

Problem A. Arthur Lero Lero

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Arthur é uma pessoa extremamente importante e raramente tem tempo para conversar. Para minimizar o tempo gasto com palavras, ele desenvolveu uma linguagem própria, a qual usa diariamente. No entanto, é sempre muito difícil de entender o que sai da sua boca.

Para tentar facilitar sua interação com o resto do mundo, Arthur pediu ajuda ao Complexo Industrial de Narrativas (CIN) para desenvolver um programa que, ao receber um fala S produzida por Arthur e um padrão w , imprima de quantas maneiras distintas podemos obter w somente pela remoção de caracteres de S . Duas maneiras são consideradas diferentes se existe $0 \leq i < |S|$ tal que S_i está presente em uma mas ausente na outra.

Como você é considerado o programador mais experiente do CIN, seu time lhe confiou a tarefa de construir esse programa.

Input

A primeira linha contém $S (1 \leq |S| \leq 10^5)$, uma string representando a fala de Arthur. S contém somente letras minúsculas e maiúsculas.

A segunda e última linha contém $w (1 \leq |w| \leq 10)$, a string sendo buscada na fala de Arthur.

Output

Imprima um único inteiro: o resto da divisão da quantidade de maneiras distintas de se obter w a partir da remoção de caracteres de S por $1000000007 = 10^9 + 7$.

Examples

standard input	standard output
weewrshkim sim	1
qqqaaabbbcccccffrrr qabcfr	729

Problem B. Batalha Naval

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Na viagem para Foz, os competidores se entreteram jogando Batalha Naval. Enquanto discutiam a melhor estratégia para ganhar o jogo, surgiu a ideia de implementarem um simulador.

O simulador deve responder aos seguintes comandos:

- 1. Colocar um navio no mapa. Se a colocação do navio o pusesse no topo de outro, não colocar.
- 2. Somar o número de posições do mapa ocupadas em um segmento de linha ou coluna.

Quando o simulador é iniciado, todas as posições do mapa estão desocupadas. Todos os comandos do tipo 1 só podem ocupar até 6 posições (o tamanho do maior navio no jogo).

Infelizmente, eles estão muito cansados de tanto jogar Batalha Naval e pediram pela sua ajuda para implementar o simulador.

Input

A primeira linha contém 3 inteiros: N , M , Q ($1 \leq N, M \leq 10^9$ and $1 \leq Q \leq 10^5$). N e M são, respectivamente, o número de linhas e de colunas no mapa.

As próximas Q linhas descrevem um comando cada. Cada descrição começa com um inteiro $query_type \in \{1, 2\}$.

- Se $query_type = 1$, seguem $ship_type\ x\ y\ size$ ($ship_type \in \{L, C\}$; $1 \leq x \leq M$; $1 \leq y \leq N$; e $1 \leq size \leq 6$).
- Se $query_type = 2$, seguem $range_type\ x\ y\ size$ ($range_type \in \{L, C\}$; $1 \leq x \leq M$; $1 \leq y \leq N$; e $1 \leq size \leq M$ se $range_type = L$, $1 \leq size \leq N$ caso contrário).

Se $ship_type = L$ o navio é colocado numa linha. Caso contrário, é colocado numa coluna. (O mesmo é válido para $range_type$).

Todos os navios são colocados completamente dentro do mapa.

Output

Para cada comando do tipo 1, imprima -1 se é impossível colocar o navio.

Para cada comando do tipo 2, imprima a quantidade de células ocupadas no segmento especificado.

Example

standard input	standard output
5 5 3 1 L 1 1 3 1 C 5 1 3 2 L 1 1 5	4

Problem C. Café da manhã

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 megabytes**

Diogo e Lucas estão em São José dos Campos para participar de uma competição na linguagem Elixir. O evento acontece hoje pela manhã, então eles devem encontrar rapidamente uma padaria onde possam tomar café da manhã.

Diogo procurou no mapa as padarias mais próximas de onde estavam e encontrou duas padarias (com localizações distintas) que pareciam ser legais, e pediu para Lucas escolher uma delas.

Lucas então disse para escolher a padaria mais próxima do local do evento e, caso as duas estejam à mesma distância, a mais próxima da posição atual deles.

Porém, Diogo disse que as duas estavam empatadas segundo o critério de Lucas, e inclusive a distância das padarias para o local do evento é igual à distância da posição atual deles para o evento.

