

Problem A. Aprimoramentos Divertidos

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

Leon acabou de terminar sua palestra e agora quer tirar uma foto dos participantes para suas redes sociais. A foto consiste em uma fileira de pessoas dispostas em uma linha horizontal. Ryei foi escolhido para montar uma foto com um nível de diversão total de exatamente D , onde cada pessoa adicionada à foto aumenta o nível de diversão em 1.

Ryei inicialmente planejou apenas enfileirar D pessoas normais para atingir o nível de diversão necessário, mas Leon queria mais variedade. Ele decidiu que, ao montar a foto, qualquer pessoa adicionada pode ser *aprimorada* usando IA, fazendo com que ela contribua com 2 para o nível total de diversão em vez de 1.

Leon agora está interessado no número de configurações diferentes de *aprimoramento* para fotos que tenham um nível de diversão de exatamente D com qualquer quantidade de pessoas, considerando que as pessoas são indistinguíveis; ou seja, o número de ordens possíveis de participantes normais e *aprimorados* que resultam em uma foto com nível de diversão D . Como esse número pode ser muito grande, Leon quer a resposta módulo 2026, porque ele é uma entidade moderna.



Aprimoramento por IA de Leon

Como Ryei não consegue parar de rir sempre que vê uma foto *aprimorada*, ele precisa da sua ajuda para escrever um programa que conte o que Leon deseja.

Input

O input consiste em um inteiro D ($1 \leq D \leq 10^6$), o número da diversão total requerida por Leon.

Output

Seu programa deve imprimir um número inteiro em uma única linha: o número de configurações de *aprimoramentos* que geram fotos de diversão exatamente D , módulo 2026.

Examples

standard input	standard output
1	1
4	5
2026	1

Note

Para o segundo exemplo ($D = 4$), existem 5 possíveis configurações:

- $1 + 1 + 1 + 1$
- $2 + 1 + 1$
- $1 + 2 + 1$
- $1 + 1 + 2$
- $2 + 2$

Problem B. Balatro

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

Drex precisava de uma pausa para descansar de implementar estruturas de dados, então decidiu relaxar com algumas rodadas de Balatro. Ele está em uma grande sequência. Para vencer o jogo, ele deve derrotar um chefe poderoso que pode revidar e causar dano a ele. Tanto Drex quanto o chefe têm mãos compostas por n inteiros. Drex pode reorganizar a ordem dos inteiros em sua mão antes do início da luta.

O dano total que Drex recebe é calculado como: $\sum_{i=1}^n |d_i - b_i|$, onde d_i representa o i -ésimo inteiro na mão de Drex (após reorganizar) e b_i representa o i -ésimo inteiro na mão do chefe.

Drex quer reorganizar sua mão de forma a minimizar o dano total que recebe.

Input

A primeira linha contém um inteiro n ($1 \leq n \leq 2 \cdot 10^5$).

A segunda linha contém n inteiros $d_1, d_2, \dots, d_n$ ($-10^9 \leq d_i \leq 10^9$), a mão de Drex.

A terceira linha contém n inteiros $b_1, b_2, \dots, b_n$ ($-10^9 \leq b_i \leq 10^9$), a mão do chefe.

Output

Imprima um único inteiro, o dano total mínimo que Drex receberá do chefe.

Example

standard input	standard output
3 1 2 3 3 1 2	0

Note

Se Drex reorganizar sua mão para $[3, 1, 2]$, ele receberá $|3 - 3| + |1 - 1| + |2 - 2| = 0$ de dano.

Problem C. Cálculos do Coach

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

O fim de uma era chegou. Com o encerramento do lendário legado de Leo, a missão de montar as equipes da MaratonaCIn para a Subregional caiu no colo do novo coach, Palão.

Mas a paz no laboratório durou pouco. Eol, o eterno archi-inimigo de Leo, sentiu que a vida estava pacata demais sem seu velho rival e decidiu implicar com o sucessor. Durante a madrugada, ele invadiu os servidores da seletiva e sabotou o sistema de avaliação.

