

Caderno de Problemas

6^a maratona unballoon de programação

17/05/2025

(Este caderno contém 14 problemas)

Organizado por:

Arthur Botelho
Daniel Porto
Eduardo Freire
Eduardo Quirino
Enzo Yoshio
Isabela Souza
Lucas Sala
Pedro Gallo
Ruan Petrus
Wallace Ben
Wilson Guimarães

Universidade de Brasília, Campus Darcy Ribeiro

Lembretes

- Não é permitido a utilização de aparelhos eletrônicos além do computador destinado à seu time.
- Não discuta problemas com outros times.
- É permitido a consulta de qualquer material impresso trago pelo time.
- As soluções enviadas são julgadas de maneira automática, portanto é necessário seguir de maneira estrita o formato indicado de entrada e saída em cada problema. Deve-se utilizar as entradas e saídas padrões.
- Cada problema é julgado utilizando tanto testes públicos, visíveis ao participante, quanto testes secretos, que nunca são revelados ao participante.
- Caso tenha uma dúvida, utilize a aba *clarifications*. As dúvidas serão respondidas ou não à critério dos juízes, que também podem escolher tornar a dúvida e sua resposta pública.
- Caso queira enviar uma solução em Java, o nome do arquivo da solução deve ser o mesmo que o nome da classe principal do programa.

Problem A. Acessando montanhas

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

Isa, buscando se juntar às camelas escaladoras, decidiu começar a escalar montanhas. Para que ela seja aceita como uma camela escaladora, Isa deve escalar todas as N montanhas disponíveis. Porém, para que Isa consiga a permissão para escalar uma montanha de um certo nível, ela deve ter escalado outras montanhas mais fáceis primeiro.

Sua missão é: dada a quantidade de montanhas e pares de montanhas (A, B) em que a montanha de nível A deve ser escalada antes da montanha de nível B , dar a menor ordem lexicográfica que Isa pode seguir para que ela se torne uma camela escaladora.

Uma ordem A é lexicograficamente menor que outra ordem B caso exista um i tal que $A_i < B_i$ e para todo j tal que $1 \leq j < i$, $A_j = B_j$.

Input

Na primeira linha são dados 2 inteiros N e M ($1 \leq N, M \leq 2 \cdot 10^5$).

Em cada uma das próximas M linhas são dados 2 inteiros A e B ($1 \leq A < B \leq N$), em que para escalar a montanha de nível B é necessário que Isa tenha escalado a montanha de nível A primeiro. $A < B$, já que uma montanha de nível menor é mais fácil.

Output

Imprima N números que representem a menor ordem lexicográfica que Isa deve seguir para se tornar uma camela escaladora ou -1 caso não exista nenhuma ordem em que Isa consiga escalar todas as montanhas.

Examples

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

Problem B. Baralho Infinito

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

Henrique Ramos está muito feliz com sua nova compra: um baralho especial com infinitas cartas!

Cada carta desse baralho tem um número escrito nela. Ao comprar seu baralho, Henrique precisou escolher um inteiro positivo n , que determina os números escritos nas cartas.

O número da primeira carta é n^0 , o da segunda é n^1 , o da terceira é n^2 e, em geral, a i -ésima carta tem o número n^i . Assim, cada carta é única, ou seja, não há duas cartas com o mesmo número.

Henrique vai jogar um jogo com seus amigos utilizando seu novo baralho. Nesse jogo, uma jogada é definida por um subconjunto finito de cartas do baralho, com exceção do subconjunto sem cartas. Além disso, o valor de cada jogada é a soma dos números de suas cartas.

Por exemplo, o baralho de Henrique para $n = 3$ contém as cartas 1, 3, 9, 27, 81 e outras. Ainda no mesmo caso, exemplos de jogadas válidas são:

- 1, essa jogada tem o valor 1;
- 1, 3, essa jogada tem o valor 4;
- 3, 243, 729, essa jogada tem o valor 975.

Henrique está interessado nas jogadas com menor valor e, para isso, criou a sequência de todos os valores possíveis de jogadas. Essa sequência é ordenada pelo valor de cada jogada. Por exemplo, os cinco primeiros termos dessa sequência para $n = 3$ são: 1, 3, 4, 9, 10.

Henrique quer saber qual é o k -ésimo termo dessa sequência. Porém, neste momento, ele está muito ocupado lendo a questão *Collatz-Estrela da Dor e do Sofrimento*. Ajude Henrique a calcular o valor do k -ésimo termo dessa sequência.

Como a resposta pode ser um número muito grande, imprima apenas o resto da divisão desse valor por $10^9 + 7$.

Input

A entrada consiste de uma única linha que contém dois inteiros n e k ($2 \leq n \leq 10^9, 1 \leq k \leq 10^9$). n é o número do baralho que Henrique comprou e k é o índice em que Henrique está interessado.

Output

Imprima uma única linha contendo um inteiro: o resto da divisão do k -ésimo termo da sequência por $10^9 + 7$.

Examples

standard input	standard output
3 4	9
4 5	17
3 98	975

Note

No primeiro caso de teste, as cinco primeiras jogadas, ordenadas pelo menor valor, são: 1, 3, 4, 9, 10. O quarto valor da sequência é 9.

Já no segundo caso de teste, as cinco primeiras jogadas são: 1, 4, 5, 16, 17. O quinto valor da sequência é 17.

