



Maratona Nordestina de Programação

Caderno de Problemas

2026

Informações Gerais

Este caderno de tarefas tem 25 páginas (sem contar a capa), numeradas de 1 a 25. Verifique se o caderno está completo.

Entrada

- A entrada deve ser lida da entrada padrão.
- Todas as linhas da entrada, incluindo a última, terminam com um caractere de quebra de linha (`\n`).
- Quando a entrada contém vários valores separados por espaço, há exatamente um caractere de espaço em branco entre dois valores consecutivos na mesma linha.

Saída

- A saída deve ser escrita na saída padrão.
- A saída deve seguir o formato especificado no enunciado do problema. A saída não deve conter nenhum dado extra.
- Todas as linhas da saída, incluindo a última, devem terminar com um caractere de quebra de linha (`\n`).
- Quando uma linha da saída exibe vários valores separados por espaço, deve haver exatamente um caractere de espaço em branco entre dois valores consecutivos.
- Quando um valor da saída for um número real, use pelo menos a quantidade de casas decimais correspondente à precisão exigida no enunciado do problema.

Problemas Interativos

A competição pode conter problemas interativos. Nesse tipo de problema, os dados de entrada fornecidos ao seu programa podem não estar predeterminados; em vez disso, eles são construídos especificamente para a sua solução. O juiz executa um programa especial (o interator), cuja saída é transferida para a entrada da sua solução, e a saída do seu programa é enviada para a entrada do interator. Em outras palavras, sua solução e o interator trocam dados e podem decidir o que imprimir com base no histórico de comunicação.

Ao escrever uma solução para um problema interativo, é importante lembrar que, se você imprimir algum dado, ele pode primeiro ser armazenado em um *buffer* interno e não ser transferido imediatamente para o interator. Para evitar essa situação, você deve usar uma operação especial de *flush* sempre que imprimir dados. Essas operações de flush estão disponíveis nas bibliotecas padrão de quase todas as linguagens de programação:

- `fflush(stdout)` em C.
- `cout.flush()` em C++.
- `sys.stdout.flush()` em Python.
- `System.out.flush()` em Java e Kotlin.

Problema A. Arretadas Remoções

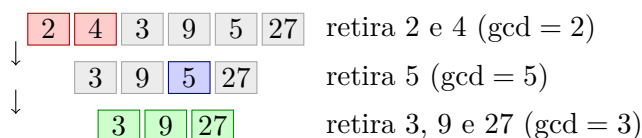
Tempo limite: 1000 ms
Memória limite: 1024 MiB

A Federação de Futebol Tengo-lengo-tengo (FFT) vai organizar a Copa do Mundo Tengo-lengo-tengo, um torneio diferente de tudo que já se viu. Para o show de abertura da copa, os organizadores colocaram no gramado uma fila de N bandeiras. A i -ésima bandeira representa uma equipe que vai disputar a copa e tem um número inteiro A_i escrito.

Na apresentação, as bandeiras serão retiradas do gramado por meio de retiradas arretadas. Uma retirada é considerada arretada quando os responsáveis pela abertura escolhem um trecho não vazio de bandeiras consecutivas, digamos da posição L até a posição R , tal que:

$$\gcd(A_L, A_{L+1}, \dots, A_R) > 1$$

Onde $\gcd$ representa o máximo divisor comum dos números envolvidos. Em uma retirada arretada, eles retiram todas as bandeiras desse trecho. Depois disso, as bandeiras que sobraram se juntam, mantendo a mesma ordem de antes.



Uma sequência possível de retiradas no segundo exemplo. Em cada linha, as bandeiras destacadas são retiradas.

Para planejar bem o show de abertura, a FFT quer saber o menor número de retiradas arretadas necessárias para remover todas as bandeiras durante a apresentação.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 200$), a quantidade de bandeiras.

A segunda linha contém N inteiros $A_1, A_2, \dots, A_N$ ($2 \leq A_i \leq 10^6$), os números escritos nas bandeiras.

Saída

Imprima um único inteiro: o menor número de retiradas arretadas necessário para remover todas as bandeiras.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
6 2 3 5 5 3 2	3
Exemplo de entrada 2	Exemplo de saída 2
6 2 4 3 9 5 27	3

Problema B. Bons Refletores

Tempo limite: 1000 ms
Memória limite: 1024 MiB

Na final da Copa do Mundo de Futebol Tongo-lengo-tengo, o show de luzes do estádio é um espetáculo no show do intervalo, enquanto vários artistas fazem um show musical. Cada configuração dos refletores é representada por um número binário de exatamente 3 bits: o bit 0 indica uma luz apagada, e o bit 1 indica uma luz acesa.

Para o show do intervalo ficar bonito, a organização quer alternar as luzes com calma. Por isso, de uma configuração para a próxima, exatamente um refletor deve mudar de estado.

Você recebe uma configuração inicial X_0 . Sua tarefa é produzir uma sequência que comece em X_0 , passe por todas as 8 configurações de 3 bits exatamente uma vez, e então volte para X_0 .

Entrada

A entrada contém uma única linha com uma string X_0 de exatamente 3 caracteres. Cada caractere de X_0 é 0 ou 1.

Saída

Imprima 9 strings binárias de exatamente 3 bits.

A primeira e a última string devem ser iguais a X_0 . Entre as primeiras 8 strings, cada uma das 8 configurações possíveis deve aparecer exatamente uma vez. Além disso, quaisquer duas strings consecutivas na sua saída devem diferir em exatamente um bit.

