2026
Caderno de Problemas
XIII Maratona Mineira de Programação

Patrocínio

Realização

Organização

Problema A. Ana, viva a Mariana!

Viva a Mariana!

Mariana conta 1 é uma música muito usada no processo de numeramento de crianças, já que apresenta uma forma didática e divertida para contar. Para crianças mais velhas, a música foi generalizada para Mariana Conta X (sendo X um número natural), e segue da seguinte maneira:

Mariana conta 1
Mariana conta 1*:* *é* 1*,* *é Ana*
Viva a Mariana, viva a Mariana!

Mariana conta 2
Mariana conta 2*:* *é* 1*,* *é* 2*,* *é Ana*
Viva a Mariana, viva a Mariana!

...

Mariana conta X
Mariana conta X*:* *é* 1*,* *é* 2*,* . . . *,* *é* X*,* *é Ana*
Viva a Mariana, viva a Mariana!

Para saber seu valor didático, imprima quantos números estão presentes na música.

Entrada

A entrada é composta por um inteiro X $(1 \leq X \leq 1000)$.

Saída

A saída deve conter um inteiro: quantos números foram contados na música.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2	7

Explicação do exemplo 1

Mariana contou 7 números: quatro vezes o número 1 e três vezes o número 2, como pode ser visto nas duas primeiras estrofes do enunciado.

Exemplo de entrada 2	Exemplo de saída 2
5	25

Exemplo de entrada 3	Exemplo de saída 3
10	75

Problema B. Bário World

It's-a-Me, Bário!

Por conta de suas divertidas mecânicas, Super Bário World é um sucesso entre os gamers mineiros. No jogo, Bário se move em fases compostas por N blocos, cada um podendo ser sólido ou um buraco e o objetivo do jogo é chegar do início da fase (o bloco 1) até o final (o bloco N) sem cair em nenhum buraco. Bário pode correr sobre os blocos sólidos e, para evitar os buracos, Bário pode pulá-los.

Bário usa botas que têm uma reserva de energia, que inicialmente é 1. Ao correr ele carrega as botas para poder pular mais longe – cada bloco sobre o qual Bário corre adiciona uma unidade de carga a seu equipamento. Um pulo de carga v faz com que Bário vá do bloco b para qualquer bloco entre $b + 1$ e $b + v$; ao aterrissar, Bário perde energia de suas botas, retornando ao valor inicial de 1.

Calcule o número mínimo de pulos para Bário chegar na posição N ou indique que é impossível.

Entrada

A primeira linha contém um inteiro T ($1 \leq T \leq 5 \times 10^5$) o número de casos de teste (é garantido que o somatório dos valores de N é menor que 5×10^5).

Depois disso seguem T pares de linhas. A primeira linha contém N ($2 \leq N \leq 5 \times 10^5$), o comprimento da fase, e a segunda contém uma string de N caracteres representando os blocos da fase, `x` se for um bloco sólido e `.` se for um buraco (é garantido que os blocos 1 e N são sólidos).

Saída

Imprima o número mínimo de pulos para Bário chegar no fim da fase ou -1 se não for possível.

Exemplos

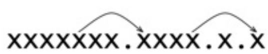
Exemplo de entrada 1	Exemplo de saída 1

`2`
`16`
`xxxxxxx.xxxx.x.x`
`3`
`x.x`
	`2`
	`-1`

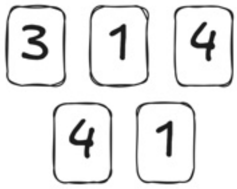
Explicação do exemplo 1

No primeiro caso de teste, uma possível solução com apenas dois pulos envolve Bário:

- Correr da posição 1 até a posição 5, carregando as botas com energia 5 e dar um pulo aterrisando na posição 9. Ao aterrisar suas botas voltam a ter energia 1.
- Correr da posição 9 até a posição 12 carregando as botas com energia 4 e pular para o fim.



Problema C. Cartas



Certo dia, Rened decidiu convidar seus amigos para um jogo de cartas. O jogo consiste em uma sequência de N cartas numeradas colocadas sobre uma mesa, inicialmente vazia. Os jogadores, em turnos, vão adicionando novas cartas ao final dessa sequência, sendo que o i-ésimo jogador irá jogar a carta de valor C_i.

A regra principal é simples: **não pode haver cartas com números repetidos sobre a mesa**.

Sempre que um jogador joga uma carta cujo número ainda não apareceu na sequência atual, ela é apenas adicionada ao final, aumentando a sequência normalmente.

Porém, se um jogador jogar uma carta de mesmo valor que outra já presente na mesa, ocorre uma eliminação: todas as cartas desde o início da sequência até a carta com este número (inclusive a carta de número repetido) são removidas. Em seguida, a nova carta jogada passa a ocupar a posição final da mesa. **Note que após este processo não restam mais valores repetidos na mesa e que a sequência nunca fica vazia.**

Para deixar o jogo mais rápido, Rened solicitou que você e sua equipe desenvolvam um sistema que acompanhe o estado do jogo, informando, ao final de cada jogada, a carta de maior valor presente na mesa e o índice do jogador que a jogou.

Entrada

A entrada tem duas linhas. Na primeira é dado o valor de N $(1 \leq N \leq 2 \times 10^5)$, representando a quantidade de jogadores. Na segunda, são informados os valores C_i $(1 \leq C_i \leq N)$ das cartas.