Problem C. Collatz-Estrela da Dor e do Sofrimento

Input file: standard input
Output file: standard output
Time limit: 6.5 seconds
Memory limit: 512 megabytes

Desde a última Grande Guerra, a Galáxia tem sido governada pelo Império Galáctico. Lord Edsidious finalmente conseguiu conquistar o status de Imperador, após séculos de planejamento. Em seu Império, as seguintes definições e convenções se aplicam:

- $X \% Y$ representa o resto da divisão inteira de X por Y se $X \geq 0$ e $Y > 0$. Além disso, $X \% Y = X$ se $X < 0$ e $Y > 0$, e $X \% Y = 0$ se $Y = 0$.
- $NCP(X, Y)$ é o menor inteiro $K \geq X$ tal que $\gcd(K, Y) = 1$.
- A Função de Super Shift, SSF , é definida como: $SSF(X, Y, Z) = ((X - Z) \% (Y - Z)) + Z$.
- A Função de Collatz, CF , é definida como:
$$CF(X) = \begin{cases} \frac{X}{2} & \text{se } X \text{ é par} \\ \frac{3 \cdot X + 1}{4} & \text{se } X \% 4 = 1. \\ 3 \cdot X + 1 & \text{se } X \% 4 = 3 \end{cases}$$
- Sobre o valor de uma fração mod X : considere a fração $\frac{P}{Q}$ com Q coprimo a X (se não for, o valor não está definido). O valor dessa fração mod X é $(P \cdot Q^{-1}) \% X$, onde Q^{-1} é o valor tal que $(Q \cdot Q^{-1}) \% X = 1$ e $0 \leq Q^{-1} < X$ (pode ser provado que existe um único Q^{-1} se Q for coprimo a X).
- Sobre a notação de grid: um grid $X \times Y$ tem X linhas e Y colunas. Uma célula (i, j) do grid está na i -ésima linha e j -ésima coluna. Um subgrid $Z = (Z_{x1}, Z_{y1}, Z_{x2}, Z_{y2})$ é o conjunto de células (i, j) com $Z_{x1} \leq i \leq Z_{x2}$ e $Z_{y1} \leq j \leq Z_{y2}$.

Para manter sua dominância, Lord Edsidious contratou o famoso botânico Collatz (o criador da Função de Collatz) para construir uma Collatz-Estrela da Dor e do Sofrimento, que é um dispositivo massivo capaz de exterminar planetas inteiros com um único disparo. E para proteger essa arma estratégica, ele designou seu general mais prestigiado: Darth Caetanus. Caetanus recentemente tomou conhecimento de uma fraqueza na Estrela: sua placa de energia.

A placa de energia é composta por células independentes, dispostas em um grid $H \times W$, e está associada a um número inteiro fixo $M > 2$. Uma célula (i, j) , a qualquer momento t (medido em alguma unidade inteira) tem dois inteiros positivos associados a ela: $A_{t,i,j}$ e $B_{t,i,j}$. Esses números descrevem a probabilidade de uma falha na célula naquele momento, que é igual a $\frac{A'_{t,i,j}}{B'_{t,i,j}}$, onde $B'_{t,i,j} = NCP(SSF(B_{t,i,j}, M, 2), M)$ e $A'_{t,i,j} = SSF(A_{t,i,j}, B'_{t,i,j}, 1)$. A qualquer momento, se uma falha ocorrer em qualquer célula, ela desliga, deixando a Collatz-Estrela sem energia temporariamente. Esses valores A e B variam no tempo da seguinte forma: $A_{t+1,i,j} = CF(A_{t,i,j})$ e $B_{t+1,i,j} = CF(B_{t,i,j})$.

Vinicius Solo, uma pessoa influente na caída República Galáctica, usou suas conexões para encontrar (no tempo 0) todos os números $A_{0,i,j}$ e $B_{0,i,j}$ e o número fixo M . Ele também tomou conhecimento de que, em um certo tempo desconhecido T , Lord Edsidious ordenará que a Collatz-Estrela dispare contra o pobre planeta U.n.Balloon. Para evitar sua destruição, Vinicius precisará da ajuda de seu aliado Daniel Skywalker, que pode usar o Foco Místico em subgrids do grid da placa de energia.

Vamos considerar que Daniel tenta usar Foco Místico em um subgrid S em um período de tempo de t_l a t_r (inclusive), e seja $Prob(S, t_l, t_r)$ a probabilidade de que qualquer célula em S desligue em algum momento entre t_l e t_r (o que significa que a Collatz-Estrela perderia energia nesse intervalo de tempo). Daniel pode explodir as células em S usando sua habilidade, mas apenas se o valor de $Prob(S, t_l, t_r)$ mod M for maior que $\lfloor \frac{M}{2} \rfloor$ (pode ser provado que $Prob(S, t_l, t_r)$ é sempre representável por uma fração $\frac{P}{Q}$ com Q coprimo a M). Essa explosão provocaria uma reação em cadeia que destruiria toda a Collatz-Estrela!

No entanto, salvar U.n.Balloon não será tão fácil. Para proteger a Collatz-Estrela, Darth Caetanus lançou um feitiço que impede Daniel de tentar usar o Foco Místico em subgrids S que não estejam contidos em um subgrid especial desconhecido K . Além disso, se Daniel tentar usar Foco Místico em algum subgrid S , ele será forçado a fazê-lo de $t_l = S_{x1} + S_{y1} + K_{x1} + K_{y1}$ a $t_r = T - (S_{x2} + S_{y2} + K_{x2} + K_{y2})$.