O sistema de Palão mantinha, para cada competidor, uma métrica representada por um polinômio $R(x)$. Para mascará-la, Eol quebrou a lógica em dois polinômios distintos, $P(x)$ e $Q(x)$, de modo que a métrica original só pode ser recuperada calculando-se a composição $R(x) = P(Q(x))$. Em outras palavras, se $P(x) = p_0 + p_1x + p_2x^2 + \dots + p_nx^n$, então

$$R(x) = P(Q(x)) = p_0 + p_1 Q(x) + p_2 Q(x)^2 + \dots + p_n Q(x)^n$$

Eol sabia que calcular essa composição de forma ingênua geraria coeficientes astronômicos, fritando os servidores. Para que a sabotagem funcionasse sem destruir tudo de vez (afinal, ele quer se divertir, não acabar com a festa), forçou todas as operações a serem realizadas módulo um primo específico. E, sendo o vilão convencido que é, não escolheu um primo qualquer: configurou o sistema para usar exatamente o seu próprio número de WhatsApp, 998244353.

Ajude Palão a salvar a seletiva. Dados os polinômios $P(x)$ e $Q(x)$, calcule os coeficientes de $R(x) = P(Q(x)) \pmod{998244353}$

Input

A primeira linha contém dois inteiros n e m ($1 \leq n, m \leq 100$): os graus dos polinômios P e Q , respectivamente.

A segunda linha contém $n + 1$ inteiros $p_0, p_1, \dots, p_n$ ($0 \leq p_i < 998244353$): os coeficientes de $P(x) = p_0 + p_1x + \dots + p_nx^n$.

A terceira linha contém $m + 1$ inteiros $q_0, q_1, \dots, q_m$ ($0 \leq q_i < 998244353$): os coeficientes de $Q(x) = q_0 + q_1x + \dots + q_mx^m$.

É garantido que $p_n \neq 0$ e $q_m \neq 0$ (i.e., os polinômios têm exatamente os graus indicados).

Output

Imprima $n \cdot m + 1$ inteiros: os coeficientes $r_0, r_1, \dots, r_{n \cdot m}$ de $R(x) = P(Q(x)) \pmod{998244353}$, separados por espaço.

Examples

standard input	standard output
2 1 1 2 1 1 1	4 4 1
1 2 0 1 1 0 1	1 0 1

Note

No primeiro sample, $P(x) = 1 + 2x + x^2$ e $Q(x) = 1 + x$

A composição $R(x) = P(Q(x))$ é calculada substituindo x por $(1 + x)$:

$$R(x) = 1 + 2(1 + x) + (1 + x)^2 = 1 + 2 + 2x + 1 + 2x + x^2 = 4 + 4x + x^2$$

Os coeficientes resultantes são 4 4 1.

No segundo sample, $P(x) = x$ e $Q(x) = 1 + x^2$.

$$R(x) = P(Q(x)) = Q(x) = 1 + x^2$$

Os coeficientes resultantes são 1 0 1.

Problem D. Duelo dos Dígitos

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

GabMei e TFG são duas lendas da MaratonaCIn, ambos finalistas mundiais e agora coaches da próxima geração. Em algum momento, ganharam os apelidos de “mastermei” e “the greatest” — ninguém lembra quem inventou, mas os nomes pegaram.

Para decidir quem realmente merece o título mais grandioso, eles inventaram um jogo. No início de uma partida, há N números sobre a mesa, correspondentes aos valores $2, 3, 4, \dots, N + 1$. Os jogadores se alternam, e em cada turno, o jogador da vez deve escolher um subconjunto não vazio dos números restantes e removê-lo. A remoção só é válida se o MDC de todos os números escolhidos for estritamente maior que 1. Se um jogador não conseguir fazer uma jogada válida, ele perde.

Sempre que um subconjunto de X números é removido, os números restantes são renomeados para serem os menores valores consecutivos a partir de 2. Ou seja, se havia Y números antes da jogada, os números após a remoção passam a ser $2, 3, 4, \dots, Y + 1 - X$.

Mastermei sempre joga primeiro. Assumindo que ambos jogam de forma ótima, determine quem vence.

Input

A primeira linha contém um único inteiro T ($1 \leq T \leq 100$), o número de casos de teste.

Cada caso de teste consiste em uma única linha contendo um inteiro N ($1 \leq N \leq 5 \cdot 10^5$).

Output

Para cada caso de teste, imprima uma única linha contendo “mastermei” (sem aspas) se o primeiro jogador vence, ou “the greatest” (sem aspas) caso contrário.