Saída

A saída deve conter N linhas, cada uma contendo dois valores separados por um espaço. O valor C da maior carta ainda presente na mesa e o índice i do jogador que jogou esta carta.

Exemplos

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

Explicação do exemplo 1

1ª jogada:
Jogador 1 joga 3
Cartas na mesa: [3]
Maior carta: 3 (jogador 1)

2ª jogada:
Jogador 2 joga 1
Cartas na mesa: [3, 1]
Maior carta: 3 (jogador 1)

3ª jogada:
Jogador 3 joga 4
Cartas na mesa: [3, 1, 4]
Maior carta: 4 (jogador 3)

4ª jogada:
Jogador 4 joga 1
Elimina-se o prefixo até o 1 anterior
Cartas na mesa: [4, 1]
Maior carta: 4 (jogador 3)

5ª jogada:
Jogador 5 joga 6
Cartas na mesa: [4, 1, 6]
Maior carta: 6 (jogador 5)

6ª jogada:
Jogador 6 joga 6
Cartas na mesa: [6]
Maior carta: 6 (jogador 6)

Exemplo de entrada 2

```
12
1 1 4 6 5 8 4 6 8 2 5 2
```

Exemplo de saída 2

```
1 1
1 2
4 3
6 4
6 4
8 6
8 6
8 6
8 9
8 9
8 9
5 11
```

Problema D. Datas incertas

A NeoSpace é uma startup inovadora que utiliza modelos avançados de Inteligência Artificial para processar volumes massivos de informações corporativas. Nos corredores da empresa, é comum ouvir uma piada clássica entre os engenheiros: apesar de serem especialistas em lidar com Big Data, o maior pesadelo do sistema ainda são as pequenas datas inseridas incorretamente pelos usuários.

Recentemente, a plataforma começou a integrar dados de novos clientes internacionais. Durante a ingestão de dados, a equipe notou um problema grave na interface: enquanto o sistema espera o padrão brasileiro de Dia/Mês (DD/MM), muitos usuários estrangeiros estão preenchendo os formulários no formato Mês/Dia (MM/DD).

Para garantir a qualidade do modelo de IA, o pipeline deve identificar quando uma entrada pode causar confusão. Por exemplo, o valor 1/6 é problemático, pois pode ser interpretado como 1º de Junho ou 6 de Janeiro. Já o valor 28/2 é seguro, pois não existe um mês 28 no calendário.

Sua tarefa é criar um algoritmo que analise a entrada fornecida pelo usuário e determine se os números informados podem gerar essa dupla interpretação.

Entrada

A entrada é composta por uma única linha contendo dois números inteiros D e M ($1 \leq D \leq 28, 1 \leq M \leq 12$) representando, respectivamente, o dia e o mês do evento.

Saída

Imprima `DATA INCERTA` se a data pode ser interpretada no formato Mês/Dia como uma outra data válida e `DATA SEGURA` caso contrário.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
`28 2`	`DATA SEGURA`

Exemplo de entrada 2	Exemplo de saída 2
`1 6`	`DATA INCERTA`

Exemplo de entrada 3	Exemplo de saída 3
`9 9`	`DATA SEGURA`

Explicação do exemplo 3

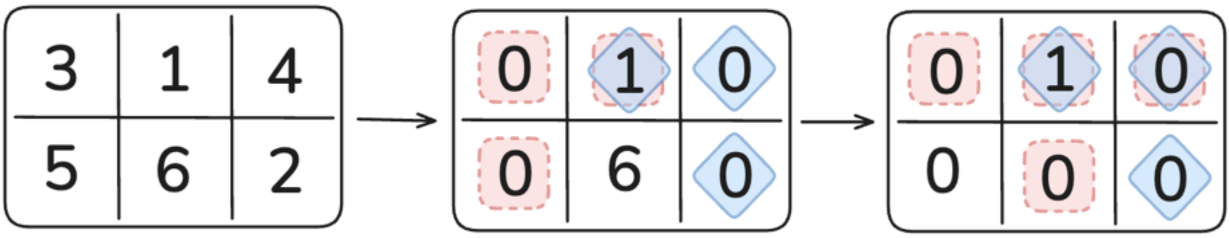
A data pode ser interpretada apenas como 9 de Setembro.

Problema E. Explorando o terreno

Giovana e Arthur acabaram de ser contratados pela ArcelorMittal Sistemas para desenvolver tecnologia de ponta na extração mineral. Como projeto inicial, ambos implementaram algoritmos e heurísticas para criar um plano de exploração de uma perfuratriz, máquina muito comum nas lavras minerais, objetivando maximizar o minério extraído após T minutos de operação da perfuratriz. A lavra sobre a qual Giovana e Arthur estão trabalhando é representada por uma matriz $N \times M$, onde cada célula contém o valor adquirido caso a posição seja escavada pela perfuratriz. Cada plano de exploração é composto por T pares de coordenadas $(i_1, j_1), \ldots, (i_T, j_T)$ sendo (i_t, j_t) a linha e coluna onde a perfuratriz estará no momento t.

Muito satisfeitos com as projeções dos algoritmos de Giovana e Arthur, os diretores da empresa decidiram que ambos os planos de exploração serão executados simultaneamente, cada um com sua própria perfuratriz. O problema é que os planos de exploração de Giovana e Arthur não foram feitos levando em consideração a presença da outra perfuratriz, o que poderia causar conflitos na extração do minério!