Vinicius e Daniel não sabem qual é o tempo T ou o subgrid especial K , então decidiram se preparar. Eles realizarão Q estudos: no estudo i , contarão, para T_i e K_i escolhidos, o número de subgrids S_i nos quais Daniel seria capaz de usar o Foco Místico para explodir, causando a destruição da Collatz-Estrela. Ajude-os a escrever um programa que calcule isso!

Input

A primeira linha de entrada contém os inteiros H, W ($1 \leq H, W \leq 30$) e M ($2 < M \leq 10^5$).

As próximas H linhas contêm W inteiros cada: os valores $A_{0,i,j}$ ($0 < A_{0,i,j} \leq 10^5$).

As outras próximas H linhas também contêm W inteiros cada: os valores $B_{0,i,j}$ ($0 < B_{0,i,j} \leq 10^5$).

A próxima linha de entrada contém o inteiro Q ($1 \leq Q \leq 100$).

Finalmente, as próximas Q linhas contêm cinco inteiros cada: T_i ($0 \leq T_i \leq 10^5$) e os $x1, y1, x2$ e $y2$ (nesta ordem) relacionados a K_i ($1 \leq x1, x2 \leq H, 1 \leq y1, y2 \leq W$).

Output

Imprima Q linhas: na linha i , o número de subgrids S_i nos quais Daniel seria capaz de usar o Foco Místico para explodir, considerando os T_i e K_i dados e os correspondentes t_l e t_r para cada subgrid (como descrito pelo enunciado).

Examples

standard input	standard output
3 3 9 12 31 46 20 35 28 91 54 18 11 14 16 10 15 77 27 15 13 4 11 1 1 2 2 30 1 3 2 3 0 1 1 3 3 8 1 1 1 1	1 0 0 1
3 3 3 55 83 63 73 62 95 95 97 27 63 55 41 95 47 63 95 54 47 4 30 2 3 3 3 20 3 3 3 3 24 1 2 3 2 36 2 1 3 2	0 0 4 4

Note

Vamos entender o primeiro estudo do primeiro caso de teste, com $T = 11$ e $K = (1, 1, 2, 2)$. O valor de M é 9.

Com esses valores, a maioria dos subgrids tem $t_l > t_r$, e assim não podem ser alvo do Foco Místico de Daniel. Os únicos possíveis são $S_1 = (1, 1, 1, 1)$ ($t_l = 4, t_r = 5$), $S_2 = (1, 1, 2, 1)$ e $S_3 = (1, 1, 1, 2)$ (ambos com $t_l = t_r = 4$).

Começando com S_1 :

No tempo 4, podemos verificar que $A_{4,1,1} = 5$ e $B_{4,1,1} = 10$. Calculando $B'_{4,1,1} = NCP(SSF(10, 9, 2), 9)$, temos $NCP((8 \% 7) + 2, 9) = NCP(3, 9) = 4$. Segue que $A'_{t,i,j} = SSF(5, 4, 1) = (4 \% 3) + 1 = 2$ e que $Prob(S_1, 4, 4) = \frac{2}{4}$.

No tempo 5, $A_{5,1,1} = 4$ e $B_{5,1,1} = 5$. Assim, calculamos $B'_{5,1,1} = NCP(SSF(5, 9, 2), 9)$ como $NCP((3 \% 7) + 2, 9) = NCP(5, 9) = 5$ e $A'_{t,i,j} = SSF(4, 5, 1)$ como $(3 \% 4) + 1 = 4$. Isso resulta que $Prob(S_1, 5, 5) = \frac{4}{5}$.

Podemos usar $Prob(S_1, t_l = 4, t_r = 5) = Prob(S_1, 4, 4) \cup Prob(S_1, 5, 5)$ para calcular a probabilidade como $Prob(S_1, 4, 4) + Prob(S_1, 5, 5) - Prob(S_1, 4, 4) \cdot Prob(S_1, 5, 5)$. Aplicando os valores, obtemos $Prob(S_1, 4, 5) = \frac{2}{4} + \frac{4}{5} - \frac{2}{4} \cdot \frac{4}{5} = \frac{18}{20}$. O valor dessa fração mod 9 é $(18 \cdot 20^{-1}) \% 9 = (18 \cdot 5) \% 9 = 0$. Isso não é maior que $\lfloor \frac{9}{2} \rfloor = 4$, então Daniel não pode usar o Foco Místico aqui.

Agora com S_2 :

Vamos tomar $S_2^1 = (1, 1, 1, 1)$ e $S_2^2 = (2, 1, 2, 1)$. Então, $Prob(S_2, t_l = 4, t_r = 4) = Prob(S_2^1, 4, 4) \cup Prob(S_2^2, 4, 4)$. Já calculamos $Prob(S_2^1, 4, 4) = Prob(S_1, 4, 4) = \frac{2}{4}$.