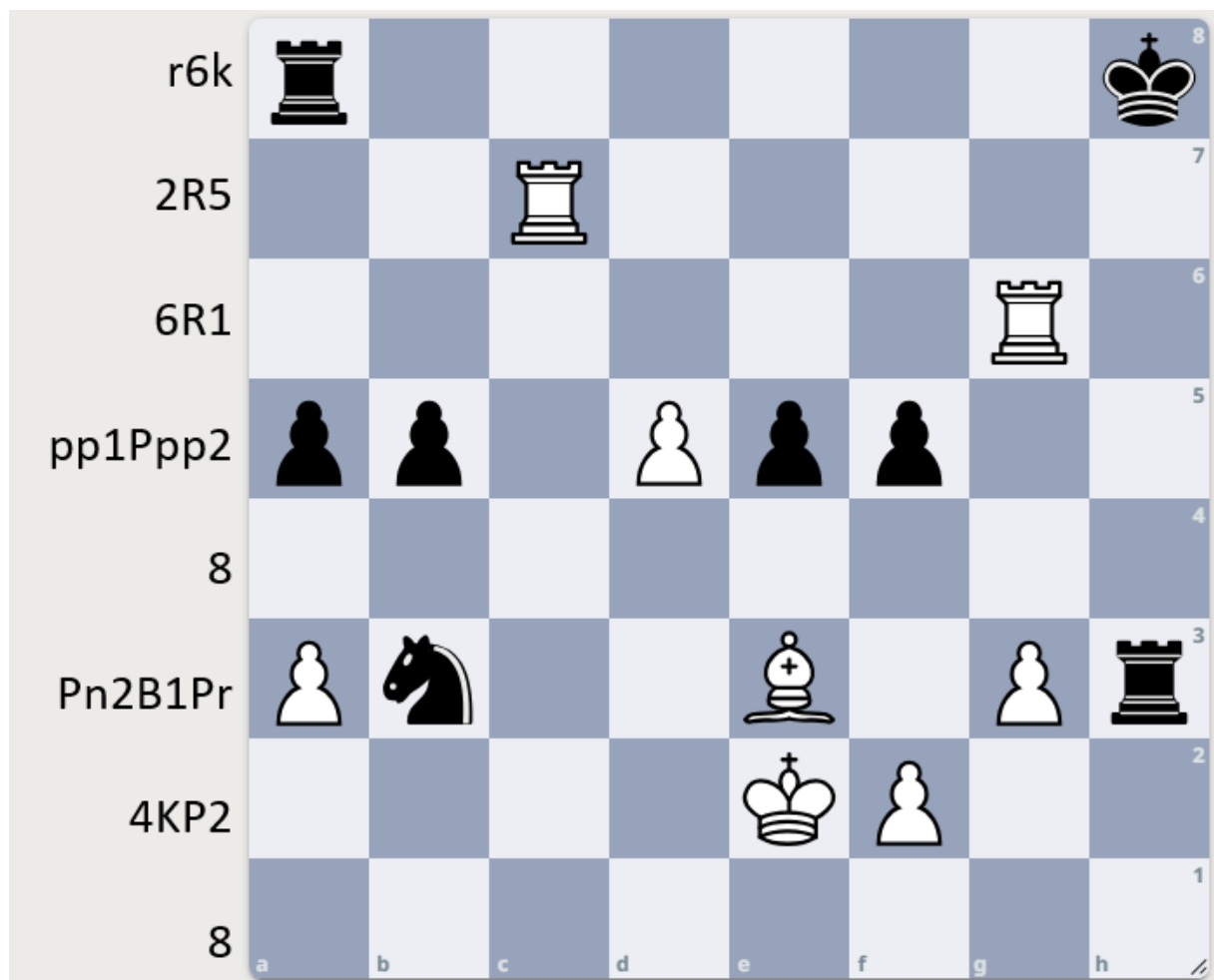
Examples

standard input	standard output
6	mastermei
1	the greatest
2	mastermei
3	mastermei
4	mastermei
5	the greatest
6	
6	mastermei
13	the greatest
62	mastermei
1000	mastermei
123456	the greatest
131070	the greatest
262142	

Problem E. Avaliação

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

Ftg, após perder várias partidas de xadrez para Yrei, decide desenvolver um motor de xadrez para auxiliá-lo em seus jogos. Para isso, o motor deve ser capaz de receber uma posição do tabuleiro e indicar o melhor lance a ser jogado. Após pesquisar, Ftg descobriu que a melhor forma de compartilhar posições de xadrez é utilizando o formato FEN (Forsyth-Edwards Notation), que funciona da seguinte maneira: o FEN descreve o tabuleiro da 8ª à 1ª fileira, usando letras maiúsculas para peças brancas e minúsculas para pretas (ex.: P = Peão Branco, n = Cavalo Preto). Casas vazias consecutivas são representadas por números (1 a 8), e cada fileira é separada por uma barra (/). A notação percorre da esquerda para a direita (coluna A até H), totalizando sempre 8 unidades por linha.