Felizmente, o software de controle desenvolvido pela ArcelorMittal Sistemas é bastante sofisticado, e consegue lidar com esse tipo de situação! Além do controle mecânico das perfuratrizes, ele estabelece uma rede de comunicação entre os equipamentos, o que impede que as perfuratrizes explorem a mesma posição simultaneamente. Desta forma, estando na posição (i, j), a perfuratriz é capaz de extrair minério da célula (i, j) e das células $(i-1, j), (i, j-1), (i, j+1), (i+1, j)$, mas só o faz se a outra perfuratriz não fará a extração naquela célula naquele momento. A extração é 100% eficiente, ou seja, após ter seu minério extraído, uma célula passa a ter zero unidades de minério. No exemplo da figura abaixo, se no instante t a perfuratriz de Giovana estiver na posição $(1, 1)$ e a de Arthur na posição $(1, 3)$, **nenhuma das perfuratrizes extrairá minério da célula** $(1, 2)$**, deixando todo o minério que lá havia intacto**. Por outro lado, todas as outras células de interesse de cada perfuratriz terão seu minério extraído.



Representação visual do Exemplo 1. À esquerda, a matriz com as quantidades de minério em cada célula. Ao centro, as células exploradas por cada perfuratriz no instante $t = 1$; em quadrados vermelhos as células exploradas por Giovana e em diamantes azul as células exploradas por Arthur. À direita, as células exploradas por cada perfuratriz no instante $t = 2$;

Curiosos em saber como seus planos irão performar, Giovana e Arthur pediram sua ajuda. Escreva um programa que, dada a matriz que representa a lavra e os planos de exploração de cada um, imprima o total de minério extraído pela perfuratriz de cada um. É garantido que ambos os planos de exploração têm a mesma duração T.

Entrada

A primeira linha contém três inteiros, N, M e T, ($1 \leq N, M \leq 500$, $1 \leq T \leq N \times M$), o número de linhas e colunas da matriz e a duração dos planos de exploração, respectivamente.

As próximas N linhas contêm M inteiros cada, ($1 \leq v_1, \ldots, v_M \leq 1000$). O j-ésimo número da i-ésima linha contém o valor de cada minério presente na posição (i, j).

As próximas T linhas contém dois inteiros cada. A t-ésima dessas linhas contém $(i_t, j_t)_G$ ($1 \leq i \leq N$, $1 \leq j \leq M$) a posição onde a perfuratriz de Giovana estará no instante t.

Por fim, as últimas T linhas contém dois inteiros cada. A t-ésima dessas linhas contém $(i_t, j_t)_A$ ($1 \leq i \leq N$, $1 \leq j \leq M$) a posição onde a perfuratriz de Arthur estará no instante t.

Saída

Imprima dois números G e A separados por espaço – a soma total dos valores extraídos por Giovana e Arthur, respectivamente.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2 3 2 3 1 4 5 6 2 1 1 1 2 1 3 1 3	14 6

Exemplo de entrada 2	Exemplo de saída 2
3 1 3 10 1 10 2 1 3 1 1 1 2 1 2 1 2 1	0 20

Exemplo de entrada 3	Exemplo de saída 3
1 2 2 6 7 1 2 1 1 1 1 1 2	0 0

Problema F. Fechando estradas

Dentre as várias delícias de Minas, queijo e doce de leite são as mais famosas. Apesar de serem feitas da mesma matéria-prima, o leite, seus processos de produção e armazenamento são bastante diferentes. Em particular, todo o queijo do estado é produzido na cidade A_1 e armazenado na cidade A_2, enquanto todo o doce de leite é produzido na cidade B_1 e armazenado na cidade B_2. Para levar os estoques de uma cidade a outra, cada produto é transportado em uma das M estradas de mão dupla, cada uma delas ligando exatamente duas das N cidades mineiras.

Além de sua importância culinária, o queijo e o doce de leite são fundamentais para a estabilidade política e social no estado: a falta de ambas as iguarias por um curto período já é capaz de transformar nossa característica hospitalidade em animosidade profunda. Preocupado com essa possibilidade, o Movimento Mineiro Pacifista (MMP) precisa saber: qual a menor quantidade de estradas que, se fechadas, impediriam o transporte do queijo de A_1 para A_2 e do doce de leite de B_1 para B_2? Escreva um programa para ajudar o MMP a encontrar essa resposta.

Entrada

A primeira linha da entrada contém dois inteiros N e M ($4 \le N \le 2000$, $0 \le M \le 2000$), indicando o número de cidades e estradas de Minas, respectivamente.

Cada uma das M linhas seguintes contém dois inteiros u e v ($1 \le u, v \le N$, $u \ne v$), indicando que existe uma estrada entre as cidades u e v. É garantido que não há mais de uma estrada entre o mesmo par de cidades.

Após essas M linhas, há uma última linha contendo quatro inteiros distintos A_1, A_2, B_1 e B_2, indicando as cidades de produção e armazenamento do queijo e do doce de leite, respectivamente.

Saída

A saída deve conter um único inteiro, indicando o menor tamanho de um conjunto de estradas que, após fechadas, impedem que nossas delícias mineiras sejam transportadas.