Temos que $A_{4,2,1} = 1$ e $B_{4,2,1} = 1$. Assim, $B'_{4,2,1} = NCP(SSF(1, 9, 2), 9)$, o que resulta $NCP((-1) \% 7 + 2, 9) = NCP(-1 + 2, 9) = 1$, e $A'_{4,2,1} = SSF(1, 1, 1)$, que dá $(0 \% 0) + 1 = 0 + 1 = 1$. Com isso, obtemos $Prob(S_2^2, 4, 4) = \frac{1}{1}$. Isso resulta em $Prob(S_2, t_l = 4, t_r = 4) = \frac{2}{4} + \frac{1}{1} - \frac{2}{4} \cdot \frac{1}{1} = \frac{4}{4}$, que tem valor 1 em mod 9 (ainda não maior que $\lfloor \frac{9}{2} \rfloor = 4$).

Por último, com S_3 :

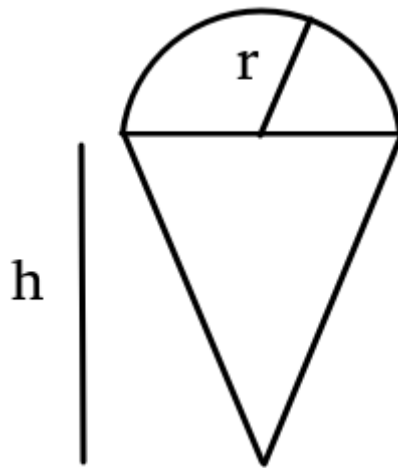
De forma semelhante a S_2 , vamos tomar $S_3^1 = (1, 1, 1, 1)$ e $S_3^2 = (1, 2, 1, 2)$. $A_{4,1,2} = 71$, $B_{4,1,2} = 34$, $B'_{4,1,2} = NCP(SSF(34, 9, 2), 9) = NCP(6) = 7$ e $A'_{4,1,2} = SSF(71, 7, 1) = 5$. Então, $Prob(S_3^2, 4, 4) = \frac{5}{7}$ e $Prob(S_3, 4, 4) = \frac{2}{4} + \frac{5}{7} - \frac{2}{4} \cdot \frac{5}{7} = \frac{24}{28}$. Isso tem valor 6 em mod 9, então Daniel conseguiria usar o Foco Místico aqui!

No total, obtemos que há apenas um subgrid de K (nosso S_3) no qual Daniel pode usar sua habilidade. Então, o resultado para este estudo é 1.

Problem D. Delírio na Unballoon

Input file: standard input
Output file: standard output
Time limit: 0.5 seconds
Memory limit: 256 megabytes

Os integrantes da Unballoon se reuniram na casa da Isa após uma maratona para comemorar e comer sorvete. A maioria das pessoas queria servir seus sorvetes em casquinhas e, então, comê-los. Contudo, Lucas Sala não gosta de casquinhas e, por isso, preferiu servir seu sorvete na sua garrafa de água. Isso gerou uma grande controvérsia, pois alguns dos maratonistas achavam que a garrafa de Sala era capaz de armazenar mais sorvete do que uma casquinha, mas eles não tinham certeza.



Casquinha de altura h e raio da base r , que também é o raio da semi-esfera.

Após vários cálculos com integrais e alguns crimes contra a física básica, os programadores não conseguiram chegar a uma conclusão; por isso, você deve ajudá-los a resolver esse mistério. Para isso, considere que as casquinhas de sorvete são cones de base circular. Além disso, as casquinhas serão preenchidas de sorvete uniformemente até seu topo, onde será colocada uma semi-esfera sólida de sorvete de raio igual ao raio do círculo da base do cone. Por fim, assuma que a garrafa de Lucas Sala é um cilindro, que também será preenchida de sorvete uniformemente até seu topo. **Não será colocada uma semi-esfera sólida de sorvete na garrafa de Lucas Sala.**

Input

A primeira linha e única linha de input contém quatro inteiros $1 \leq r_L, h_L, r_C, h_C \leq 10^2$: o raio da base da garrafa de Lucas Sala, a altura da garrafa, o raio da base do cone que representa sua casquinha e sua altura, respectivamente. Todos são dados em centímetros.

Output

Se a garrafa de Lucas Sala é capaz de armazenar um volume estritamente maior de sorvete do que a casquinha, imprima “Injusto”. Caso contrário, imprima “Justo”.

Examples

standard input	standard output
1 4 1 10	Justo
27 100 45 18	Justo
78 57 60 40	Injusto

Problem E. Escola Mágica de Palavras

Input file: `standard input`
Output file: `standard output`
Time limit: 3 seconds
Memory limit: 512 megabytes

Em um reino distante chamado Alfabétia, os magos não usavam feitiços comuns — eles usavam palavras mágicas. Cada palavra era formada por letras minúsculas do alfabeto, e seu poder não estava nas letras em si, mas na forma como elas se repetiam.

O mais sábio dos magos, o Arquimagus MagePetrus, percebeu algo curioso: duas palavras podiam ser igualmente poderosas, mesmo que tivessem letras diferentes, desde que tivessem o mesmo padrão de repetição.

Por exemplo:

- A palavra "aba" tinha um padrão em que a primeira e a terceira letras eram iguais, e a segunda era diferente.
- A palavra "cbc" tinha o mesmo padrão!
- E até a estranha "z fz" também era assim!

MagePetrus chamou isso de simetria de letras. Ele decretou que, para fins mágicos, palavras com o mesmo padrão de repetição de letras eram equivalentes — não importava quais letras usavam, apenas como elas se repetiam.