Note que tanto a notação FEN quanto as fileiras do tabuleiro são lidas do 8 ao 1.

Ftg tem ótimas habilidades de programação e poderia codificar isso rapidamente. Porém, Yrei acabou de desafiá-lo para uma partida, e ele precisará se concentrar totalmente nela. Você pode ajudar Ftg a criar um conversor de posições de xadrez para o formato FEN?

Input

A entrada consiste em exatamente 8 linhas, cada uma contendo exatamente 8 caracteres. Essas linhas representam o estado atual do tabuleiro de xadrez da 8ª à 1ª fileira.

Uma casa vazia é representada por um ponto (.). Peças brancas são representadas por letras maiúsculas (P, N, B, R, Q, K), e peças pretas são representadas por letras minúsculas (p, n, b, r, q, k).

Output

Imprima uma única string — a representação em FEN (Forsyth-Edwards Notation) da posição do tabuleiro fornecida, conforme descrito no enunciado.

Examples

standard input	standard output
rnbqkbnr pppppppp PPPPPPPP RNBQKBNR	rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR
r.bk...r p..pBpNp n....n.. .p.NP..PP. ...P.... P.P.K... q.....b.	r1bk3r/p2pBpNp/n4n2/1p1NP2P/6P1/3P4/P1P1K3/q5b1
R.....k ..R.....R. pp.Ppp.. Pn..B.PrKP..	R6k/2R5/6R1/pp1Ppp2/8/Pn2B1Pr/4KP2/8

Note

O primeiro exemplo mostra a posição inicial padrão de uma partida de xadrez. As duas primeiras fileiras são impressas exatamente como estão. As quatro fileiras seguintes consistem inteiramente em casas vazias, portanto cada uma é convertida para o número 8. As duas últimas fileiras são impressas exatamente como estão.

No segundo exemplo, a primeira fileira `r.bk...r` contém dois grupos de casas vazias: um ponto entre `r` e `b` torna-se 1, e três pontos entre `k` e `r` tornam-se 3, resultando em `r1bk3r`.

O terceiro exemplo é a imagem.

Problem F. Frações de um Graveto

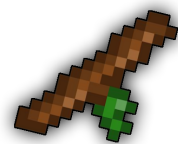
Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Toni estava caminhando pelo selvagem estacionamento do CIn, desviando de feras e tentando achar onde estacionou seu carro, quando tropeçou em um objeto formidável: um graveto de comprimento inteiro exato n .

Como todo bom problem-solver, a mente de Toni imediatamente pensou em uma "gambiarra". Ele precisa construir uma base de sustentação para seu novo projeto de hardware. Para que a base fique estável, ele precisa formar um triângulo perfeito usando pedaços desse graveto.

Toni, no entanto, está exausto após uma maratona de codificação. Ele decide cortar o graveto em dois pontos distintos, escolhidos de forma totalmente aleatória. Ele só tem energia para esses dois cortes.

Sua missão é calcular as chances de a gambiarra de Toni dar certo, para que ele saiba se deve ou não desistir e ir tomar um café.



Formalmente, temos um segmento de reta de comprimento n . O conjunto de pontos de corte possíveis são as coordenadas inteiras internas $\{1, 2, \dots, n-1\}$.

Toni escolhe dois pontos de corte distintos desse conjunto, uniformemente ao acaso. Isso divide o graveto original em três pedaços com comprimentos inteiros positivos a , b e c (tais que $a + b + c = n$).

Qual é a probabilidade de que esses três pedaços resultantes (a, b, c) possam formar um triângulo não degenerado (ou seja, um triângulo com área maior que zero)?

Como a resposta pode ser uma fração irredutível $\frac{p}{q}$, você deve imprimir o valor de $p \cdot q^{-1} \pmod{10^9 + 7}$, onde q^{-1} é o inverso modular de q módulo $10^9 + 7$.

Input

A primeira linha da entrada contém um único inteiro t ($1 \leq t \leq 10^5$) — o número de casos de teste. Cada uma das t linhas seguintes contém um único inteiro n ($3 \leq n \leq 10^9$), representando o comprimento do graveto que Toni encontrou naquele cenário.