Agora Lucas está se perguntando se isto é possível, e em caso afirmativo quais são duas possíveis localizações para as padarias.

Input

A entrada consiste de 5 números inteiros, A_x, A_y, B_x, B_y, D ($|A_x|, |A_y|, |B_x|, |B_y|, D \leq 10^4$). Onde (A_x, A_y) é a localização atual deles, (B_x, B_y) representa a localização do evento e D representa a distância da localização atual deles para as padarias.

Output

Na primeira linha da saída imprima “YES” caso seja possível e “NO” caso contrário. Caso seja possível, a segunda linha deve conter 4 números reais P_x, P_y, Q_x, Q_y . Onde (P_x, P_y) e (Q_x, Q_y) indicam as posições das duas padarias.

A saída deve possuir erro máximo de 10^{-5} .

Examples

standard input	standard output
0 5 0 0 7	YES -4.998999990 0.100000000 4.998999990 0.100000000
0 5 0 0 100	NO

Problem D. Droid perdido

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

O Centro Interestelar de Navegação (CIN) lançou uma sonda em um planeta distante. Para saber a localização precisa da nova sonda, o centro irá manter um satélite acompanhando os passos dela. Dessa forma, o satélite precisa primeiro encontrar a sonda no terreno do planeta desconhecido, que pode ser considerado uma matriz de N linhas e M colunas.

Durante sua operação, a sonda emite o log das ações tomadas a cada instante de tempo. Sendo (X_i, Y_i) a posição da sonda no instante i , essas ações podem ser:

- C : A sonda se move para cima ($X_{i+1} = X_i + 1$)
- D : A sonda se move para a direita ($Y_{i+1} = Y_i + 1$)
- B : A sonda se move para baixo ($X_{i+1} = X_i - 1$)
- E : A sonda se move para a esquerda ($Y_{i+1} = Y_i - 1$)

No instante T lhe será fornecido todos os logs emitidos pela sonda desde o instante 0. Sua tarefa é computar todas as posições que precisam ser verificadas no instante T para que a sonda seja encontrada com certeza.

O satélite é capaz de analisar quantas posições quanto forem necessárias, mas você não deve verificar nenhuma posição para a qual você tenha informações suficiente a ponto de saber que a sonda não pode estar ali.

Input

A primeira linha contém 2 inteiros separados por espaço, N e M ($1 \leq N, M \leq 10^6$, $1 \leq N \times M \leq 10^6$), o número de linhas e o número de colunas da matriz.

A segunda linha contém 1 inteiro T ($1 \leq T \leq 10^6$), o instante atual. T linhas seguem. A $(i + 2)$ -ésima linha contém 1 caractere, l_i ($l_i \in \{C, D, B, E\}$), o log da sonda no instante i .

Output

Na primeira linha, imprima 1 inteiro, Q ($1 \leq Q \leq N * M$), o número de posições a serem verificadas.

Em seguida, imprima Q linhas. Na $(j + 1)$ -ésima linha, imprima 2 inteiros separados por espaço, X_j ($1 \leq X_j \leq N$) e Y_j ($1 \leq Y_j \leq N$), as j -ésimas coordenadas da matriz a serem verificadas.

Examples

standard input	standard output
3 3	2
3	1 3
D	2 3
D	
B	
1 3	2
1	1 1
E	1 2

Problem E. Equivalência

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 64 megabytes

Como Guga estava sem nada interessante para fazer, ele resolveu ler o dicionário e aprendeu uma nova palavra:

- Palíndromo: diz-se de ou frase ou palavra que se pode ler, indiferentemente, da esquerda para a direita ou vice-versa.

ovo, **ana**, **x** e **arara** são exemplos de palavras palíndromas.

Analisando algumas palavras, Guga percebeu que a maioria não é um palíndromo e resolveu criar uma nova categoria chamada **K-Palíndromo**, cuja definição é:

- K-Palíndromo: diz-se de ou frase ou palavra que substituindo até K letras é equivalente a um palíndromo.

Por exemplo, **xyz** não é um palíndromo, mas é um 1-Palíndromo, pois podemos substituir o **x** por um **z** e obter **zyz**. Já **abcb** não é um palíndromo nem um 1-Palíndromo, mas é um 2-Palíndromo, pois podemos substituir duas letras e obter um palíndromo.

Com essa nova definição, Guga conseguiu encontrar muito mais palavras que a satisfizessem.