Se houver mais de uma resposta válida, imprima qualquer uma delas.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
010	010 110 111 101 100 000 001 011 010

Problema C. Categorias de Base

Tempo limite: 2000 ms
Memória limite: 1024 MiB

Um clube de futebol está reorganizando suas categorias de base para participar de um campeonato organizado pela FFT, a Federação de Futebol Tengo-lengo-tengo.

Existem N profissionais disponíveis, e cada profissional possui uma função principal dentro de uma comissão técnica.

As funções são representadas por inteiros. Por exemplo, a função 1 pode representar um treinador, a função 2 um preparador físico, a função 3 um fisioterapeuta, a função 4 um analista de desempenho, e a função 5 um treinador de goleiros.

O i -ésimo profissional possui função A_i .

O clube deseja dividir todos os N profissionais em exatamente K comissões técnicas, uma para cada categoria da base.

Porém, para evitar responsabilidades repetidas dentro de uma mesma comissão, uma comissão técnica não pode conter dois profissionais com a mesma função. Além disso, cada comissão técnica deve conter pelo menos um profissional.

Duas divisões são consideradas diferentes se existe algum grupo de profissionais que forma uma comissão técnica em uma divisão, mas não forma uma comissão técnica na outra. A ordem das comissões técnicas não importa, e a ordem dos profissionais dentro de uma mesma comissão também não importa.

Determine a quantidade de formas diferentes de formar as K comissões técnicas, módulo 998244353.

Entrada

A primeira linha contém dois inteiros N e K ($1 \leq N, K \leq 10^5$).

A segunda linha contém N inteiros $A_1, A_2, \dots, A_N$ ($1 \leq A_i \leq N$), onde A_i representa a função principal do i -ésimo profissional.

Saída

A saída deve conter apenas um inteiro, a quantidade de formas diferentes de dividir os profissionais em exatamente K comissões técnicas, módulo 998244353.

Exemplos

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

Explicação do exemplo 1

Existem 5 profissionais e devem ser formadas 2 comissões técnicas. As funções são:

- Função 1: 2 profissionais;
- Função 2: 2 profissionais;

- Função 3: 1 profissional.

Uma divisão válida é colocar um profissional com função 1 e um profissional com função 2 em uma comissão técnica, e os profissionais restantes na outra comissão técnica.

Nenhuma comissão técnica contém dois profissionais com a mesma função, e todos os profissionais são usados exatamente uma vez.

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

Explicação do exemplo 2

Existem 3 profissionais, todos com funções diferentes entre si, e devem ser formadas 3 comissões técnicas. A única divisão válida é colocar um profissional em cada comissão.

Problema D. Decodificador Maluco

Tempo limite: 1000 ms
Memória limite: 1024 MiB

O ano é 3026 e a inteligência artificial GaLaDOS está prestes a dominar o mundo. Felizmente, a humanidade tem uma última chance, pois foi descoberto que a IA compartilhou uma senha de administrador com os seus aliados. Ao obter esta senha, a IA poderá ser desativada e a humanidade será salva.

No entanto, é claro que as coisas não serão tão fáceis assim. Primeiro, foram definidos pela GaLaDOS dois números, N e K . Em seguida, a senha X que inicialmente era um número entre 0 e $2^{32} - 1$ passou pelo seguinte processo de criptografia N vezes:

1. Seja Y o valor de $X \oplus (X \cdot 2^K)$
2. Altere X para ser o valor dos 32 primeiros bits de Y

Onde $\oplus$ representa a operação de XOR* entre dois números.

Seu objetivo é: Dados N , K e o resultado final da criptografia S , construa um decodificador que retorne a senha original X . Caso exista mais de uma senha válida, você pode retornar qualquer uma delas.

* A operação XOR entre dois números é o resultado do OU exclusivo entre os bits das representações binárias dos dois operandos. Exemplo: $5 \oplus 3 = (101)_2 \oplus (011)_2 = (110)_2 = 6$

Entrada

O input é uma única linha contendo 3 inteiros N ($1 \leq N \leq 10^6$), K ($1 \leq K \leq 30$) e S ($1 \leq S < 2^{32}$)

Saída

O output deve ser um único inteiro, a senha original X , impressa como um **inteiro de 32 bits sem sinal**

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2 2 51	3

Explicação do exemplo 1

Neste exemplo, $N = 2$. Se começarmos com $X = 3$, no primeiro passo teremos $Y = 3 \oplus (3 \cdot 2^2) = 3 \oplus 12 = 15$ e o valor de X é atualizado.

No segundo passo, temos $Y = 15 \oplus (15 \cdot 2^2) = 15 \oplus 60 = 51$. O valor de X é atualizado e o processo acaba com $X = 51$, que é o valor de S .

Exemplo de entrada 2	Exemplo de saída 2
2 3 6	2249744774

Explicação do exemplo 2

Lembre-se que a senha é um número entre 0 e $2^{32} - 1$

Exemplo de entrada 3

1 1 4294967295

Exemplo de saída 3

1431655765

Explicação do exemplo 3

Lembre-se que S é um número entre 1 e $2^{32} - 1$.

Problema E. Enigma da Balança

Tempo limite: 1000 ms
Memória limite: 1024 MiB

Este é um problema interativo. Você pode ler mais sobre esse tipo de problema na página 1 do caderno completo de prova.

Na Feira de Caruaru, famosa pelas suas bancas de roupas, um feirante separou N camisas aparentemente idênticas para um desafio. Dessas camisas, $N - 1$ têm exatamente o mesmo peso, enquanto uma camisa diferente é mais pesada que todas as outras.

