialônia!

O primeiro passo do jogo é decidir qual equipe vai escolher seu lado de campo, e qual equipe vai começar com a bola. Tradicionalmente isto é decidido com um simples "cara ou coroa" feito pelo árbrito com os capitães, onde o capitão que vence o "cara ou coroa" tem o poder de escolha de lado ou bola. Mas dessa vez, o árbitro resolveu fazer um jogo diferente. Ele colocou N buracos no campo, ligados por M conexões diretas (de sentido único) entre si. Ele também espalhou K moedas entre os N buracos.

Os capitães jogam em turnos alternados, e você é o primeiro a jogar. A cada turno, o jogador deverá escolher alguma moeda que está em algum buraco, e movê-la para outro buraco através de uma conexão direta. O jogador que não puder mais fazer uma jogada perde o jogo. O árbitro garantiu que não há ciclos no campo e, logo, o jogo terminará em um número finito de turnos.

O capitão da Nfatorialônia é esperto, e fará o possível para tentar vencer. Dada a configuração inicial do jogo, escolha a melhor estratégia e vença o capitão adversário!

É garantido que a configuração inicial do jogo é dada de tal forma que você consegue vencer, se seguir uma estratégia ótima.

Interação

Seu programa deve começar lendo a descrição do campo. A primeira linha da descrição contém dois inteiros N e M ($1 \leq N \leq 10^5$, $0 \leq M \leq 2 \times 10^5$), o número de buracos e o número de conexões diretas. Os buracos são numerados de 1 a N.

As próximas M linhas descrevem as conexões diretas. Cada linha contém dois inteiros A e B ($1 \leq A, B \leq N, A \neq B$) descrevendo uma conexão direta do buraco A para o buraco B.

A próxima linha contém um inteiro K ($1 \leq K \leq 20$), o número de moedas. A linha seguinte contém uma sequência de K inteiros b_i ($1 \leq b_i \leq N$) separados por espaço, indicando que a moeda i inicialmente está no buraco b_i. Um buraco pode conter mais de uma moeda.

Após ler a descrição do campo, o jogo irá começar. No seu turno, imprima uma linha com `!` (exclamação) seguido de dois inteiros A e B, indicando que sua jogada consiste em mover uma moeda do buraco A para o buraco B através de uma conexão direta.

Após sua jogada, o capitão adversário irá jogar. Seu programa deve ler uma linha contendo dois inteiros A e B, indicando que o adversário moveu uma moeda do buraco A para o buraco B através de uma conexão direta.

Após a jogada do adversário, volta a ser o seu turno, e a interação se repete.

Quando o adversário não puder mais jogar (e, portanto, você venceu), seu programa irá receber a linha `-1 -1` no lugar da jogada do adversário (isto é, a jogada adversária será $A = -1$ e $B = -1$). Ao ler esta jogada, seu programa deve encerrar, sem produzir mais saída.

Se você fizer alguma jogada inválida (formato incorreto; A ou $B \notin [1..N]$; não há conexão direta de A para B; ou não há moeda em A no momento), a interação será encerrada imediatamente, e você receberá o veredito `WRONG ANSWER`. Faça uma jogada inválida para desistir do jogo.

Certifique-se de fazer *flush* na saída após cada linha impressa. Você pode usar: `fflush(stdout)` em C/C++; `cout.flush()` em C++; `System.out.flush()` em Java; e `stdout.flush()` em Python.