Output

Para cada caso de teste, imprima um único inteiro em uma nova linha: a probabilidade da gambiarra funcionar, módulo $10^9 + 7$.

Example

standard input	standard output
5	1
3	0
4	500000004
5	800000006
7	636363641
12	

Problem G. Gabmei, o Hacker

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

Gab “gabmei” mei é conhecido por seus hacks em contests do Codeforces. Ele uma vez hackeou três maratonistas no mesmo contest!

gabmei	matheuslafayette 324872687, all	D - 1709	View test	Successful hacking attempt (wa)
gabmei	miguel 324899845, all	D - 1709	View test	Successful hacking attempt (wa)
gabmei	SamuellH12 324915997, all	D - 1709	View test	Successful hacking attempt (rt)

Triple kill!

Agora, ele está planejando um retorno. Desta vez, ele vem ao UFPE Starters Final Try-Outs 2026 e pretende “hackear” os balões dos competidores arremessando seus dardos supremos, comprados de Quincy Jr., filho de Quincy. Os computadores dos competidores podem ser representados como pontos inteiros no plano, e seus balões estão presos a esses pontos.

Seu objetivo é estourar todos os balões dos competidores. Porém, ele quer usar a menor quantidade de dardos possível. Gabmei pode arremessar um dardo de qualquer posição que escolher e, uma vez arremessado, o dardo continua infinitamente na direção em que foi lançado, estourando todos os balões em seu caminho. Ajude-o a encontrar a quantidade mínima de dardos que ele precisa comprar de Quincy Jr. para estourar todos os balões dos competidores.

Note que um dardo lançado por Gabmei funciona como uma reta infinita no plano, visto que ele pode arremessá-lo de qualquer lugar.

Input

A primeira linha contém um inteiro n ($1 \leq n \leq 20$), o número de balões.

As próximas n linhas contém um par de inteiros separados por um espaço, x_i e y_i , ($-10^6 \leq x_i, y_i \leq 10^6$), as coordenadas do i -ésimo balão.

É garantido que todas as coordenadas dos balões são distintas, ou seja, $(x_i, y_i) \neq (x_j, y_j)$ para todo $i \neq j$.

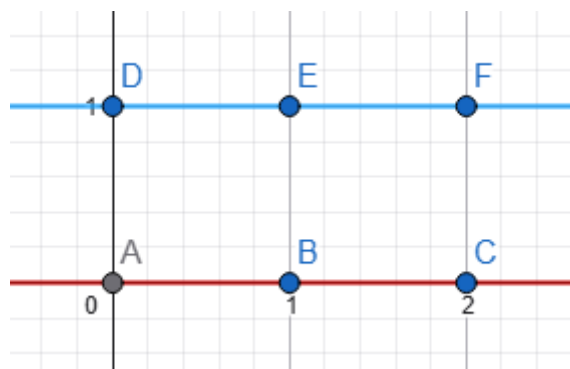
Output

Imprima uma única linha contendo um inteiro indicando a quantidade mínima de dardos necessária para estourar todos os balões.

Example

standard input	standard output
6 0 0 1 0 2 0 0 1 1 1 2 1	2

Note



Uma solução com 2 dardos para o primeiro exemplo

Problem H. Hora do Pou

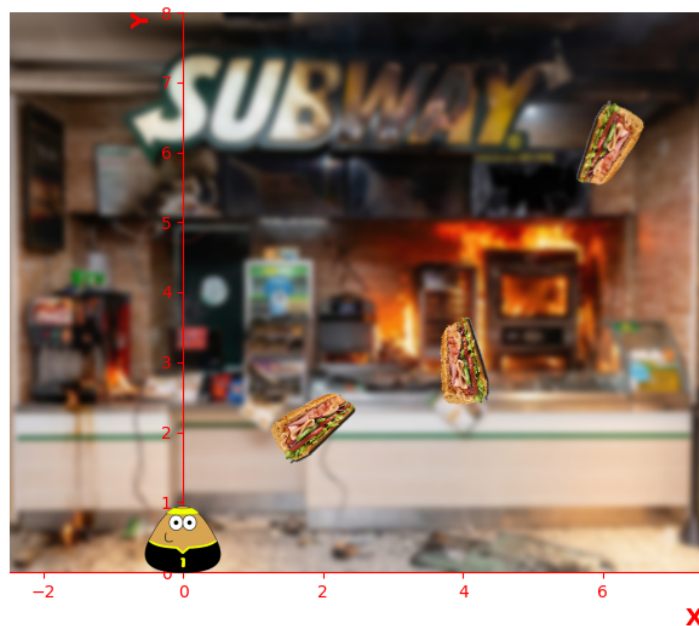
Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Pou, Grisi e Guilherme estavam com muita fome, então decidiram jantar no Subway. Enquanto esperavam pelos pedidos jogando Brawl Stars, a cozinha explodiu do nada, jogando comida para todo lado! Pou, sendo o único ágil o suficiente, decidiu aproveitar a situação para saciar sua fome. Ele fechou o jogo e partiu para o ataque!