Intrigado com as possibilidades, MagePetrus propôs um desafio aos jovens aprendizes de Alfabétia:

Dado um número n , quantos padrões mágicos diferentes (ou seja, quantas classes de palavras distintas por simetria de letras) podem ser formados com palavras de tamanho n , usando letras de “a-z”?

Consideramos duas strings iguais, por simetria de letras, se uma pode ser obtida a partir da outra por meio de uma permutação de suas letras. Formalmente, seja s a string formada pela concatenação dos caracteres $s_1, s_2, \dots, s_n$ nessa ordem e $A = \{a, b, \dots, z\}$ o conjunto das letras minúsculas do alfabeto latino. Dizemos que s é igual a uma string t por simetria de letras se e somente se existe uma função injetora $f : A \rightarrow A$ tal que t é a string formada pela concatenação dos caracteres $f(s_1), f(s_2), \dots, f(s_n)$. A função f ser injetora significa que, para todos os caracteres $x, y \in A$, se $x \neq y$ então $f(x) \neq f(y)$.

Por exemplo, as strings "aba", "cbc" e "z fz" são consideradas equivalentes, pois todas possuem a mesma estrutura de repetição de letras; já as strings "aa" e "ab" são diferentes.

Input

A entrada consiste de um único inteiro n ($1 \leq n \leq 10^6$), representando o tamanho das strings a serem consideradas.

Output

A saída deve ser um único inteiro, representando o número de strings diferentes de tamanho n que podem ser formadas, levando em conta a simetria de permutação de letras. Como a resposta pode ser enorme, encontre-a módulo $(10^9 + 7)$.

Examples

standard input	standard output
3	5
1	1
5	52
1000000	281893449

Note

Para $n = 3$:

As strings “aaa”, “aab”, “aba”, “abb” e “abc” são exemplos das strings distintas que podem ser formadas.

Problem F. Fibonacci

Input file: `standard input`
Output file: `standard output`
Time limit: 0.25 seconds
Memory limit: 256 megabytes

A sequência de Fibonacci (f_n) é definida da seguinte forma:

$$f_n = \begin{cases} 0 & n = 0, \\ 1 & n = 1, \\ f_{n-1} + f_{n-2} & n \geq 2. \end{cases}$$

Como exemplo, os 6 primeiros termos dessa sequência são 0, 1, 1, 2, 3 e 5, e $f_9 = 34$.

Dados dois inteiros a e b ($1 \leq a, b \leq 10^9 + 9$) e um inteiro M ($1 \leq M \leq 10^9 + 9$), calcule o resto da divisão de f_{a^b} por M . Ou seja, encontre o único inteiro A tal que $f_{a^b} - A$ é um múltiplo de M e $0 \leq A < M$.

Input

A primeira e única linha de entrada consiste em três inteiros a, b e M , satisfazendo $1 \leq a, b, M \leq 10^9 + 9$.

Output

Imprima um único inteiro: o resto da divisão de f_{a^b} por M .

Examples

standard input	standard output
3 2 100	34
2 3 6	3
314159265 271828182 987654321	28908313

Note

No segundo caso de teste, note que $f_{2^3} = f_8 = 21$. A resposta é o resto da divisão de 21 por 6, que é 3.

Problem G. Grafo do Amor

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

Luiz Eduardo é um dos últimos remanescentes de uma raça quase extinta na Terra: os românticos. Ele acredita que todas as pessoas do mundo, uma hora ou outra, vão encontrar seu único amor verdadeiro, seja ele quem for (inclusive, seu amor verdadeiro pode ser até você mesmo!).

Ele tirou um ano sabático e viajou pelo mundo, conversando com pessoas e ouvindo sobre seus amores verdadeiros. Levando seu caderninho, ele anotou o nome das pessoas que conversou ou ouviu falar e o nome da pessoa amada correspondente. Depois de sua longa viagem, ele mostrou suas anotações para sua colega Iasmim (para proteger a privacidade das pessoas, ele trocou os nomes por números).

Iasmim não achou as anotações de Luiz muito interessantes, então decidiu transformá-las em algo mais legal: um grafo! Mais especificamente, ela construiu um grafo onde há uma aresta direcionada de A para B se e somente se a pessoa A ama verdadeiramente a pessoa B .

Ela então desafiou Luiz com um problema: qual é a maior quantidade de casamentos monogâmicos parcialmente felizes (C.M.P.F.s) que poderiam acontecer ao mesmo tempo nesse grafo? Um C.M.P.F. é composto de duas pessoas, A e B , tais que:

- $A \neq B$
- Existe no grafo de Iasmim uma aresta de A para B ou de B para A .
- Nem A nem B estão em outro C.M.P.F.

Tanto Luiz quanto Iasmim têm péssimas lembranças da última vez que ouviram as palavras casamento e grafo na mesma frase. Por isso, ajude-os a resolver esse problema.

Input

A primeira linha da entrada contém o número N , a quantidade de pessoas nas anotações de Luiz ($1 \leq N \leq 10^5$).

A segunda linha da entrada contém N inteiros A_i , cada um representando o amor verdadeiro da pessoa i ($1 \leq i, A_i \leq N$, é possível que $A_i = i$).