Exemplos

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

Explicação do exemplo 1

Neste exemplo, se a estrada que liga as cidades 3 e 4 for interditada, não haverá mais caminho entre as cidades 1 e 5, nem entre 2 e 6.

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

Explicação do exemplo 2

Neste exemplo, se as estradas entre 1 e 2, e entre 4 e 5 forem interditadas, isso impediria tanto a existência de um caminho entre 1 e 3 quanto de 4 para 6. É possível provar que essa é a quantidade mínima de estradas fechadas para que isso aconteça.

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

Explicação do exemplo 3

Neste exemplo, já é impossível ir da cidade 1 até a cidade 2. Portanto, se a estrada de 3 para 4 for interditada, fica impossível tanto ir de 1 para 2 quanto de 4 para 5.

Problema G. Grupos de camisas

Após cada maratona, cabe a nós contemplar tudo aquilo que vivemos: problemas passados, balões conquistados, as amizades que fizemos pelo caminho, e as camisas usadas durante a competição.

Na figura acima, temos quatro camisas, usando sete pregadores, formando três grupos de camisas.

Para pregar uma camisa no varal, devemos usar dois pregadores (um em cada ponta da camisa). Por outro lado, com duas camisas, podemos usar apenas três pregadores, pregando as pontas de duas camisas consecutivas com um mesmo pregador. Quando isso acontece, dizemos que foi formado um grupo de camisas e que as camisas pregadas por um pregador pertencem ao mesmo grupo. Estar em um grupo é uma propriedade transitiva, ou seja, se a camisa A pertence ao mesmo grupo da camisa B, e a camisa B pertence ao mesmo grupo da camisa C, então as camisas A e C também pertencem ao mesmo grupo.

Dilson quer pregar N camisas no varal e ao mesmo tempo quer que sejam formados exatamente G grupos de camisas. Qual a menor quantidade de pregadores que Dilson precisa utilizar para alcançar seu objetivo?

Entrada

A única linha de entrada contém dois inteiros N $(1 \leq N \leq 10^6)$ e G $(1 \leq G \leq N)$, o número de camisas que Dilson quer pregar e o número de grupos que ele quer formar, respectivamente.

Saída

A saída deve conter um único inteiro, que o número de pregadores que Dilson precisa utilizar para pregar N camisas em G grupos.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
`4 3`	`7`

Explicação do exemplo 1

Como na figura, Dilson deve usar sete pregadores para formar três grupos.

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

Explicação do exemplo 2

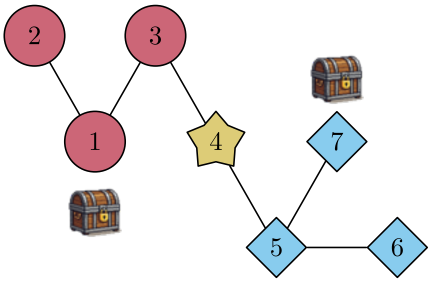
Dilson deve usar três pregadores: um na ponta da primeira camiseta, um pregando a outra ponta da primeira camiseta e uma ponta da segunda camiseta, e um na outra ponta da segunda camiseta. Assim, as duas camisetas ficam pregadas por um mesmo pregador, formando um grupo de camisas.

Problema H. Herança cultural

A equipe da Maratona Mineira está organizando uma caça ao tesouro. O tesouro serão dois baús do mais delicioso doce de leite do mundo, herança cultural de Minas Gerais, que serão escondidos no mapa do jogo.

O mapa é composto por N vértices, numerados de 1 a N, e por $N - 1$ arestas, de modo que é possível ir de qualquer vértice a qualquer outro. Em outras palavras, o mapa forma uma árvore. A distância entre dois vértices é definida como o número de arestas no único caminho simples entre eles.

A equipe já possui o mapa, mas ainda precisa escolher em quais vértices esconder os dois baús. Uma escolha é considerada válida se os baús forem colocados em vértices distintos e se o número de vértices mais próximos do primeiro baú for igual ao número de vértices mais próximos do segundo baú. Vértices que estejam à mesma distância de ambos os baús não contam para nenhum dos dois.



A figura acima mostra uma escolha válida para o exemplo 1. Os baús foram colocados nos vértices 1 e 7. Dessa maneira, os vértices 1, 2 e 3 (círculos vermelhos) estão mais próximos do baú que está no vértice 1, enquanto os vértices 5, 6 e 7 (losângulos azuis) estão mais próximos do baú no vértice 7. O vértice 4 (estrela amarela) está à mesma distância de ambos os baús.

Determine o número de maneiras válidas de escolher as posições dos baús. Duas maneiras são consideradas distintas se o conjunto de vértices onde um baú foi colocado é diferente.

Entrada

A primeira linha da entrada contém um inteiro N ($2 \leq N \leq 2 \times 10^5$), o número de vértices do mapa.

Cada uma das próximas $N-1$ linhas contém dois inteiros u_i e v_i ($1 \leq u_i, v_i \leq N$, $u_i \neq v_i$), indicando que existe uma aresta entre os vértices u_i e v_i. É garantido que as arestas formam uma árvore.

Saída

Imprima um único inteiro: o número de maneiras válidas distintas de escolher as posições dos dois baús.