Todos os itens de comida e Pou estão localizados em um plano bidimensional, onde $y = 0$ representa o chão. No instante $t = 0$, Pou está na posição $(0, 0)$ e o i -ésimo alimento está na posição (x_i, y_i) . Cada alimento cai a uma velocidade vertical constante de 1 unidade por segundo, enquanto Pou pode se mover para a esquerda ou para a direita a uma velocidade constante de 1 unidade por segundo.

Em outras palavras, se em um instante t um alimento está em (x, y) , no instante $t + 1$ ele estará em $(x, y - 1)$. Simultaneamente, se em um instante t Pou está em $(x, 0)$, no instante $t + 1$ ele pode estar em $(x, 0)$, $(x + 1, 0)$ ou $(x - 1, 0)$.

Pou pode comer o alimento i se e somente se estiver na posição x_i no exato momento em que o alimento toca o chão ($y = 0$). Por questões de higiene, ele não comerá nenhum alimento que já tenha tocado o chão antes de sua chegada.



Grisi e Guilherme ficaram tão atordoados com a explosão que esqueceram quantos alimentos Pou comeu, mas lembram das posições iniciais de todos eles. Sabendo que Pou comeu a quantidade máxima possível de alimentos, sua missão é determinar esse número.

Input

A entrada consiste em várias linhas. A primeira linha contém n ($1 \leq n \leq 2 \cdot 10^3$), o número de alimentos. As n linhas seguintes contêm dois inteiros x e y ($1 \leq x, y \leq 10^6$), a i -ésima linha representa a posição inicial do alimento i .

É garantido que não há dois alimentos na mesma posição inicial, ou seja:
 $\forall i \forall j ((i \neq j) \implies (x_i, y_i) \neq (x_j, y_j))$.

Output

A saída deve conter um único inteiro: a quantidade máxima de alimentos que Pou poderia ter comido.

Examples

standard input	standard output
3 2 2 4 3 6 6	2
3 2 3 2 4 727 67	2

Problem I. Infinitos Dinheiros

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

Tiago estava com seus amigos Leonardo, Gabriel e Paulo jogando o famoso jogo Banco Direcionado. Como todos sabem, o tabuleiro do Banco Direcionado possui N casas e M movimentos possíveis, sendo o i -ésimo movimento representado pela tupla (U_i, V_i, W_i) , isto é: se o jogador está na casa U_i , ele pode ir para a casa V_i somando W_i ao seu saldo. É possível que o saldo fique negativo durante o jogo.

O jogo sempre começa na casa de número 1 e, em uma rodada, o jogador pode realizar um movimento válido ou esperar na mesma casa, sem perder ou ganhar nada.

Sendo um exímio jogador de Banco Direcionado, Tiago logo percebeu um ~~bug~~ feature no jogo! É possível acumular dinheiro infinito! Mas não estava tão claro como isso poderia acontecer, e logo Leonardo e Gabriel discordaram da afirmação, exigindo uma prova. Tiago disse que demonstraria com um exemplo de infinitas jogadas durante a partida, o que levaria um tempo infinitamente grande também.

Mas como a pizza que eles pediram já estava quase chegando, só houve tempo suficiente para Tiago jogar um número finito de rodadas, e o problema ficou em aberto.

Sabendo que você é um(a) grande programador(a) e que está treinando para a Maratona de Programação da UFPE, Paulo lhe propôs um desafio: provar ou refutar Tiago!

Mas como infinito é muito tempo, você só precisa simular $10^{998244353}$ rodadas para convencer o grupo de amigos. Isto é, dada a descrição do tabuleiro, diga qual o **maior valor médio por rodada** possível de se obter neste tabuleiro após $10^{998244353}$ rodadas.