Enquanto pesquisava sobre palíndromos, Guga descobriu o seguinte problema: dado uma string S , quantas substrings contínuas de S são palíndromas?

Intrigado, ele resolveu criar um problema equivalente: dado uma string S e um inteiro K , quantas substrings contínuas de S são K -palíndromas?

Guga acha que é muito fácil achar uma resposta errada para o problema dele com strings grandes, e por isso quer um programa de computador que faça isso. Ajude Guga (:

Input

Cada caso de teste consiste de duas linhas.

A primeira linha contém uma string S ($1 \leq |S| \leq 3000$) composta apenas por letras minúsculas, a string que Guga quer achar os K -Palíndromos.

A segunda linha contém um inteiro K ($0 \leq K \leq |S|/2$), a quantidade de letras que podem ser substituídas em S .

Output

A saída consiste de uma única linha contendo um inteiro X , o número de substrings de S que são K -palíndromas.

Examples

standard input	standard output
ovo 1	6
ovo 0	4

Note

Por definição:

- Todo K -Palíndromo é também um K' -Palíndromo, onde $K' > K$.
- Todo palíndromo é um 0-palíndromo.

Problem F. Floresta Bolada

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Em suas férias, Bolado decidiu se aventurar. Entediado com a pacata vida urbana, decidiu viajar para as montanhas, onde teria tempo para desbravar novos lugares e pensar em piadas. Ao chegar em seu destino, Bolado percebeu que estava rodeado de uma vasta e densa floresta com árvores de variados tipos, o que logo despertou seus instintos, levando-o ao seguinte pensamento:

"Por que o pinheiro nunca se perde numa floresta?"

"Porque ele tem uma pinha!"

Contente com a piada que elaborou, Bolado só pensava nela enquanto procurava pela sua hospedaria de férias. Entretanto, não demorou para que o mirabolante piadista percebesse que estava perdido. Infelizmente, Bolado não é um pinheiro pra ter pinha, mas, felizmente, ele tem um mapinha (yay!).

No mapa, é possível ver N áreas, numeradas de 1 a N . Bolado encontra-se perdido na área 1 e deseja chegar a sua hospedaria que se encontra na área N . Também é possível distinguir M conexões direcionadas no mapa que conectam duas áreas distintas. Cada uma das conexões é acompanhada da descrição do tempo necessário pra realizar sua travessia em minutos.

Adicionalmente, por questões de preservação ambiental, P das N áreas apresentam um pinheiro plantado em cada um de seus territórios. Bolado ao chegar em uma dessas áreas, precisa rir por K segundos por se recordar de sua obra. Não haverá pinheiros na área 1, nem na área N . Faltam T minutos para anoitecer e Bolado quer chegar à hospedaria o mais rápido possível, para obviamente pensar em piadas relativas à hospedaria.

Input

A primeira linha contem 5 inteiros, N ($2 \leq N \leq 3 * 10^4$), M ($0 \leq M \leq 10^5$), T ($0 \leq T \leq 5 * 10^7$), K ($1 \leq K \leq 5 * 10^7$), P ($0 \leq P \leq N - 2$) como descritos acima. Os P inteiros p_i seguintes representam os índices das áreas que contêm pinheiro plantado ($1 \leq p_i \leq N, \forall i : 1 \leq i \leq P$). Em seguida, haverá M linhas descrevendo cada uma das conexões, por meio de 3 inteiros em cada linha, x_j , y_j e w_j que indicam que Bolado pode se deslocar da área x_j à área y_j em w_j minutos, para $1 \leq j \leq M$ ($x_j \neq y_j; 1 \leq x_j, y_j \leq N$ and $1 \leq w_j \leq 10^5$).

Output

Imprima o tempo mínimo utilizado por Bolado para chegar à hospedaria. Caso ele não consiga chegar antes ou no mesmo instante do sol se por, imprima "-1" sem as aspas.

Examples

standard input	standard output
5 7 312 10 2 3 2 1 2 8 4 5 98 3 2 12 5 2 30 5 1 103 3 4 65 2 3 1	10340
4 6 29370 22446 1 3 4 2 32014 2 3 24 2 1 67 4 3 16 1 2 633 2 4 4298	295860
3 2 701 8561 1 2 2 1 346 3 1 9	-1

Problem G. Gosto de Tu Não!