Exemplos

Exemplo de interação 1

Leitura	Escrita
3 3	
1 2	
1 3	
2 3	
2	
1 2	
	! 1 2
2 3	
	! 2 3
-1 -1	

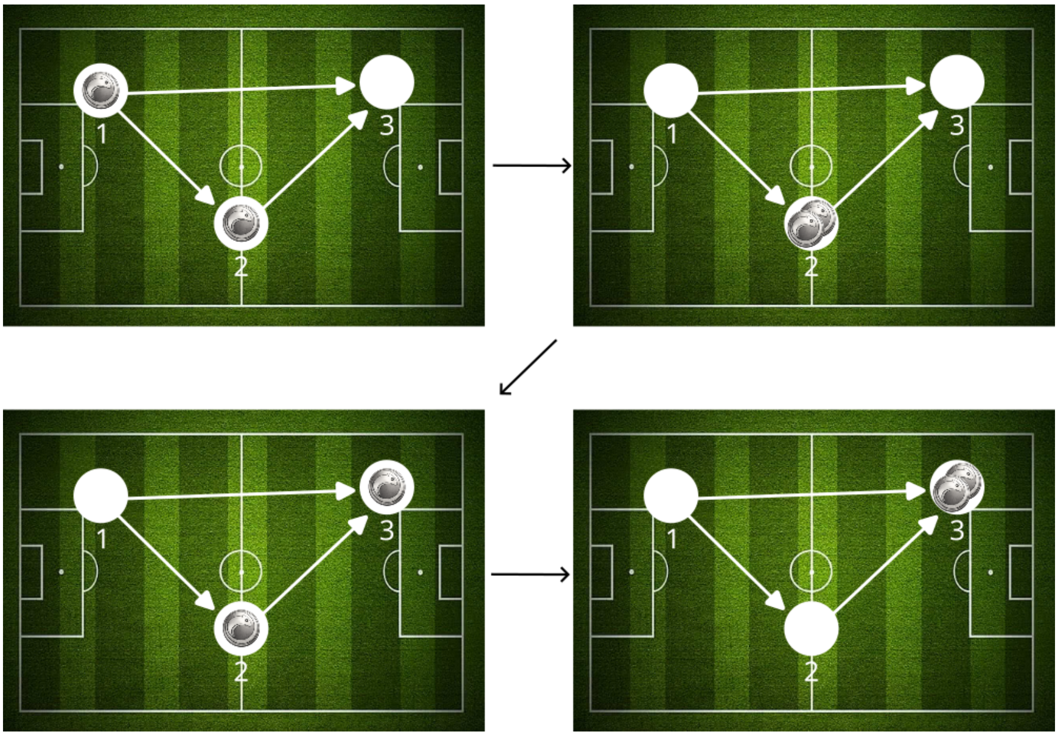
Exemplo de interação 2

Leitura	Escrita
4 3	
1 2	
2 3	
3 4	
3	
1 1 1	
	! 1 2
1 2	
	! 1 2
2 3	
	! 3 4
2 3	

	`! 3 4`
`2 3`	
	`! 3 4`
`-1 -1`	

Observações

A figura abaixo ilustra o primeiro exemplo de interação:



Problema M. Metades Iguais

Tempo limite: 2000 ms
Memória limite: 256 MiB

A Copa está intensa: Brasil em campo, torcida gritando, fogos de artifício no ar e churrasquinho rolando. Enquanto isso, Ancelotti passou uma tarefa para o Endrick que, pra variar, estava no banco de reservas, e disse que, caso ele conseguisse resolvê-la, teria uma chance de entrar em campo.

Só que o Endrick não é lá muito habilidoso em programação competitiva; na verdade, ele é péssimo, algo que lembra os seus companheiros de equipe Rafinha e Casemiro jogando bola. Por isso, o Endrick pediu para você ajudá-lo a resolver o problema e, quem sabe, poder entrar em campo.

São dados m conjuntos de inteiros positivos, cada um com uma quantidade par de elementos.

Será que dá para distribuir todos esses inteiros em dois grupos E e D? Cada elemento de cada conjunto vai para exatamente um dos grupos (nunca para os dois) e, de cada conjunto, exatamente metade dos elementos deve ir para E e a outra metade para D.

Ao final, os grupos E e D devem ser iguais enquanto multiconjuntos, isto é, cada valor deve aparecer a mesma quantidade de vezes em E e em D.

Ajude o Endrick: encontre uma distribuição que satisfaça todas essas condições ou determine que ela não existe.

Entrada

A primeira linha contém um inteiro m ($1 \leq m \leq 10^5$), o número de conjuntos.

As $2 \cdot m$ linhas seguintes descrevem os conjuntos. Cada conjunto é dado em duas linhas: a primeira contém um inteiro par n ($2 \leq n \leq 2 \times 10^5$), a quantidade de elementos; a segunda contém n inteiros $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$).

A soma dos valores de n em todos os conjuntos não excede 2×10^5.

Saída

Se existe uma distribuição, imprima `SIM` e, em seguida, m linhas.

Na i-ésima dessas linhas, imprima uma string com exatamente n caracteres, sem espaços, onde cada caractere é `E` ou `D` e indica para qual grupo vai o elemento correspondente do i-ésimo conjunto, na ordem em que os elementos aparecem.

Caso não exista nenhuma distribuição válida, imprima `NAO`.

Se houver mais de uma distribuição válida, qualquer uma delas será aceita.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
3 2 1 2 4 1 2 3 3 6 1 1 2 2 3 3	SIM DE EDED DEEDDE