Você pode usar uma balança de dois pratos. Em cada pesagem, você escolhe uma mesma quantidade de camisas para colocar no prato esquerdo e no prato direito. A balança informa se o prato esquerdo ficou mais pesado, se o prato direito ficou mais pesado, ou se os dois pratos ficaram com o mesmo peso.

Seu objetivo é descobrir qual é a camisa mais pesada realizando, no máximo, 8 pesagens.

Interação

Inicialmente, o interator imprime uma linha contendo um inteiro N ($2 \leq N \leq 2000$), a quantidade de camisas. As camisas são numeradas de 1 a N .

O interator é **não adaptativo**: a camisa mais pesada é escolhida antes do início da interação e não muda durante a execução.

Seu programa pode realizar pesagens. Para realizar uma pesagem, imprima uma linha no formato

? t a_1 a_2 ... a_t b_1 b_2 ... b_t

em que $t \geq 1$. As camisas $a_1, a_2, \dots, a_t$ serão colocadas no prato esquerdo, e as camisas $b_1, b_2, \dots, b_t$ serão colocadas no prato direito. Todos esses $2t$ índices devem ser distintos e estar entre 1 e N .

O interator responderá com uma linha contendo um único caractere:

- E, se o prato esquerdo ficou mais pesado;
- D, se o prato direito ficou mais pesado;
- I, se os dois pratos ficaram com o mesmo peso.

Quando seu programa souber qual é a camisa mais pesada, ele deve imprimir uma linha no formato

! p

em que p é o índice da camisa mais pesada.

A operação de resposta não conta como uma pesagem. Apenas as operações iniciadas por ? são contabilizadas, e seu programa pode realizar no máximo 8 delas.

Após imprimir qualquer linha, seu programa deve realizar *flush* da saída. Por exemplo, em C++ você pode usar `cout << endl;` ou `cout.flush()`.

Se seu programa imprimir um valor inválido, realizar mais de 8 pesagens, ou responder uma camisa incorreta, ele receberá o veredito de *NO - Wrong answer*.

Exemplos

Leitura	Exemplo de interação 1	Escrita
4		
	? 2 1 2 3 4	
D		
	? 1 3 4	
E		
	! 3	

Explicação do exemplo 1

Nesse exemplo, a camisa mais pesada é a de número 3. O programa usa duas das 8 pesagens permitidas para determinar qual é a camisa mais pesada.

Na primeira pesagem, as camisas 1 e 2 são colocadas no prato esquerdo, enquanto as camisas 3 e 4 são colocadas no prato direito. Como o prato direito fica mais pesado, a camisa diferente deve ser a 3 ou a 4.

Na segunda pesagem, o programa compara as camisas 3 e 4. Como o prato da esquerda fica mais pesado, o programa conclui que a camisa 3 é a mais pesada.

Se, na segunda pesagem, o prato da direita fosse o mais pesado, a camisa 4 seria a mais pesada.

Não é necessário realizar todas as 8 pesagens permitidas. O programa pode responder assim que determinar o índice da camisa mais pesada.

Problema F. Fugindo do Sol

Tempo limite: 1000 ms
Memória limite: 1024 MiB

Durante suas férias em Aracaju, Samuel resolveu passar a manhã tomando um banho de mar na Praia de Atalaia, um dos cartões-postais da cidade, famosa por sua belíssima e extensa faixa de areia.

Na hora de ir embora, veio o problema: com o sol a pino, ele precisa atravessar toda a praia, do mar até o calçadão, e pedir um Uber. Para não sair queimado, Samuel planeja usar as sombras dos guarda-sóis espalhados pela areia e traçar uma rota que minimize a distância percorrida sob o sol.

Formalmente, a praia pode ser representada pela faixa entre duas retas paralelas: uma corresponde à linha onde o mar encontra a areia ($y = 0$), e a outra, ao calçadão ($y = H$), a uma distância H uma da outra. Existem N guarda-sóis, o i -ésimo projeta uma sombra circular de centro (x_i, y_i) e raio r_i .

Considere que Samuel pode nadar livremente debaixo da água, sem ficar exposto ao sol, e sair da água em qualquer ponto da linha do mar. A partir da areia, ele pode:

- caminhar sob a sombra de um guarda-sol, inclusive passando diretamente para outro quando suas sombras se tocam ou se sobrepõem;
- sair da areia por qualquer ponto do calçadão.

Ajude Samuel a determinar a menor distância que ele precisa percorrer sob o sol para alcançar o calçadão.

Entrada

A primeira linha contém dois inteiros N e H ($0 \leq N \leq 1000$, $1 \leq H \leq 10^5$), o número de guarda-sóis e a distância entre a linha do mar e o calçadão, respectivamente.

Cada uma das N linhas seguintes contém três inteiros x_i , y_i e r_i ($0 \leq x_i \leq 10^5$, $0 \leq y_i \leq H$, $1 \leq r_i \leq 10^5$), o centro e o raio do i -ésimo guarda-sol. A sombra de um guarda-sol pode ultrapassar a linha do mar ou o calçadão.

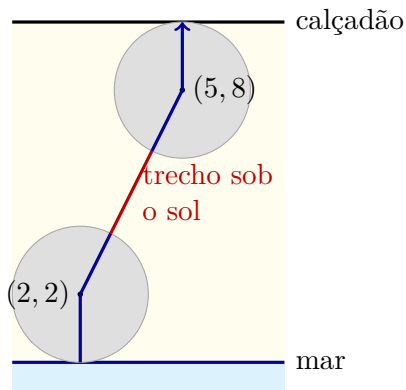
Saída

Imprima um único número real: a menor distância sob o sol que Samuel precisa percorrer para atravessar a areia, do mar até o calçadão.