Exemplos

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

Explicação do exemplo 1

As 4 escolhas válidas, representadas pelas posições dos baús, são: $\{1,7\}$, $\{1,6\}$, $\{3,5\}$ e $\{6,7\}$.

Exemplo de entrada 2	Exemplo de saída 2
7 1 2 1 3 3 4 1 5 5 6 6 7	0

Explicação do exemplo 2

Neste exemplo não existem escolhas válidas para as posições dos baús.

Problema I. Idiomas

Durante eras, as criaturas desenvolveram suas próprias formas de falar, cada idioma vinha carregado de toda cultura e estilo de vida das raças. Os Elfos criaram o élfico, uma língua mais suave e elegante, os dragões criaram o dracônico que representava sua dominância e ferocidade e, aos poucos, todas as outras raças criaram suas próprias línguas ao seu bel prazer.

Uma grande reunião entre diplomatas de N raças será realizada para discutir o futuro do continente. O evento conta com um total de M línguas catalogadas. O problema é que nem todos os diplomatas compartilham uma língua em comum. Para que a mensagem chegue de um ponto a outro, pode ser necessário o uso de intermediários.

Contudo, cada vez que a mensagem precisa ser traduzida de uma língua para outra (quando um intermediário ouve em uma língua e fala em outra), ocorre uma **troca de língua**. Para minimizar desentendimentos, a organização do conclave precisa descobrir o número mínimo de trocas necessárias para que dois diplomatas consigam se entender.

Dados os conjuntos de línguas faladas por cada um dos N diplomatas e Q consultas (a, b), determine o número mínimo de trocas de língua para que os diplomatas a e b se comuniquem. Se a comunicação for impossível, imprima -1.

Entrada

A primeira linha contém três inteiros N, M e Q ($2 \leq N \leq 10^4, 1 \leq M \leq 30, 1 \leq Q \leq 2 \times 10^5$), o número de diplomatas, o número total de línguas e quantas consultas, respectivamente.

Em seguida temos N linhas, cada uma composta primeiramente pelo inteiro m_i ($1 \leq m_i \leq M$) – quantas línguas o i-ésimo diplomata fala, seguido por m_i valores distintos $\ell_1, \ldots, \ell_{m_i}$ ($1 \leq \ell \leq M$) – as línguas faladas pelo diplomata.

Por fim, a entrada contém mais Q linhas cada uma contendo dois inteiros a e b ($1 \leq a, b \leq N, a \neq b$) o diplomata inicial e final na consulta.

Saída

A saída deve conter Q linhas, cada uma respondendo quantas trocas de língua são necessárias para a e b se comunicarem; se não for possível, imprima -1.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
6 7 3 2 3 6 3 3 2 7 2 1 2 2 1 5 1 4 1 1 4 1 2 3 2 5	2 0 -1

Explicação do exemplo 1

- Na primeira consulta uma possível forma de transmitir a mensagem é:
  $4 \xrightarrow{\text{língua 1}} 6 \xrightarrow{\text{língua 1}} 3 \xrightarrow{\text{língua 2}} 2 \xrightarrow{\text{língua 3}} 1$ – tendo duas trocas de língua, 1/2 e 2/3.
- Na segunda consulta a mensagem pode ser transmitida sem trocas de língua $2 \xrightarrow{\text{língua 2}} 3$.
- Na terceira consulta, o representante 5 fala somente a língua de número 4, que não é falada por mais ninguém, logo não é possível transmitir a mensagem para ele.

Problema J. Jogo do beta

Um grupo de N amigos vai participar de um jogo no qual cada jogador escolhe um número inteiro P_i. Em seguida, deveria ser sorteado um número inteiro X maior que 1, mas Pedro descobriu uma forma de trapacear e pode escolher esse valor.

A pontuação de cada jogador é dada por $P_i \bmod X$, ou seja, o resto da divisão de P_i por X.

Cada jogador que tenha sua pontuação igual a 0 é chamado de *beta*, pois não sobra nada da divisão.

Ajude Pedro a escolher um valor de X que maximize a quantidade de jogadores beta.

Entrada

A primeira linha da entrada contém um único inteiro N ($1 \leq N \leq 10^5$), representando o número de jogadores.

A segunda linha contém N inteiros P_i ($2 \leq P_i \leq 10^6$), representando o valor escolhido pelo i-ésimo jogador.

Saída

Imprima um único inteiro X ($2 \leq X \leq 10^6$) que maximize o número de jogadores beta.

Em caso de múltiplos valores de X que produzam a mesma quantidade máxima de jogadores beta, imprima qualquer um deles.

Exemplos

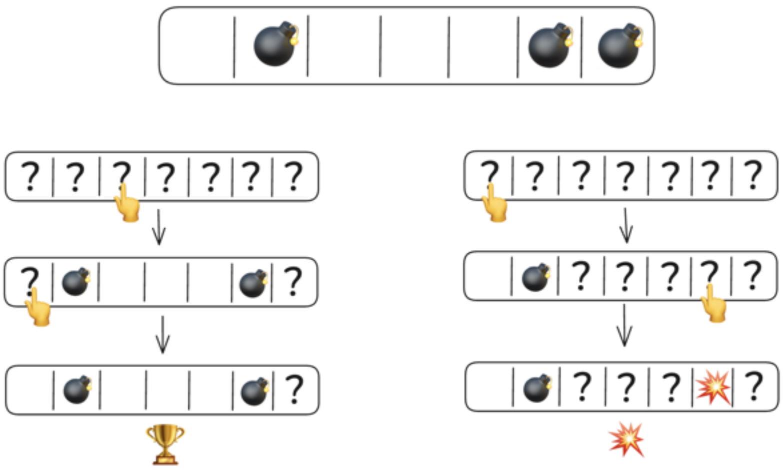
Exemplo de entrada 1	Exemplo de saída 1
5 2 5 4 5 3	2