Input

A primeira linha da entrada contém dois inteiros $2 \leq N \leq 2000$, $1 \leq M \leq 2000$, representando o número de casas no tabuleiro e o número de movimentos válidos possíveis.

As M linhas seguintes contêm três inteiros U_i, V_i, W_i , com $1 \leq U_i, V_i \leq N$ e $-10^9 \leq W_i \leq 10^9$, sendo $U_i \neq V_i$. Representando a lista de movimentos válidos.

Note que múltiplas arestas entre o mesmo par de posições são permitidas.

Output

Imprima um número real, a resposta do problema. Sua solução será considerada correta caso apresente erro relativo ou absoluto menor ou igual a 10^{-6} .

Examples

standard input	standard output
2 2 1 2 2 2 1 -1	0.5
3 3 1 2 -5 2 3 5 3 1 -5	0

Note

No **exemplo 1**, uma estratégia válida é ir da casa 1 para a casa 2, voltar para a casa 1, ir para a 2 em seguida e assim por diante $(1, 2, 1, 2, 1, 2, \dots)$. Após infinitas rodadas, a média de pontos acumulados por rodada será 0.5.

No exemplo 2, uma estratégia válida é nunca sair da primeira casa, logo nunca perdendo pontos de saldo.

Problem J. Jornada por Uvas

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

Na casa de praia de Miguel, existe um quarto mágico: quem resolver o enigma do quarto tem um desejo realizado. Percebendo que seu amigo Walfrido estava triste, Thawan decidiu entrar no quarto para animá-lo. Seu objetivo é pedir uma generosa porção de uvas com margarina, o lanche favorito de Walfrido!

No entanto, o quarto só realizará o desejo se Thawan conseguir resolver o seguinte problema:

Considere um polígono regular com N vértices, numerados de 0 a $N-1$ no sentido horário. Thawan começa sua jornada no vértice 0. Em cada movimento, ele dá um salto de tamanho S , avançando exatamente S vértices no sentido horário. Ou seja, se ele está atualmente no vértice x , o salto o levará ao vértice $(x + S) \bmod N$.

Sabendo que Thawan pode continuar saltando indefinidamente, determine o número máximo de vértices distintos que ele conseguirá visitar ao longo do caminho. Ajude Thawan a garantir o lanche de Walfrido!

Input

A única linha da entrada contém dois inteiros, N e S ($1 \leq S \leq N \leq 10^9$), representando o número de vértices do polígono e o tamanho do salto de Thawan, respectivamente.

Output

Imprima um único inteiro representando o número de vértices distintos que Thawan consegue alcançar.

Examples

<code>standard input</code>	<code>standard output</code>
2 1	2
14 8	7

Problem K. K-Lugares

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 256 megabytes

Após se classificarem para o Mundial do ICPC, os N membros da MaratonaCIn decidiram comemorar com uma viagem juntos. Eles encontraram K pontos turísticos, convenientemente numerados de 1 a K . Há, porém, um problema: maratonistas são notoriamente exigentes. O i -ésimo membro tem m_i lugares que se recusa a visitar.

Querendo deixar todos felizes, eles estão dispostos a se dividir em grupos, desde que cada grupo seja um intervalo contíguo da lista original. Para ajudar o Coach Palão a planejar a viagem, responda Q consultas: quais lugares podem ser visitados se ele levar os membros de um dado intervalo $[l, r]$?

Input

A primeira linha contém dois inteiros N ($1 \leq N \leq 10^5$) e K ($1 \leq K \leq 60$).

Em seguida, N linhas. A i -ésima linha começa com um inteiro m_i ($0 \leq m_i \leq K$), a quantidade de lugares que a i -ésima pessoa se recusa a ir. Depois, m_i inteiros p_j ($1 \leq p_j \leq K$), os números dos lugares.

Em seguida, um inteiro Q ($1 \leq Q \leq 2 \cdot 10^5$), o número de consultas.

Por fim, Q linhas contendo dois inteiros l r , ($1 \leq l \leq r \leq N$).

Output

Para cada consulta, imprima os lugares que o grupo formado por todos os maratonistas de l a r pode visitar, **em ordem crescente**. Se não houver nenhum lugar disponível, imprima -1 .