Sua resposta será considerada correta se o erro absoluto ou relativo em relação à resposta esperada for no máximo 10^{-6} .

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2 10 2 2 2 5 8 2	2.7082039325

Explicação do exemplo 1

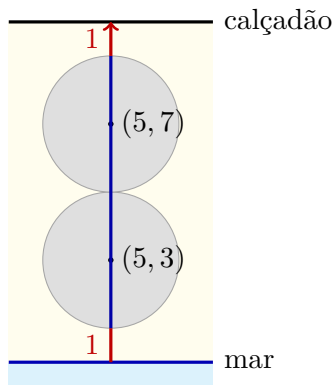
No primeiro exemplo, Samuell pode fazer um trajeto no qual percorre $3\sqrt{5} - 4$ sob o sol.

Exemplo de entrada 2

```
2 10
5 3 2
5 7 2
```

Exemplo de saída 2

```
2
```

Explicação do exemplo 2

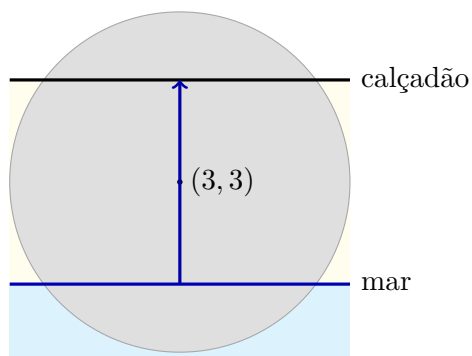
No segundo exemplo, Samuell percorre uma unidade sob o sol para alcançar a primeira sombra. As duas sombras se tocam, permitindo que ele passe de uma para a outra sem se expor. Depois, ele percorre mais uma unidade até o calçadão. Portanto, a distância total sob o sol é $1 + 1 = 2$.

Exemplo de entrada 3

```
1 6
3 3 5
```

Exemplo de saída 3

```
0
```

Explicação do exemplo 3

No terceiro exemplo, a única sombra tem raio 5 e centro em $(3, 3)$. Como ela alcança tanto a linha do mar quanto o calçadão, Samuell pode atravessar toda a praia permanecendo na sombra. Portanto, a distância percorrida sob o sol é 0.

Problema G. Gaiatice na Fila

Tempo limite: 3000 ms
Memória limite: 1024 MiB

A grande final da Copa do Mundo Tongo-lengo-tengo está prestes a começar! O jogo decisivo será disputado entre o **Altos** e o **Bahia**, e a fila para entrar no estádio está uma verdadeira loucura.

A fila é composta inteiramente por torcedores das duas equipes. Como manda a boa tradição, sempre que um torcedor do Altos está posicionado antes (mais perto da entrada do estádio) de um torcedor do Bahia, dizemos que existe uma *resenha* acontecendo entre eles.

Matematicamente, se considerarmos a fila como uma string S composta pelas letras 'A' e 'B', uma *resenha* é formada por cada par de índices (i, j) tal que $i < j$, onde $S[i]$ é um torcedor do Altos ('A') e $S[j]$ é um torcedor do Bahia ('B').

A organização do estádio registrou Q eventos na fila, que podem ser de 4 tipos:

- **Tipo 1:** Um torcedor do time X ('A' ou 'B') entra no final da fila.
- **Tipo 2:** Um torcedor gaiato do time X ('A' ou 'B') fura a fila e entra no exato início dela.
- **Tipo 3:** O torcedor que estava no final da fila se agunea e vai embora.
- **Tipo 4:** O torcedor que estava no início da fila passa pela catraca e entra no estádio.

Sua tarefa é calcular o número total de *resenhas* acontecendo na fila imediatamente após cada um dos eventos.

Entrada

A primeira linha da entrada contém um inteiro Q ($1 \leq Q \leq 10^6$), representando o número de eventos.

As próximas Q linhas descrevem os eventos. Cada linha começa com um inteiro T ($1 \leq T \leq 4$):

- Se $T = 1$, ele é seguido por um caractere X ('A' ou 'B'), indicando o time do torcedor que entrou no final da fila.
- Se $T = 2$, ele é seguido por um caractere X ('A' ou 'B'), indicando o time do torcedor que entrou no início da fila.
- Se $T = 3$, o torcedor no final da fila sai (nenhum caractere é fornecido).
- Se $T = 4$, o torcedor no início da fila entra no estádio (nenhum caractere é fornecido).

É garantido que as operações de tipo 3 e 4 nunca serão chamadas quando a fila estiver vazia.

Saída

Para cada query, imprima um único inteiro em uma linha: a quantidade de resenhas acontecendo na fila no momento atual.

Exemplos

Exemplo de entrada 1

1	0
2 A	

Exemplo de saída 1

Exemplo de entrada 2

5
1 A
1 B
1 B
1 A
1 B

Exemplo de saída 2

0
1
2
2
4

Explicação do exemplo 2

No segundo exemplo, a fila passa a ser, após cada evento, respectivamente, A, AB, ABB, ABBA e ABBAB. As quantidades de resenhas nessas filas são 0, 1, 2, 2 e 4, pois cada resenha corresponde a um A que aparece antes de um B.

Exemplo de entrada 3

5
1 A
2 B
2 A
3
4

Exemplo de saída 3

0
0
1
1
0

Explicação do exemplo 3

No terceiro exemplo, a fila passa a ser, após cada evento, respectivamente, A, BA, ABA, AB e B. Assim, as quantidades de resenhas são 0, 0, 1, 1 e 0.