Exemplo de entrada 2	Exemplo de saída 2
2 19 20	19

Problema K. Kampo Minado

A Universidade do Heptágono Mineiro está criando um novo sistema operacional, o "Janelas OS" que virá instalado com o famoso jogo Campo Minado. Para fugir de problemas legais decidiu-se fazer o Campo Minado 1-D, uma versão diferente o suficiente para acalmar os advogados da concorrência. O jogo funciona da seguinte maneira:

- O campo minado tem N posições e cada posição pode ser vazia ou ter uma mina.
- Cada posição pode estar revelada ou secreta – inicialmente todas são secretas.
- A cada rodada o jogador escolhe uma posição **secreta** e clica nela:
  - Se a posição tiver uma mina, o jogador perde.
  - Se a posição for vazia, ela e todas as casas à esquerda e à direita do clique são reveladas até chegar no fim do campo ou em uma mina (que também é revelada).
- Se em algum momento todas as posições vazias forem reveladas o jogador ganha.



Na figura acima, temos dois exemplos de partida dado um campo minado. A da esquerda termina em vitória pois todas as casas vazias são encontradas e a da direita termina em derrota pois o jogador clica em uma mina.

Para ajudar a universidade a entender por quanto tempo o jogo pode entreter os usuários, dado N, imprima quantas partidas diferentes podem ser jogadas em um campo de tamanho N. Duas partidas são consideradas diferentes se o campo minado é diferente ou se alguma das jogadas foi diferente em ordem ou posição. Como o número de partidas pode ser muito grande, imprima a resposta módulo P – ou seja, o resto da divisão por P.

Entrada

A entrada contém dois inteiros N ($1 \leq N \leq 3000$) e P primo ($5 \times 10^8 \leq P \leq 10^9$), o tamanho do campo e o módulo em que a resposta deve ser impressa, respectivamente.

Saída

A saída deve conter um inteiro, o número de partidas possíveis módulo P.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
`2 500000003`	`7`

Explicação do exemplo 1

As partidas possíveis são:

- Campo `BB` - 1 partida possível, o jogador já começa ganhando.
- Campo `B.` - 2 partidas possíveis, o jogador clica na posição da esquerda e perde, ou na da direita e ganha.
- Campo `.B` - 2 partidas possíveis, o jogador clica na posição da esquerda e ganha, ou na da direita e perde.
- Campo `..` - 2 partidas possíveis, o jogador clica na posição da esquerda ou da direita e ganha.

Exemplo de entrada 2	Exemplo de saída 2
`5 500000003`	`205`

Exemplo de entrada 3	Exemplo de saída 3
`3000 999999733`	`842327137`

Problema L. Lavando a louça

Após um excelente café com pão de queijo em família, Beto ficou encarregado de lavar a louça. Após ensaboar tudo, Beto se lembrou o quão chato é colocar os copos no escorredor linear de sua casa, em particular, a trabalheira que é colocar um copo entre dois outros que já estão secando. Ele estimou que o esforço para colocar um copo em uma determinada posição no escorredor é igual à soma do número de copos consecutivos à esquerda e à direita da posição.

Na figura acima, temos o estado final de uma solução para o Exemplo 1 – 4 copos em um escorredor de 6 posições.

Dado o tamanho N do escorredor e a quantidade C de copos, calcule qual o esforço total mínimo que Beto deve fazer para colocar todos os copos no escorredor e dê uma sequência de inserções que atinja esse esforço mínimo; as posições do escorredor são numeradas de 1 a N. Se houver mais de uma maneira de atingir o esforço mínimo, imprima qualquer uma delas.

Entrada

A única linha da entrada contém dois inteiros N ($2 \le N \le 10^6$) e C ($1 \le C \le N$), o tamanho do escorredor e a quantidade de copos, respectivamente.

Saída

A primeira linha da saída deve conter um único inteiro, o esforço total mínimo que Beto deve fazer para colocar todos os copos no escorredor.

A segunda linha da saída deve conter C inteiros separados por espaço, onde o i-ésimo inteiro representa a posição do escorredor onde o i-ésimo copo deve ser colocado.

Exemplos

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

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

Explicação do exemplo 2

Uma maneira de colocar os copos é [1 2 − 3 − 4 5]. Os copos 1, 3, e 4 possuem custo 0 de colocação, já os copos 2 e 5 possuem custo 1, totalizando um esforço de 2.

Exemplo de entrada 3	Exemplo de saída 3
15 14	20 1 3 2 5 7 6 4 9 11 10 13 15 14 12

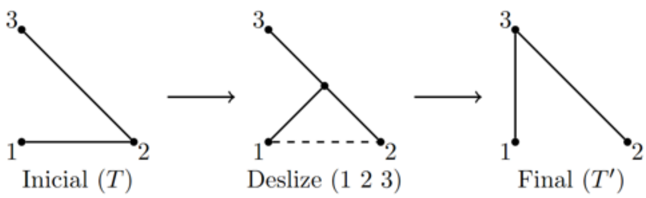
Problema M. Movendo varetas

Os renomados criadores do pega-varetas acabam de lançar o quebra-cabeça desliza-varetas. O quebra-cabeça consiste em $N-1$ varetas conectadas a N pinos dispostos no plano em **posição convexa**, formando uma **árvore planar**, isto é, uma estrutura conexa e sem cruzamentos.