Output

Imprima um único inteiro: a maior quantidade de casamentos monogâmicos parcialmente felizes (C.M.P.F.s) que poderiam acontecer ao mesmo tempo no grafo de Iasmim.

Example

standard input	standard output
12 4 1 5 1 9 7 9 12 9 11 12 10	5

Note

No caso de teste de exemplo, para gerar a maior quantidade de C.M.P.F.s, uma de múltiplas possibilidades é juntar as pessoas nos seguintes casamentos: $(1, 4)$, $(3, 5)$, $(6, 7)$, $(8, 12)$, $(10, 11)$.

Apesar das pessoas 2 e 9 ficarem fora de C.M.P.F.s, como não existe aresta entre elas, não é possível juntá-las. Pode-se verificar que não é possível gerar mais do que 5 C.M.P.F.s ao mesmo tempo.

Problem H. Hora do Impacto

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

Pedro Pacoca gosta de jogar um jogo online de carros e, recentemente, descobriu que seu colega Pedro Gallo também joga esse jogo. Assim, os Pedros decidiram se ajudar.

No jogo, cada jogador dirige um carro em um circuito, e vence aquele que completar o percurso no menor tempo. Para isso, é importante evitar colisões entre os carros.

Sabendo disso, os dois Pedros decidiram compartilhar entre si a posição inicial de seus carros, suas velocidades e as direções que estavam se movendo. Eles também combinaram que manteriam constantes tanto a velocidade quanto a direção.

Agora, eles precisam da sua ajuda para determinar se haverá uma colisão entre os dois carros. Dadas as posições iniciais e as velocidades nas direções x e y de cada carro, responda o instante de tempo em que ocorrerá uma colisão. Caso não haja colisão, responda -1 .

Neste problema, para fins de detecção de colisão, cada carro é considerado como um triângulo no plano.

Input

A entrada consiste em quatro linhas. As duas primeiras linhas descrevem o primeiro carro e as duas últimas linhas descrevem o segundo carro.

Cada carro é descrito por duas linhas:

- A primeira linha contém seis inteiros x_1, y_1, x_2, y_2, x_3 e y_3 ($-10^5 \leq x_i, y_i \leq 10^5$). (x_1, y_1) , (x_2, y_2) e (x_3, y_3) são as coordenadas dos vértices do triângulo de colisão do carro.
- A segunda linha contém dois inteiros v_x e v_y ($-10^5 \leq v_x, v_y \leq 10^5$). (v_x, v_y) é a velocidade do carro nas direções x e y .

É garantido que os vértices fornecidos formam um triângulo não degenerado. Além disso, também é garantido que triângulos de colisão não se intersectam, isto é, não existe um ponto p que pertença simultaneamente aos dois triângulos.

Output

Imprima uma única linha contendo um número real: o instante de tempo em que ocorrerá uma colisão entre os carros ou -1 , caso não haja colisão.

A sua resposta será considerada correta se o erro absoluto ou relativo não exceder 10^{-6} .

Examples

standard input	standard output
0 0 1 0 0 1 1 0 1 1 2 1 1 2 0 -1	0.5
0 0 1 0 0 1 1 0 5 0 5 -1 4 0 -3 0	0.75
2 2 2 0 7 3 4 2 3 3 8 7 10 8 1 0	-1

Problem I. Inventando Nomes

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

Quirino é um aspirante a professor Pokémon. Ele tem especialização em Taxonomia, e uma de suas responsabilidades como assistente do renomado Professor Rowan é decidir os nomes dos novos Pokémon que são descobertos na região de Sinnoh.

Não se pode dar nomes iguais a Pokémon diferentes, então Quirino recebeu uma lista dos últimos N nomes dados a Pokémon para evitar repetições. Além disso, por restrições no sistema de nomenclatura, nomes devem ter no máximo K caracteres.

Quirino é metódico, e deseja escolher o lexicograficamente menor nome possível para todo Pokémon que ele for nomear. Porém, desde que ele subiu o Mount Coronet para desvendar os segredos do Trio da Criação, nunca mais foi visto. Em nome de Arceus, ajude-o a escrever um programa que nomeie novos Pokémon de acordo com suas restrições e preferências enquanto ele está fora!

Nota: uma palavra A é lexicograficamente menor que outra B se ela viria antes no dicionário. Ou seja, se $A \neq B$ e uma das duas vale:

1. A é um prefixo de B .
2. Na primeira posição i (da esquerda para a direita) onde elas diferem, $A_i < B_i$ (sendo P_i o i -ésimo caractere de P da esquerda para a direita).

Input

A primeira linha da entrada contém os números N e K : o número de nomes na lista e o maior tamanho para um nome ($1 \leq N, K \leq 2 \cdot 10^5$).

As próximas N linhas contém cada uma um dos nomes da lista: uma palavra composta de caracteres de a a z tendo entre 1 e K caracteres.

Restrições adicionais:

- Não há dois nomes iguais na lista.
- A quantidade total de caracteres nos nomes da lista não excede $2 \cdot 10^5$.
- Existe algum nome (com entre 1 e k caracteres de a a z) que não está na lista.

Output

Imprima, em uma única linha, o lexicograficamente menor nome (com entre 1 e k caracteres de a a z) que não está na lista dada no input.

Examples

standard input	standard output
2 10 torterra pikachu	a
2 2 a aa	ab

Problem J. Juntando Xegos

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

Tiago DFS está indo para a China!!