Problema H. Histórias de Cordel

Tempo limite: 1000 ms
Memória limite: 1024 MiB

Desde menino, o cordelista Severino guarda causos ouvidos nas feiras do sertão: pelejas de cantadores, bichos encantados e valentias que atravessam gerações. Agora ele prepara seu maior folheto, uma aventura em que, a cada capítulo, o leitor escolhe por onde a história continua. Para não perder nenhum desfecho, Severino espalhou pelo chão as páginas de seu roteiro e ligou cada escolha com um pedaço de barbante.

Visto de cima, o roteiro assemelha-se a uma estrutura de árvore: são N capítulos numerados de 1 a N , sendo o capítulo 1 o início de todas as histórias. O roteiro atende as seguintes regras:

- Cada ligação une um capítulo a uma de suas continuações imediatas;
- Para cada ligação, Severino deve escolher uma letra maiúscula entre A e Z;
- Nenhum capítulo pode ter duas continuações com a mesma letra;
- Cada capítulo sem continuações encerra uma história completa, descrita pela sequência de letras no caminho do capítulo 1 até ele.

Severino percebeu ainda que trocar letras no roteiro nem sempre muda o folheto: duas atribuições de letras são consideradas equivalentes quando produzem exatamente o mesmo conjunto de histórias completas.

Severino gostaria de saber quantos conjuntos distintos de histórias completas podem ser obtidos a partir desse roteiro dentre todas as atribuições possíveis de letras. Como a resposta pode ser muito grande, ele pede que você informe o resultado módulo 998244353.

Entrada

A primeira linha contém um inteiro N ($2 \leq N \leq 10^5$), o número de capítulos.

Cada uma das próximas $N - 1$ linhas contém dois inteiros U e V ($1 \leq U, V \leq N$), indicando que existe uma ligação entre os capítulos U e V .

É garantido que essas ligações formam uma árvore, enraizada no capítulo 1.

Saída

Imprima um único inteiro: a quantidade de conjuntos distintos de histórias completas que podem ser obtidos, módulo 998244353.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
3 1 2 1 3	325
Exemplo de entrada 2	Exemplo de saída 2
2 2 1	26

Problema I. Incrível XORlândia

Tempo limite: 2000 ms
Memória limite: 1024 MiB

XORlândia é uma cidade localizada no litoral do Nordeste brasileiro, famosa por suas praias de águas cristalinas, por sua enorme orla e pelo seu slogan *XORlândia é massa*. Diz a lenda que, durante a construção da cidade, um antigo pescador conhecido como Velho Sururu amaldiçoou algumas ruas da região após perder uma partida de dominó para um morador local.

Desde então, cada rua da cidade passou a possuir uma energia, representada por um número inteiro associado a ela. A cidade possui N locais turísticos numerados de 1 até N , no qual o local 1 é o Marco dos XORais e o local N é onde fica a Cadeira XORgante, conectados por M ruas bidirecionais. A i -ésima rua conecta os locais U_i e V_i , possuindo energia W_i . Além disso, sabe-se que é possível chegar de qualquer local a qualquer outro e que existe, no máximo, uma rua ligando cada par de locais.

Como uma tradição local, os moradores definem o valor de um caminho como o XOR das energias de todas as ruas percorridas. Um caminho pode passar pelo mesmo local e percorrer a mesma rua múltiplas vezes. Cada vez que uma rua é percorrida, sua energia é incluída novamente no cálculo do XOR, mesmo que essa mesma rua já tenha sido utilizada anteriormente no caminho. A prefeitura considera XORlândia uma cidade incrível se todo caminho que parte do Marco dos XORais e termina na Cadeira XORgante tiver valor igual a 0.

Assim, a prefeitura decidiu que pode escolher **exatamente uma rua** e alterar sua energia para qualquer outro valor inteiro não negativo, desde que seja diferente do valor anterior.

Para cada rua, determine se é possível alterar apenas o valor daquela rua, de forma que a cidade passe a ser considerada incrível.

Entrada

A primeira linha contém dois inteiros N e M , indicando respectivamente o número de locais turísticos em XORlândia e a quantidade de ruas.

Cada uma das próximas M linhas contém três inteiros U_i , V_i e W_i ($1 \leq U_i, V_i \leq N$, $U_i \neq V_i$, $0 \leq W_i < 2^{30}$), representando uma rua bidirecional que liga os locais U_i e V_i , possuindo energia W_i .

- $2 \leq N \leq 200000$
- $N - 1 \leq M \leq \min\left(\frac{N(N-1)}{2}, 200000\right)$

Saída

Imprima uma string binária de tamanho M .

O i -ésimo caractere deve ser:

- 1, se for possível modificar apenas a i -ésima rua para que XORlândia se torne incrível;
- 0, caso contrário.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
5 5 1 2 1 1 3 0 3 4 0 2 4 2 4 5 3	01100

Explicação do exemplo 1

No primeiro exemplo, é possível alterar a energia da segunda ou da terceira rua para que todos os caminhos do local 1 ao local 5 tenham valor 0. Nenhuma das outras ruas permite obter essa condição com uma única alteração. Por isso, a resposta é 01100.

Exemplo de entrada 2	Exemplo de saída 2
3 2 1 2 127 2 3 127	00

Explicação do exemplo 2

No segundo exemplo, o caminho do local 1 ao local 3 tem valor $127 \oplus 127 = 0$. Porém, como é obrigatório alterar a energia de exatamente uma rua para um valor diferente, não é possível manter esse valor igual a 0. Portanto, a resposta é 00.