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 64 megabytes

Fada é um empresário muito famoso que adora dar festas para seus sócios. Certo dia ele resolve fazer um jantar como nunca antes e reunir todos os seus sócios em um só encontro. Ele pediu para ser alocada para os convidados uma mesa redonda, de modo que ele possa ver todo mundo da mesa quando se sentar. Como Fada é gente fina, não tem nenhuma restrição sobre em qual lugar ele irá se sentar à mesa. Entretanto, por mais que seu carisma contagie a todos, seus sócios só vão aos jantares organizados por Fada por conta dele. Na verdade, eles odeiam todos os outros sócios senão Fada, e até brigam entre si se estiverem perto um do outro. Consequentemente eles exigem nos jantares estarem separados por no mínimo uma quantidade R de cadeiras, para que nem mesmo consigam ver direito o rosto dos outros sócios.

Fada, por ser muito curioso, gostaria de saber de quantas formas ele pode organizar seus sócios na mesa de modo que não briguem. Entretanto ele é muito ocupado e por isso pede a você para fazer essas contas por ele. Você irá receber o número N de cadeiras da mesa ($1 \leq N \leq 10^5$), o número C de convidados ($1 \leq C \leq N$) e a quantidade R de cadeiras que precisa pelo menos ter entre os convidados ($1 \leq R \leq N$). Ajude Fada a descobrir a quantidade de maneiras de ajustar seus convidados.

Input

Três inteiros N , C e R como descrito no enunciado.

Output

Imprima a quantidade de maneiras de se colocar os convidados à mesa. Como o número pode ser grande, calcule-o módulo $(10^9 + 7)$.

Se não for possível colocar os colocados à mesa respeitando as restrições, imprima -1 .

Examples

standard input	standard output
9 3 2	18
10 2 1	280

Note

Duas maneiras são distintas se o conjunto de cadeiras ocupadas pelos convidados é distinto ou se a cadeira na qual Fada se senta é distinta.

Problem H. Hoje tem Pão de Queijo

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 64 megabytes

Voltando de Foz, Gabi e Divino estavam indo para o portão de embarque. Gabi, com fome, percebeu a grande quantidade de Casas do Pão de Queijo no caminho e se perguntou qual caminho deveriam tomar para passar pelo máximo de Casas do Pão de Queijo. Divino, por outro lado, não estava com tanta fome, e imaginou qual caminho minimizaria a quantidade de Casas do Pão de Queijo.

Adicionalmente, o aeroporto Boladense tem uma propriedade muito estranha. As passarelas entre os terminais são unidirecionais e, se existe um caminho partindo do Terminal A para o Terminal B, não existe um caminho do Terminal B para o Terminal A.

Inicialmente eles se encontram no Terminal 1 e o Portão fica no Terminal N .

Input

A primeira linha contém 3 inteiros, N , C e M ($1 \leq N \leq 10^5$, $1 \leq C \leq N$, $1 \leq M \leq 10^5$) indicando, respectivamente, o número de terminais (identificados de 1 a N), Casas do Pão de Queijo e passarelas. A segunda linha contém C inteiros, os índices dos terminais que têm uma Casa do Pão de Queijo. Por fim, as próximas M linhas contém 2 inteiros v e w , indicando que existe uma passarela de v para w .

Output

A saída deve conter uma única linha com 2 inteiros separados por espaço Min e Max , os valores mínimo e máximo de Casas do Pão de Queijo pelas quais eles podem passar ao ir do Terminal 1 ao Terminal N .

Example

standard input	standard output
5 3 6 1 3 4 1 2 2 3 3 4 4 5 1 5 1 4	1 3

Problem I. Ih! Onde estou?

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 megabytes**

Tiago mora numa rua com N casas, numeradas de 1 a N , sendo a primeira a dele. Ele decidiu sair do Codeforces e visitar um amigo, mas a cabeça dele está presa num problema há vários dias. Por causa disso, ele está um pouco distraído e não sabe exatamente para onde ele está indo.

A caminhada pode ser considerada como K passos, onde se ele estiver na casa i em um passo, $\frac{A_{ij}}{10^6}$ é a probabilidade dele estar na casa j no próximo passo. Ele quer saber qual é o valor esperado de onde ele vai parar, mas está distraído demais para resolver esse problema.

Input

A primeira linha da entrada consiste de dois inteiros N e K ($1 \leq N \leq 100$, $1 \leq K \leq 10^9$). As próximas N linhas contém cada uma N inteiros, onde o j -ésimo número na i -ésima linha é A_{ij} ($0 \leq A_{ij} \leq 10^6$) e a soma de cada linha é 10^6 .