O quebra-cabeça vem montado em uma configuração inicial T, e o objetivo é transformá-lo na configuração final T' ilustrada na caixa do jogo.

No entanto, há uma limitação mecânica importante: uma vareta não pode ser movida livremente. O único movimento válido é chamado de **deslize** e funciona da seguinte forma:

- Considere duas varetas ab e bc. É permitido soltar a extremidade comum em b da vareta ab e movê-la ao longo da vareta bc, até conectá-la a c.
- Ao fim do movimento, não é permitido que duas varetas se cruzem.



Exemplo: deslize da aresta 12 pela aresta 32.

Dizemos que duas varetas se **cruzam** se seus segmentos têm um ponto em comum que não seja uma extremidade compartilhada; veja a explicação do segundo exemplo abaixo.

Como você tem um jantar se aproximando e está com pouco tempo, resolveu adicionar uma restrição extra: seu objetivo é encontrar uma sequência de no máximo $2N$ deslizes que transforme T em T'.

Entrada

A primeira linha da entrada contém um inteiro N ($3 \le N \le 10^5$), indicando o número de pinos.

Cada uma das próximas N linhas contém dois inteiros x_i e y_i ($-10^9 \le x_i, y_i \le 10^9$), as coordenadas dos pinos no plano. Os pinos são fornecidos em ordem anti-horária ao longo do fecho convexo.

Depois, seguem $N-1$ linhas, cada uma contendo inteiros a e b ($1 \le a, b \le N, a \ne b$), indicando as varetas da configuração inicial T.

Por fim, as próximas $N-1$ linhas contêm inteiros a' e b' ($1 \le a', b' \le N, a' \ne b'$), indicando as varetas da configuração final T'.

É garantido que os pinos estão em posição convexa, isto é, não há três pinos colineares e, na ordem em que são dados, formam um polígono convexo (logo, nenhum ponto está no interior desse polígono). Além disso, as configurações T e T' são árvores planares, ou seja, não possuem cruzamentos.

Saída

Imprima um inteiro M $(0 \leq M \leq 2N)$, representando o número de deslizes. **Não é preciso minimizar o número de deslizes**; é possível provar que sempre há solução com até $2N$ deslizes.

Nas próximas M linhas, imprima três inteiros a, b e c, indicando um deslize no qual a aresta ab é modificada ao soltar a extremidade em b e deslizá-la ao longo da aresta bc, resultando na aresta ac.

Exemplos

Exemplo de entrada 1

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

Exemplo de saída 1

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

Exemplo de entrada 2

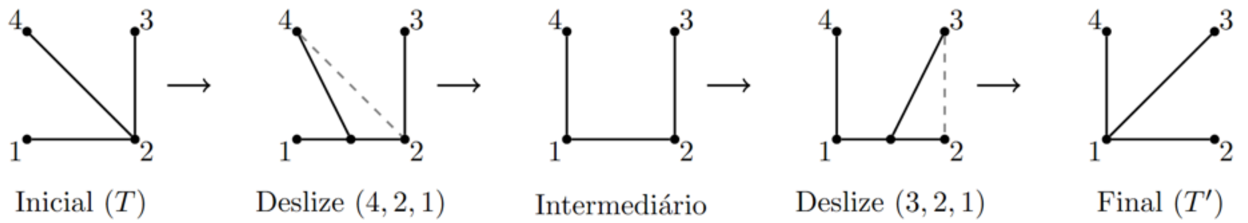
```
4
0 0
1 0
1 1
0 1
1 2
2 3
2 4
1 2
1 3
1 4
```

Exemplo de saída 2

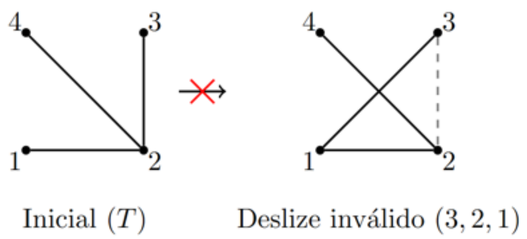
```
2
4 2 1
3 2 1
```

Explicação do exemplo 2

A figura abaixo mostra a sequência de operações do estado inicial ao final do exemplo 2. Note que existe mais de uma solução.



A figura abaixo mostra um deslize inválido. Na configuração T, o deslize da aresta 32 pela aresta 21 é inválido, pois gera um cruzamento entre as varetas 31 e 42.



Problema N. N-amas

Poucos sabem disso, mas o nome 'damas' é na verdade uma corrupção linguística de 'dez-amas', um subconjunto do lendário n-amas.

O tabuleiro de n-amas é um grid $n \times n$ com casas alternando entre branco e preto, sendo branca a casa no canto superior esquerdo. Uma casa pode estar vazia ou, caso seja preta, conter uma peça.