Exemplo de entrada 3	Exemplo de saída 3
3 2 1 2 127 2 3 128	11

Explicação do exemplo 3

No terceiro exemplo, pode-se alterar a energia da primeira rua para 128 ou a energia da segunda rua para 127. Em qualquer uma das escolhas, o caminho do local 1 ao local 3 passa a ter valor 0. Portanto, a resposta é 11.

Problema J. Jogo do Brega

Tempo limite: 1000 ms
Memória limite: 1024 MiB



Reginaldo e Adilson irão se apresentar em vários dias durante um festival de brega no interior de Pernambuco e para decidir quais músicas cada um cantará, eles criaram um jogo, carinhosamente apelidado de *Jogo do Brega*.

Nesse jogo, há uma sequência com N cartas dispostas no centro de uma mesa. Cada carta possui um número distinto de 1 a N , ou seja, a sequência forma uma permutação desses valores. Cada carta representa uma música do repertório, e o número escrito nela corresponde à pontuação dessa música, indicando o quanto o público gosta dela.

O festival contará com vários dias de apresentações e, em cada dia, eles escolhem um intervalo da sequência, indicando quais músicas estarão disponíveis para serem cantadas naquele dia. O jogo é então realizado considerando apenas esse intervalo contíguo de cartas. Os cantores jogam alternadamente, sendo que Adilson sempre faz a primeira jogada. Em cada turno, o jogador da vez deve remover exatamente uma carta da sequência restante: ou a carta mais à esquerda ou a carta mais à direita. A carta removida representa uma música que será cantada pelo respectivo cantor durante o show.

A pontuação de cada cantor é o maior número escrito entre todas as cartas que ele removeu durante a partida, indicando a música que aquele cantor cantou que mais animou o público. Caso um cantor não remova nenhuma carta, sua pontuação é 0. Quando todas as cartas forem removidas, vence o cantor com maior pontuação. Como todos os números da sequência são distintos, sempre existe exatamente um vencedor.

Assim, para cada intervalo informado, determine qual cantor vence supondo que ambos jogam da melhor forma possível.

Entrada

A primeira linha contém dois inteiros N e Q ($1 \leq N \leq 2 \times 10^5$, $1 \leq Q \leq 2 \times 10^5$), respectivamente a quantidade de músicas no repertório e a quantidade de dias do festival.

A segunda linha contém N inteiros $P_1, P_2, \dots, P_N$, uma permutação dos inteiros de 1 a N .

Para cada $1 \leq i \leq Q$, a i -ésima das próximas Q linhas contém dois inteiros L_i e R_i ($1 \leq L_i \leq R_i \leq N$), indicando que nesse dia de festival apenas as músicas $P_{L_i}, P_{L_i+1}, \dots, P_{R_i}$ serão cantadas.

Saída

Para cada dia de festival, imprima uma linha contendo o nome do cantor vencedor: **Reginaldo**, se Reginaldo vence, ou **Adilson**, se Adilson vence.

Exemplos

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

Explicação do exemplo 1

No primeiro dia, são usadas todas as cartas, e Reginaldo vence quando ambos jogam da melhor forma possível. No segundo dia, há apenas a carta 2, que Adilson remove na primeira jogada. No terceiro dia, as cartas são 5, 1 e 4, e Adilson pode remover a carta 5 imediatamente.

Problema K. Karamelos no São João

Tempo limite: 1000 ms
Memória limite: 1024 MiB

João foi para o São João de Caruaru com uma missão simples, pelo menos na cabeça dele: comprar doces para os amigos e ainda garantir um agradinho para comer depois do forró.

João tem A amigos e encontrou uma barraquinha vendendo exatamente X tipos diferentes de doces. Ele quer escolher um inteiro positivo Y e comprar exatamente Y doces de cada tipo. Assim, no total, João compraria $X \cdot Y$ doces.

Depois da compra, João vai distribuir os doces igualmente entre os A amigos, possivelmente sobrando alguns doces. Cada amigo precisa receber pelo menos um doce, porque aparecer na festa e sair de mãos abanando é motivo para resenha até o próximo São João.

Para fechar a conta do jeito que ele imaginou, exatamente um doce deve sobrar para o próprio João, assim, ele quer saber se existe algum valor de Y que permita isso. Se existir, determine o menor possível.

Entrada

A entrada contém dois inteiros A e X ($2 \leq A \leq 10^4$ e $1 \leq X < A$), a quantidade de amigos e de tipos de doces.

Saída

Imprima o menor valor possível de Y , ou -1 se ele não existir.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
5 2	3

Explicação do exemplo 1

Com $Y = 3$, João compra 6 doces: distribui 1 doce para cada um dos 5 amigos e fica com o doce restante.

Exemplo de entrada 2	Exemplo de saída 2
6 4	-1

Explicação do exemplo 2

Os números 6 e 4 têm 2 como maior divisor em comum. Assim, toda compra total é múltipla de 4 e, dividindo por 6, a sobra nunca é 1. Logo, não há valor de Y que seja válido.

Exemplo de entrada 3	Exemplo de saída 3
7 1	8

Explicação do exemplo 3

Com $Y = 1$ sobraria 1 doce, mas nenhum amigo receberia doce. O menor valor válido é $Y = 8$.

