

Universidade de Brasília

---

# Seletiva para a Maratona de Programação 2016

---

18 de junho de 2016

## A) Sobre a entrada

1. A entrada de seu programa deve ser lida da *entrada padrão*.
2. Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
3. Cada linha, incluindo a última, contém o caractere final-de-linha.
4. O final da entrada coincide com o final do arquivo.

## B) Sobre a saída

1. A saída de seu programa deve ser escrita na *saída padrão*.
2. Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
3. Cada linha, incluindo a última, deve conter o caractere final-de-linha.

## C) Sobre os problemas

As situações retratadas nos problemas são inteiramente fictícias e não correspondem à realidade. Nada escrito nos enunciados tem a intenção de desrespeitar o leitor. Tudo foi escrito de maneira a se adequar às histórias do desenho animado escolhido como tema.

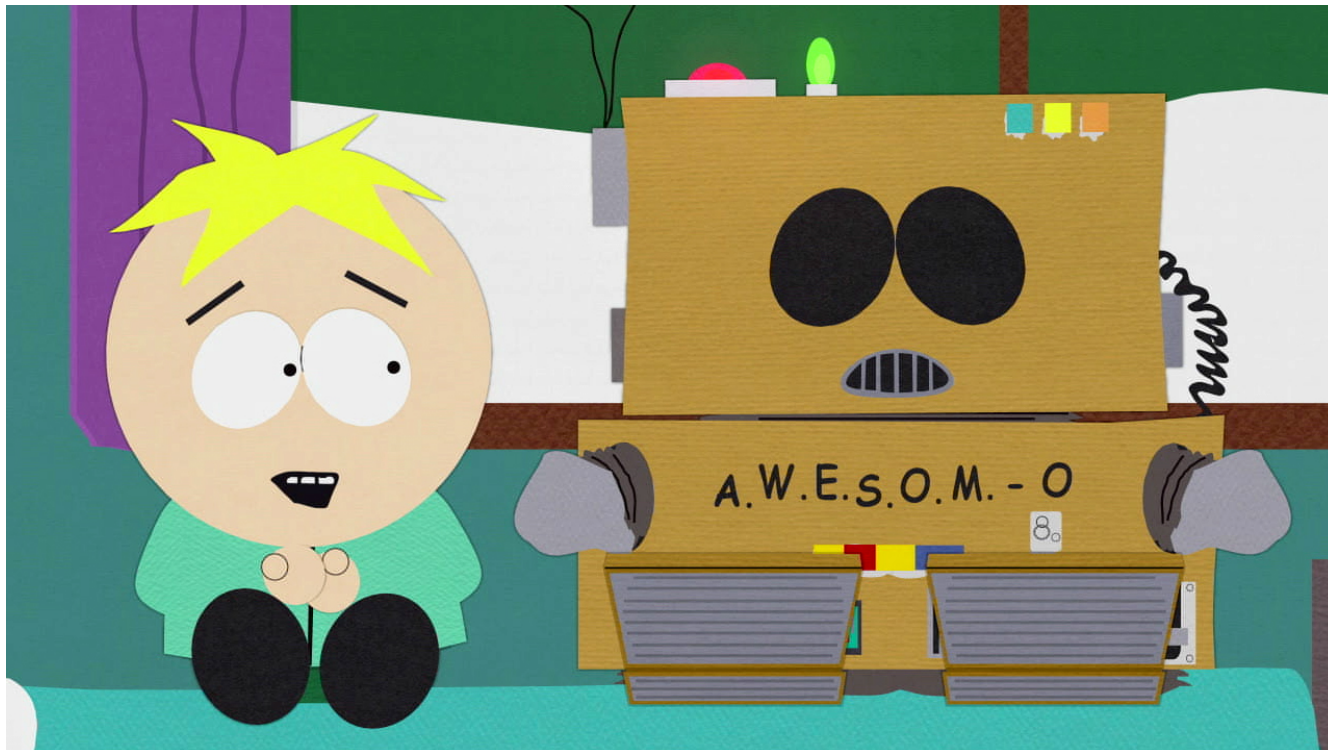
ALL CHARACTERS AND  
EVENTS IN THIS SHOW--  
EVEN THOSE BASED ON REAL  
PEOPLE--ARE ENTIRELY FICTIONAL.  
ALL CELEBRITY VOICES ARE  
IMPERSONATED.....POORLY. THE  
FOLLOWING PROGRAM CONTAINS  
COARSE LANGUAGE AND DUE TO  
ITS CONTENT IT SHOULD NOT BE  
VIEWED BY ANYONE



# A AWESOM-O

Autor: Matheus Pimenta

Butters está muito feliz com seu novo presente que chegou do Japão, o A.W.E.S.O.M.-O 4000.



Para testar a inteligência de seu amigo robô, Butters inventou um jogo. Ele irá escrever uma palavra e A.W.E.S.O.M.-O deve lê-la e dizer se a palavra é a mesma se for lida de trás para frente.

No entanto, se A.W.E.S.O.M.-O irá acertar as perguntas de Butters é algo que irá depender de você, o programador japonês responsável pela fabricação do robô.

## Entrada

A entrada é composta por uma única linha com uma palavra de no máximo 20 letras minúsculas.