E, aproveitando a viagem, ele decide ir para a maior loja ancestral da cidade para comprar as peças que faltam em sua coleção de xegos.

O setor de xegos pode ser representado por uma imensa prateleira com N xegos disponíveis, em posições numeradas de 1 a N , para comprar.



Mas o dono chinês da loja é bem excêntrico e, todo dia, ele escolhe um intervalo contínuo $[L, R]$ e permite que os clientes comprem apenas os itens desse intervalo. Além disso, ele permite somente que o cliente entre uma única vez por dia na loja, então sempre escolha bem quais itens você irá comprar antes de sair da loja.

Depois do expediente, o próprio dono sempre repõe as peças que foram compradas no dia com os mesmos itens.

Ademais, cada peça de xego tem um identificador, que pode ser representado por um número inteiro único positivo a_i .

E todo mundo sabe que se juntarmos xegos, podemos formar outros xegos, os quais podem também, posteriormente, ser combinados em outros xegos.

Após brincar bastante com xegos, Tiago notou uma curiosa propriedade: a combinação de dois xegos A e B em um outro xego C pode ser exatamente descrita como a operação de XOR dos seus identificadores. Ou seja, $A \oplus B = C$.

Sabendo que ele terá Q dias livres para comprar xegos, ajude Tiago a responder, para cada dia, quantas combinações distintas de posições da prateleira que ele poderá comprar para formar o xego X , sabendo que naquele dia o dono permitiu a venda apenas do intervalo contínuo $[L, R]$.

Input

A primeira linha contém um inteiro positivo N ($1 \leq N \leq 3 \cdot 10^5$) que indica a quantidade de xegos na prateleira.

A segunda linha contém N inteiros positivos a_i ($1 \leq a_i \leq 10^{18}$), que representam os xegos que estão dispostos nas prateleiras.

A terceira linha contém um inteiro positivo Q ($1 \leq Q \leq 3 \cdot 10^5$) que representa a quantidade de dias que Tiago irá fazer compras.

Depois, nas próximas Q linhas, serão dados 3 inteiros positivos L, R ($1 \leq L \leq R \leq N$), e X ($1 \leq X_i \leq 10^{18}$), representando os limites do intervalo e o xego do dia.

Output

Para cada dia Q , responda de quantas formas Tiago pode formar o xego X , sabendo que apenas o intervalo contínuo $[L, R]$ está disponível para compra. Sabendo que o valor pode ser bem alto, imprima a resposta como o resto da divisão desse valor por 998244353.

Examples

standard input	standard output
5 1 2 4 8 16 3 1 3 7 1 5 19 4 4 16	1 1 0
4 2 1 1 2 3 1 3 3 1 4 2 1 4 4	2 4 0

Note

Caso Teste 1:

Para o dia 1, Tiago pode escolher todos os xegos no intervalo $[1, 3]$, obtendo $1 \oplus 2 \oplus 4 = 7$.

Para o dia 2, Tiago pode escolher os xegos nas posições #1, #2 e #5, obtendo $1 \oplus 2 \oplus 16 = 19$.

Para o dia 3, Tiago pode escolher apenas o item na posição #4, que é do tipo 8, não conseguindo gerar 16.

Caso Teste 2:

Para o dia 1, há 2 combinações válidas #1, #2 e #1, #3.

Para o dia 2, há 4 combinações válidas #1, #4, #1, #2, #3 e #2, #3, #4.

Para o dia 3, não há combinações válidas.

Problem K. Koga precisa de sua ajuda

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

O grande treinador Pokémon Koga decidiu que seu time com Pokémon de tipo venenoso precisa de mais variedade. Para isso, Koga precisa entender um pouco mais sobre os tipos de Pokémon. Koga tem 3 Pokémon diferentes: Torterra, Staraptor e Luxray. Ele vai enfrentar um desses 3 Pokémon e você deve ajudá-lo falando qual Pokémon Koga deve utilizar.

Torterra tem vantagem contra Luxray, Luxray tem vantagem contra Staraptor e Staraptor tem vantagem contra Torterra.

Input

É dada uma linha no formato: "`Vamos la, S!`"(Sem aspas). *S* pode ser Torterra, Staraptor ou Luxray.

Output

Imprima uma linha no formato: "`T, eu escolho voce!`"(Sem aspas). *T* pode ser Torterra, Staraptor ou Luxray.

Example

<code>standard input</code>	<code>standard output</code>
<code>Vamos la, Torterra!</code>	<code>Staraptor, eu escolho voce!</code>

Note

Staraptor tem vantagem contra Torterra, então se *S* for Torterra, *T* deve ser Staraptor.

Problem L. Legado de Lagrange

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

Lórien, o último dos magos do Reino de Lagrange, encontrou uma fórmula secreta esculpida nas paredes de uma antiga torre. Ela representa o valor absoluto de um poderoso polinômio que, segundo as lendas, possui propriedades mágicas se aplicado corretamente. A fórmula é:

$$f(x) = |(A + C)x^3 - Bx^2 + BCx + (x + C)(x - A)|.$$

Para proteger esse conhecimento, Lórien precisa aplicar a função a vários valores e revelar apenas o resultado final de um feitiço de disfarce: o XOR ($\oplus$) de todos os resultados das avaliações $f(x)$ nesses valores.