Problema L. Lampions League

Tempo limite: 1000 ms
Memória limite: 1024 MiB

A cada quatro anos, no Nordeste, a FFT (Federação de Futebol Tengo-lengo-tengo) promove a Copa do Mundo Tengo-lengo-tengo. Sabendo da grande importância dessa competição, a torcida logo batizou carinhosamente o torneio de Lampions League, fazendo referência a aquela competição europeia de nível mais fraco, afinal, todos sabemos que na Copa do Mundo Tengo-lengo-tengo o nível de competição é elevado.

Durante a partida final, o narrador oficial irá narrar os lances da partida de uma forma especial. No primeiro lance, ele grita **Tengo** 1 vez; no segundo, grita **Lengo** 2 vezes; no terceiro, grita **Tengo** 3 vezes; no quarto, grita **Lengo** 4 vezes; e assim por diante, alternando a palavra até o lance N .

A FFT precisa saber se a diferença entre a quantidade de vezes que **Tengo** foi gritado e a quantidade de vezes que **Lengo** foi gritado, em valor absoluto, é par ou ímpar.

Dado o valor de N , determine se essa diferença é par ou ímpar.

Entrada

A entrada contém uma única linha com um inteiro N ($1 \leq N \leq 10^4$), representando a quantidade de lances narrados.

Saída

A saída deve conter uma única linha com a palavra **PAR**, se a diferença em valor absoluto entre a quantidade de **Tengos** e **Lengos** gritados for par, ou **IMPAR**, caso essa diferença seja ímpar.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2	IMPAR
Exemplo de entrada 2	Exemplo de saída 2
10000	PAR

Problema M. Maior São João do Mundo

Tempo limite: 1000 ms
Memória limite: 1024 MiB

Todo ano, uma discussão movimentada o Nordeste: afinal, qual é o maior São João do mundo, o de Campina Grande ou o de Caruaru?

Para tentar resolver essa questão, um instituto de pesquisa decidiu realizar duas votações: uma durante a festa de São João de Campina Grande e outra durante a festa de São João de Caruaru.

As N pessoas entrevistadas participaram das duas festas, conhecendo de perto tanto o São João de Campina Grande quanto o de Caruaru. Porém, por questões de organização da pesquisa, cada pessoa participou de exatamente uma das duas votações.

É permitido que uma das votações fique vazia.

Cada pessoa tem uma opinião verdadeira sobre qual festa prefere: Campina Grande ou Caruaru.

No entanto, as votações aconteceram no meio da festa, com muito forró, fogos e barulho. Por isso, quando uma pessoa dizia sua resposta, o instituto podia acabar entendendo errado. Assim, o voto computado para uma pessoa pode ser diferente da opinião verdadeira dela.

Alguns dias depois, cada participante recebe o resultado da votação em que participou: quantos votos foram computados para Campina Grande e quantos votos foram computados para Caruaru.

A avaliação de cada pessoa sobre a votação depende apenas da sua opinião verdadeira:

- para cada voto computado para a festa que ela prefere, sua avaliação aumenta em 1;
- para cada voto computado para a outra festa, sua avaliação diminui em 1.

A pontuação total de uma votação é a soma das avaliações de todos os seus participantes.

Supondo que cada participante tenha participado de uma votação de forma a maximizar a soma das pontuações, determine a maior soma possível das pontuações das duas votações.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 2 \times 10^5$), a quantidade de pessoas entrevistadas.

A segunda linha contém uma string P de tamanho N , onde P_i representa a opinião verdadeira da pessoa i :

- G indica que ela prefere o São João de Campina Grande;
- R indica que ela prefere o São João de Caruaru.

A terceira linha contém uma string V de tamanho N , onde V_i representa o voto que foi computado para a pessoa i :

- G indica que o voto foi computado para Campina Grande;
- R indica que o voto foi computado para Caruaru.

Saída

A saída deve conter apenas um inteiro, a maior soma possível das pontuações das duas votações.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
4 GRRR GRGR	2

Explicação do exemplo 1

Uma divisão ótima é colocar a pessoa 1 na primeira votação e as pessoas 2, 3 e 4 na segunda votação.

Na primeira votação, existe apenas a pessoa 1. O voto computado dela foi G, e ela prefere G, então sua avaliação é 1. Portanto, a pontuação da primeira votação é 1.

Na segunda votação, os votos computados das pessoas 2, 3 e 4 são, respectivamente, R, G e R. Assim, o resultado dessa votação contém 1 voto para Campina Grande e 2 votos para Caruaru.

A pessoa 2 prefere G, então sua avaliação é $1 - 2 = -1$.

A pessoa 3 prefere R, então sua avaliação é $2 - 1 = 1$.

A pessoa 4 prefere R, então sua avaliação é $2 - 1 = 1$.

Logo, a pontuação da segunda votação é $-1 + 1 + 1 = 1$. A soma das pontuações das duas votações é $1 + 1 = 2$, que é o maior valor possível.

Exemplo de entrada 2	Exemplo de saída 2
3 GGG GGG	9

Explicação do exemplo 2

Todas as pessoas preferem Campina Grande, e todos os votos foram computados para Campina Grande.

Uma divisão ótima é colocar todas as pessoas em uma mesma votação e deixar a outra votação vazia.

Na votação com as três pessoas, existem 3 votos computados para Campina Grande e 0 votos computados para Caruaru. Como todas as pessoas preferem Campina Grande, cada uma delas tem avaliação $3 - 0 = 3$.

Portanto, a pontuação dessa votação é $3 + 3 + 3 = 9$. A outra votação está vazia e contribui com 0 pontos.

Assim, a maior soma possível das pontuações é 9.

Problema N. Não visse meu álbum?