Output

A resposta pode ser sempre representada como uma fração irredutível $\frac{P}{Q}$. Imprima o inteiro $P * Q^{-1}$ módulo $10^9 + 7$ (Q^{-1} é o inverso modular de Q módulo $10^9 + 7$).

Examples

standard input	standard output
2 2 500000 500000 500000 500000	500000005
4 24255 283383 274729 386113 55775 167247 49080 415640 368033 449168 182769 352198 15865 127605 304475 230640 337280	96636490
2 2 0 1000000 1000000 0	1
2 1 0 1000000 1000000 0	2

Problem J. Joana e as Baterias

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

Joana trabalha no BBB (Banco de Baterias do Brasil). Ela foi encarregada de organizar o deslocamento das baterias de forma ótima mas não conseguiu resolver o problema. Sabendo que você é um programador, ela pediu sua ajuda.

No BBB, as baterias são organizadas em pilhas. Em um banco, as baterias não podem estar descarregadas. Para isso, o banco gasta M unidades de energia onde M é o tamanho da maior pilha. Cada pilha tem inicialmente a_i baterias.

Está disponível uma máquina que move uma bateria de uma pilha i para a pilha j com custo $b_i + c_j$. Joana tem um orçamento limitado de K que ele pode usar para mover as baterias e o banco quer que Joana atinja o menor M possível.

Input

A primeira linha da entrada consiste dos inteiros N e K ($1 \leq N \leq 10^5$, $1 \leq K \leq 10^{18}$). As próximas N linhas têm a_i , b_i e c_i ($0 \leq a_i, b_i, c_i \leq 10^6$).

Output

Imprima o menor M possível.

Example

standard input	standard output
5 20 5 3 8 2 8 3 3 2 4 7 2 1 6 1 1	5

Problem K. Kd o enunciado?

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

Lucas queria muito fazer uma questão interessante de strings, mas ele passou tanto tempo pensando que ficou sem criatividade para criar um enunciado melhor para a questão.

O problema que ele pensou foi o seguinte:

Dado um vetor v de strings, você deve ser capaz de realizar as seguintes queries:

- 1. $update(i, s)$ substitua a i -ésima posição do vetor pela string s .
- 2. $vprefix(l, r, s)$ verifique se alguma string do subvetor $[l, r]$ é prefixo de s .
- 3. $sprefix(l, r, s)$ verifique se s é prefixo de alguma string do subvetor $[l, r]$.

Input

Na primeira linha da entrada encontra-se um inteiro n ($1 \leq n \leq 10^5$) indicando o tamanho do vetor v . Nas próximas n linhas tem-se, em ordem, cada string do vetor, onde a $(i+1)$ -ésima linha contém a string na posição i do vetor.

Na $(n+2)$ -ésima linha encontra-se um número inteiro q ($1 \leq q \leq 10^5$), indicando o número de queries que serão realizadas. Em cada uma das próximas q linhas encontram-se as descrições das q queries, uma em cada linha.

Cada query é descrita da seguinte forma:

- Tipo 1: 1 i s , onde i é um inteiro ($1 \leq i \leq n$) e s é uma string.
- Tipo 2: 2 l r s , onde l e r são inteiros ($1 \leq l \leq r \leq n$) e s é uma string.
- Tipo 3: 3 l r s , onde l e r são inteiros ($1 \leq l \leq r \leq n$) e s é uma string.

Output

Para cada query do tipo 2 e 3, você deve mostra na tela “Y” se o retorno da query é verdadeiro ou “N” se é falso.

Example

standard input	standard output
3	Y
aaa	N
ab	Y
acc	Y
10	Y
2 1 3 abb	N
2 3 3 abb	Y
2 1 1 aaa	N
3 1 3 a	Y
3 1 3 ac	
3 3 3 b	
3 3 3 ac	
3 1 3 e	
1 1 eae	
3 1 3 e	

Note

- A soma dos tamanhos das strings no vetor inicialmente não ultrapassa $5 * 10^5$
- A soma dos tamanhos das strings presentes nas queries não ultrapassa $5 * 10^5$
- Todas as strings contém apenas caracteres minúsculos.
- A string a é prefixo da string b se: $\exists x \in \Sigma^* | ax = b$