Example

standard input	standard output
5 6	3
1 1	4 6
3 2 4 6	1 3 4 6
2 2 5	-1
0	
2 1 3	
4	
1 3	
3 5	
3 4	
1 5	

Note

Note que a saída deve estar em ordem estritamente crescente; caso contrário, sua resposta pode não ser aceita.

Problem L. Letras Encadeadas

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

Lua é de Vitória da Conquista, uma cidadezinha do interior da Bahia, ou pelo menos é o que ela diz. Ryei nunca engoliu essa história: a cidade tem mais de três mil quilômetros quadrados de área, aproximadamente dezessete vezes a área de Recife. “Pequena” e “interior” não são exatamente as palavras que vêm à mente.

Durante uma tarde particularmente parada no laboratório, Lua inventou uma teoria linguística: duas palavras estão conectadas se é possível transformar uma na outra através de uma cadeia de mutações, onde cada mutação troca uma palavra por outra que compartilhe pelo menos uma letra em comum com ela. Ela estava convicta de que isso resolveria o debate: se conseguisse mostrar que “CONQUISTA” alcança “INTERIOR” através de uma cadeia de mutações sobre um dado dicionário, a ligação entre sua cidade e o interior estaria matematicamente provada.

Ryei, naturalmente, quer refutar toda conexão que ela encontrar. A discussão escalou rápido, agora o laboratório inteiro está envolvido e todos continuam submetendo novas consultas para testar.

Dado um dicionário de N palavras e Q consultas, determine para cada consulta se a palavra a consegue alcançar a palavra b através de uma ou mais mutações.

Input

A primeira linha contém dois inteiros N e Q ($2 \leq N \leq 2 \cdot 10^5$, $1 \leq Q \leq 2 \cdot 10^5$): o tamanho do dicionário e o número de consultas.

Cada uma das próximas N linhas contém uma string s_i ($1 \leq |s_i| \leq 50$) composta por letras minúsculas do alfabeto inglês. Todas as palavras são distintas.

Cada uma das próximas Q linhas contém duas strings a e b , separadas por um espaço: uma consulta perguntando se a palavra a consegue alcançar a palavra b através de uma ou mais mutações. É garantido que ambas a e b aparecem no dicionário e que $a \neq b$.

Output

Para cada consulta, imprima uma linha contendo “LUA” (sem aspas) se a palavra a pode ser transformada na palavra b através de uma cadeia de uma ou mais mutações, ou “RYEI” (sem aspas) caso contrário.

Examples

standard input	standard output
6 4 conquista interior recife capital lua ryei conquista interior recife capital conquista recife lua ryei	LUA LUA LUA LUA
5 5 lua palao ryei grisi recife lua palao ryei grisi lua ryei palao recife grisi palao	LUA LUA RYEI RYEI RYEI
5 4 ab bc cd da xyz ab cd ab xyz da xyz ab da	LUA RYEI RYEI LUA

Note

Note que duas palavras compartilham uma letra em comum se ambas contêm essa letra em qualquer posição da palavra, independentemente da posição. Por exemplo, “ab” e “bc” compartilham a letra b, portanto estão conectadas.

Problem M. Me Encontre no Meio

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

Palão e seus amigos moram no mesmo condomínio, indexado de 0 a $m - 1$. O condomínio tem um layout circular porque o arquiteto, Peppy, adora círculos. Do prédio i , é possível ir para o prédio $(i + 1) \bmod m$ ou para o prédio $(i - 1) \bmod m$.

Eles querem realizar uma LAN party para jogar Counter-Strike no salão de um dos prédios do condomínio, h . Cada pessoa viajará até h a partir de onde mora pelo caminho mais curto possível ao longo do círculo.

Ajude Palão a determinar a menor soma possível das distâncias percorridas por ele e seus amigos, considerando todas as escolhas possíveis do prédio h .

Input

A primeira linha da entrada contém dois inteiros n, m ($1 \leq n, m \leq 2 \cdot 10^5$), o número de pessoas que jogarão na LAN party e o número de prédios no condomínio, respectivamente.

A segunda linha contém n inteiros $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq m - 1$), onde a i -ésima pessoa mora.

Output

Imprima a menor soma possível das distâncias percorridas pelos jogadores, considerando todas as escolhas possíveis do prédio h .

Examples

standard input	standard output
3 10 0 0 9	1
5 5 3 4 3 3 0	3

Note

No primeiro exemplo, se $h = 0$, a_3 pode ir em um movimento para $(9 + 1) \bmod 10 = 0$.