Tempo limite: 1000 ms
Memória limite: 1024 MiB

Xand Avião está muito animado para começar a completar o novo álbum de figurinhas da Federação de Futebol Tengo-lengo-tengo (FFT). Isso porque o novo álbum da FFT é como um bingo, no dia que você completar, você ganha um prêmio!

O álbum inicialmente conta com N ($N \leq 2026$) jogadores licenciados pela FFT. A cada dia, pela manhã, existe uma chance de $P\%$ de um novo jogador entrar para o álbum. Isso só é possível se há menos de 2026 jogadores no álbum. Afinal, esta é a quantidade de jogadores de futebol no mundo!

Xand inicialmente não tem nenhuma figurinha. Então todo dia, pela tarde, Xand irá comprar uma nova. A figurinha tem chance equiprovável de ser qualquer jogador que está no álbum. Neste momento, se Xand possuir todas as figurinhas dos jogadores que já estão no álbum, ele imediatamente troca o álbum pelo seu prêmio.

Xand está tão animado pelo prêmio que mal consegue contar os dias para completar o álbum. Logo, ele pediu sua ajuda para contar por ele: Qual a quantidade esperada de dias que Xand vai precisar para ganhar o prêmio do álbum de figurinhas?

Entrada

O input é uma única linha composta por dois inteiros, N ($1 \leq N \leq 2026$) e P ($0 \leq P \leq 100$)

Saída

Sua saída deve conter um único número indicando a quantidade esperada de dias que Xand vai precisar para completar o álbum de figurinhas. Sua resposta será considerada correta se possuir um erro relativo ou absoluto de no máximo 10^{-6} em relação a resposta do juiz.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
1 0	1.000000000000

Explicação do exemplo 1

No primeiro dia, é garantido que nenhum jogador novo será adicionado e Xand irá obter a única figurinha necessária.

Exemplo de entrada 2	Exemplo de saída 2
2 0	3.000000000000

Explicação do exemplo 2

É garantido que nenhum jogador novo será adicionado. Sendo assim, Xand tem probabilidade $\frac{1}{2}$ de completar o álbum em 2 dias, $\frac{1}{4}$ de completar em 3 dias e assim por diante. Desta forma, o valor esperado é $\sum_{i=1}^{\infty} (i+1) \left(\frac{1}{2}\right)^i = 3$

Exemplo de entrada 3	Exemplo de saída 3
5 10	203.072212695099

Problema O. Operação Mandacaru

Tempo limite: 5000 ms
Memória limite: 1024 MiB

A Viação Mandacaru percorre diariamente uma rota que atravessa M cidades do Nordeste, numeradas de 1 a M na ordem em que são visitadas. O ônibus usado nessa rota possui exatamente C assentos.

Certo dia, uma falha no sistema interrompeu simultaneamente N pedidos de passagens. Embora nenhuma passagem tenha sido emitida, os dados dos pedidos foram recuperados. O pedido do passageiro i é descrito por três inteiros L_i , R_i e P_i : o passageiro deseja embarcar na cidade L_i , desembarcar na cidade R_i e pagaria P_i reais pela passagem.

A companhia não permite que os passageiros escolham seus assentos. Portanto, um pedido só pode ser aceito se o ônibus tiver um assento disponível durante todo o percurso entre as cidades L_i e R_i . Em cada cidade, todos os passageiros que terminam sua viagem nela desembarcam antes que os novos passageiros embarquem. Assim, um passageiro que embarca na cidade j pode ocupar um assento liberado nessa mesma cidade.

Como todos os pedidos foram recebidos ao mesmo tempo, talvez não seja possível aceitá-los por completo sem exceder a capacidade do ônibus em algum trecho da rota. Sempre preocupada em resolver a situação da maneira mais justa possível — especialmente para o próprio caixa —, a Viação Mandacaru decidiu escolher quais passagens serão emitidas de modo a maximizar sua arrecadação.

Determine o maior valor total que a empresa pode arrecadar emitindo um subconjunto dos pedidos, sem que o ônibus transporte mais de C passageiros simultaneamente em nenhum trecho da viagem.

Entrada

A primeira linha contém três inteiros N , M e C ($1 \leq N \leq 10^5$, $2 \leq M \leq 10^5$ e $1 \leq C \leq 50$), representando, respectivamente, a quantidade de pedidos, a quantidade de cidades da rota e o número de assentos do ônibus.

Cada uma das próximas N linhas contém três inteiros L_i , R_i e P_i ($1 \leq L_i < R_i \leq M$ e $1 \leq P_i \leq 10^9$), indicando que o passageiro do pedido i deseja embarcar na cidade L_i , desembarcar na cidade R_i e pagaria P_i reais pela passagem.

Saída

Imprima um único inteiro: o maior valor total, em reais, que a Viação Mandacaru pode arrecadar.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
5 5 2 1 3 8 2 5 10 1 2 6 3 5 7 2 4 9	31

Explicação do exemplo 1

Uma possibilidade ótima é emitir os quatro primeiros pedidos, arrecadando $8 + 10 + 6 + 7 = 31$ reais. Os desembarques nas cidades 2 e 3 liberam assentos antes dos novos embarques nessas cidades.

Exemplo de entrada 2

```
4 6 1
1 4 10
1 2 4
2 3 4
3 6 4
```

Exemplo de saída 2

```
12
```

Explicação do exemplo 2

Como o ônibus possui apenas um assento, não é possível emitir passagens para viagens que compartilhem algum trecho. Emitindo os três últimos pedidos, o mesmo assento pode ser reutilizado nas cidades 2 e 3, e a empresa arrecada $4 + 4 + 4 = 12$ reais.