O XOR entre dois números a e b é um outro número, vamos chamá-lo de c . Para calcular $a \oplus b = c$, escreva as representações em binário de a e b . Cada bit (dígito da representação binária) de c será 1 se os bits correspondentes em a e b forem diferentes, e 0 caso contrário.

Sua missão é ajudá-lo a calcular esse valor final.

Input

A primeira linha contém três inteiros A, B e C ($-100 \leq A, B, C \leq 100$).

A segunda linha contém um inteiro Q ($2 \leq Q \leq 1000$), o número de valores.

A terceira linha contém Q inteiros $X_1, X_2, \dots, X_Q$ ($-100 \leq X_i \leq 100$), os valores para os quais a função f deve ser avaliada.

Output

Imprima um único inteiro: o XOR de todos os valores $f(X_i)$, para $1 \leq i \leq Q$.

Examples

standard input	standard output
1 2 3 3 0 1 2	34
13 100 -17 4 1 -10 5 -5	7509

Note

No primeiro caso de teste, temos $f(0) = |-3| = 3$, $f(1) = |8| = 8$ e $f(2) = |41| = 41$. As representações em binário de $f(0)$, $f(1)$ e $f(2)$, respectivamente, são as seguintes:

000011

001000

101001

. Assim, o XOR desses valores em binário é 100010, que é 34 em decimal.

Problem M. Menor Menor Caminho

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

Após uma expedição bem sucedida para uma maratona da UFG, os competidores AlbertoTDNeto, duduFreire, EYZ, henriquemos.js e MagePetrus fizeram um pacto: toda vez que os 5 se encontrarem, eles devem rolar um D100 (dado de 100 lados). Se o número rolado for 100, eles devem imediatamente ir para Goiânia.

Depois de muitos encontros, isso finalmente aconteceu e os competidores irão manter suas promessas. Todos entram no carro de Henrique, que os conduzirá até Goiânia. Os maratonistas querem percorrer o menor caminho possível de Brasília até Goiânia, é claro. Além disso, como o carro está bastante pesado, Henrique ficou preocupado com a possibilidade do combustível acabar no meio da estrada. Então, dentre os caminhos mais curtos, ele quer um caminho que minimize a maior distância que o carro irá percorrer entre dois postos de gasolina.

Os postos de gasolina estão enumerados de 1 até N . Henrique partirá de Brasília a partir do posto número 1 e finalizará sua jornada no posto número N , em Goiânia. Existem M estradas conectando pares de postos distintos. Para cada i de 1 até M , a estrada de número i parte do posto a_i , leva ao posto b_i e tem um comprimento de w_i quilômetros. Só é possível percorrer essa estrada no sentido de a_i para b_i . É garantido que existe um caminho de 1 até N .

Ajude Henrique a encontrar um menor caminho que minimize a maior distância percorrida entre dois postos de gasolina consecutivos.

Input

A primeira linha contém dois inteiros $2 \leq N, M \leq 2 \cdot 10^5$: a quantidade de postos de gasolina e a quantidade de estradas ligando pares de postos distintos, respectivamente.

Em seguida, existem M linhas de entrada. A linha de número i contém três inteiros a_i, b_i e w_i , com $1 \leq a_i, b_i \leq N$ e $1 \leq w_i \leq 10^5$, representando uma estrada unidirecional do posto a_i para o posto b_i de w_i quilômetros de comprimento.

Output

Imprima dois inteiros, nesta ordem: o comprimento do menor caminho e o comprimento da maior aresta percorrida em um menor caminho que minimiza a maior aresta percorrida.

Examples

standard input	standard output
5 6 1 2 1 1 3 10 1 4 1 2 3 3 3 5 2 4 5 5	6 3
2 2 1 2 1 1 2 2	1 1

Problem N. Na zdrowie

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

Frank e seus amigos estão jogando um jogo no postinho. O jogo possui uma lista de n línguas distintas $a_1, a_2, \dots, a_n$. Cada língua é representada por um número de 1 a n e só aparece uma vez. Frank não confia nas habilidades de programação competitiva dos brasileiros, então ele quer que você responda q consultas, cada uma contendo dois inteiros l e r . Responda para Frank se a sublista $a_l, a_{l+1}, \dots, a_r$ é um **bom brinde**. Como Frank só fala polonês, responda “TAK” se a lista é um **bom brinde** e “NIE” caso contrário.

Uma lista de tamanho m é **bom brinde** se ela é uma permutação, ou seja, possui todos os números de 1 até m exatamente uma vez.

Input

A primeira linha contém dois inteiros n e q ($1 \leq n, q \leq 10^5$), o tamanho da lista e o número de consultas, respectivamente.

A segunda linha contém n inteiros $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$), representando as línguas da lista. É garantido que a lista é uma permutação.

As próximas q linhas contêm dois inteiros l e r ($1 \leq l \leq r \leq n$) cada, representando as consultas.

Output

Para cada consulta, imprima uma linha com a palavra “TAK”, se a sublista $a_l, a_{l+1}, \dots, a_r$ forma uma permutação, ou “NIE”, caso contrário.

Examples

standard input	standard output
4 3 1 2 3 4 1 2 2 3 1 4	TAK NIE TAK
5 4 4 1 2 3 5 2 3 1 3 2 4 1 4	TAK NIE TAK TAK
1 1 1 1 1	TAK