Em sua rodada, o jogador deve escolher uma de suas peças para movimentar. A peça escolhida pode capturar uma peça adversária se esta estiver em uma casa diagonalmente adjacente e a casa seguinte (na mesma direção diagonal) estiver vazia. Neste caso a peça escolhida "pula" a peça adversária, assumindo a posição vazia imediatamente seguinte e removendo a peça adversária do tabuleiro. Após uma captura o jogador pode continuar a jogada com a mesma peça escolhida.

Nas regras oficiais de n-amas existe a lei da maioria: se em seu turno um jogador tiver a oportunidade de capturar peças adversárias, ele **deve obrigatoriamente realizar uma sequência de movimentos que resulte no maior número possível de peças capturadas**.

Durante a final do Torneio Mineiro de n-amas uma disputa surgiu entre os finalistas. Alice, jogando com as peças pretas, acusou Bob, jogando com as brancas, de ter realizado um movimento ilegal. Bob afirma que capturou o máximo que podia, mas Alice diz que havia um caminho maior. Escreva um programa que determine qual o número máximo de peças que Bob pode capturar.

Entrada

A primeira linha contém um único número inteiro n ($4 \leq n \leq 14$), a dimensão do tabuleiro.

Depois seguem n linhas de n caracteres, representando o tabuleiro. Cada linha contém apenas os caracteres '`B`','`P`' e '`.`' representando peças brancas, peças pretas e casas vazias, respectivamente.

O tabuleiro conterá pelo menos uma peça de cada cor e nenhuma peça estará em casas brancas.

Saída

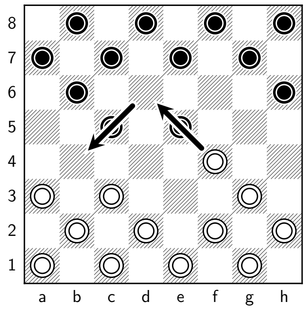
Imprima o número máximo de peças que Bob pode capturar.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1

`8`
`.P.P.P.P`
`P.P.P.P.`
`.P.....P`
`..P.P...`
`.....B..`
`B.B...B.`
`.B.B.B.B`
`B.B.B.B.`
	`2`

Explicação do exemplo 1



O movimento que deve ser feito, onde uma peça de Bob captura duas peças de Alice.

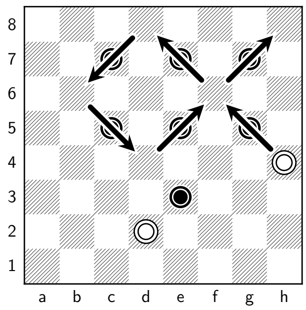
Exemplo de entrada 2

```
8
........
..P.P.P.
........
..P.P.P.
.......B
....P...
...B....
........
```

Exemplo de saída 2

```
6
```

Explicação do exemplo 2



Um dos possíveis movimentos que devem ser feitos, onde uma peça de Bob captura seis peças de Alice.

Exemplo de entrada 3

```
14
.B.P.B.B.P.B.P
B...B...B...B.
.P.B.P.P.B.P.B
..B...B...B...
.B.P.B.B.P.B.P
B...B...B...B.
.P.B.P.P.B.P.B
..B...B...B...
.P.B.P.P.B.P.B
B...B.B.B...B.
.B.P.B.B.P.B.P
..B...B...B...
.P.B.P.P.B.P.B
B...B.B.B...B.
```

Exemplo de saída 3

```
12
```

Problema O. Operações matemáticas

Durante a aula de matemática, os alunos do fundão fizeram a maior algazarra. Irritada com a situação, a professora não deixou barato e decidiu aplicar uma tarefa desafiadora como punição.

A tarefa consiste em uma lista, inicialmente vazia, de operações matemáticas contendo soma e multiplicação. Também são dadas N instruções. Cada instrução tem um dos seguintes formatos:

- $+\ x$: Adicionar no final da lista de operações, uma soma com o valor x
- $*\ y$: Adicionar no final da lista de operações, uma multiplicação por y
- $?\ z$: Partindo do valor z, aplicar a lista de operações **em ordem**, e apresentar o resultado módulo $10^9 + 7$ – o resto da divisão inteira por $10^9 + 7$.

Dada a lista de instruções, ajude os alunos com a tarefa solicitada.

Entrada

A entrada consiste em diversas linhas. A primeira linha da entrada contém o valor de N ($2 \leq N \leq 5 \times 10^5$).

Em cada uma das próximas N linhas da entrada são informadas instruções conforme descrito no enunciado – um caracter (+, * ou ?) seguido por um número $1 \leq x, y, z \leq 10^9$. É garantido que a última instrução é do tipo '?' e que antes da primeira instrução do tipo '?' há pelo menos uma instrução do tipo '+' ou '*'.

Saída

A saída consiste em várias linhas. Cada linha deve conter um número inteiro representando a resposta para cada uma das instruções do tipo "? X". Imprima a resposta módulo $10^9 + 7$.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
4 + 1 * 3 + 2 ? 2	11

Explicação do exemplo 1

A primeira consulta consiste da sequência de operações $2 \xrightarrow{+1} 3 \xrightarrow{\times 3} 9 \xrightarrow{+2} 11$.

Exemplo de entrada 2	Exemplo de saída 2
6 * 1000000000 + 6 ? 1 + 1 ? 1 ? 998244353	1000000006 0 12289585