## Saída

Imprima uma linha com a mensagem **SIM** caso a palavra seja a mesma se lida de trás para frente, ou **NAO** em caso contrário.

## Exemplos

Entrada	Saída
leopoldstotch	NAO
awesomosewa	SIM

## B Bradley Biggle

Autor: Matheus Pimenta

Bradley Biggle, cujo verdadeiro nome é Gok'Zarah, é a identidade secreta do grande super-herói conhecido como Cereal de Fruta, integrante da liga Guaxinim e Amigos. Seu planeta natal, Kokujon, é conhecido pelo universo por suas grandes minas de frutas.

Gok'Zarah está neste momento investigando o passado de um planeta longínquo e sombrio, em busca de respostas para derrotar o mal que se abateu sobre a Terra.



A primeira descoberta de Gok'Zarah foi como entrar na câmara proibida de pergaminhos antigos do planeta sombrio. Ele precisa conjurar a marca secreta que abre o portal da câmara. Os  $N$  vértices da marca formam um polígono convexo. Para conjurar a marca, Gok'Zarah precisa decompor o polígono em triângulos e, utilizando seus poderes de hortelã e frutas com deliciosa crocância, traçar no ar os segmentos de reta internos ao polígono associados a esta decomposição.

Para ganhar tempo e evitar as armadilhas desconhecidas do planeta sombrio, Gok'Zarah pediu ajuda ao programador da liga Guaxinim e Amigos (ou seja, a você) para encontrar uma decomposição mínima. Uma decomposição é mínima se a soma dos comprimentos dos segmentos de reta escolhidos para formar a marca for mínima.

### Entrada

A primeira linha da entrada contém o inteiro  $N$ .

As próximas  $N$  linhas dão as coordenadas reais  $x$  e  $y$  de cada um dos  $N$  vértices, em sentido horário.

### Saída

Imprima uma única linha com soma dos comprimentos dos segmentos de reta de uma decomposição mínima. Imprima o valor da soma com 4 casas decimais.

### Restrições

- $3 \leq N \leq 400$
- $0 \leq x, y \leq 10^4$

## Exemplos

Entrada	Saída
4	1.0000
1 0	
0 0	
0 1	
0.7071 0.7071	

## C Criptografando no Pentágono

Autor: Matheus Pimenta

Após o episódio ocorrido com Imaginalândia, em que dois garotos da quarta série o invadiram facilmente, o Pentágono decidiu reformar seus sistemas de segurança.



Em relação a segurança computacional, o Pentágono desenvolveu um novo método para gerar chaves criptográficas. Parte do método funciona da seguinte maneira. Primeiro, seleciona-se um número primo  $p$  e dois inteiros positivos  $M$  e  $N$ . Em seguida, são escolhidas  $M$  sequências de  $N$  inteiros não negativos menores que  $p$ . Para cada sequência que puder ser obtida combinando as  $M$  sequências escolhidas através de somas posição a posição e multiplicações por um número, tomando sempre os restos nas divisões por  $p$  de todos os resultados, gera-se uma chave. Por exemplo, se  $p = 7$ ,  $M = 3$ ,  $N = 2$  e foram escolhidas as sequências  $a_1 = (2, 0)$ ,  $a_2 = (0, 6)$  e  $a_3 = (1, 0)$ , então  $(1, 5)$  é uma das sequências que irá gerar uma chave criptográfica, pois

$$a_3 + 2a_2 = (1, 0) + 2 \cdot (0, 6) = (1, 0) + (2 \cdot 0, 2 \cdot 6) = (1, 0) + (0, 12) = (1 + 0, 0 + 12) = (1, 12)$$

Tomando os restos nas divisões por  $p$ , temos que

$$(1, 12) \equiv (1, 5) \pmod{7}$$

Os programadores da equipe de segurança computacional do Pentágono perceberam que algumas das  $M$  sequências escolhidas para gerar chaves criptográficas podem ser descartadas, sem alterar o conjunto de chaves que podem ser geradas. Você é o programador da equipe que ficou responsável por calcular o número máximo de sequências que podem ser descartadas!

### Entrada

A primeira linha da entrada contém os inteiros  $p$ ,  $M$  e  $N$ .

As próximas  $M$  linhas descrevem as  $M$  sequências de  $N$  inteiros não negativos menores que  $p$ .

### Saída

Imprima uma única linha com o número máximo de sequências que podem ser descartadas.

## Restrições

- $2 \leq p < 10^9$  e  $p$  é primo
- $1 \leq M, N \leq 100$

## Exemplos

Entrada	Saída
7 1 1 5	0
7 1 1 0	1
7 3 2 2 0 0 6 1 0	1
7 3 2 4 0 6 0 3 0	2
7 3 2 0 0 0 0 0 0	3



## D Desordenando a Blockbuster

Autor: Matheus Pimenta

Desde que Randy comprou uma Blockbuster, ele obriga Shelly a organizar o estoque de filmes, o que ela faz contra a própria vontade, porque sabe que será feito inutilmente.



Hoje, a paciência de Shelly se esgotou. Para irritar seu pai, ela decidiu que irá escolher filmes do estoque e colocá-los em uma nova ordem nas prateleiras da loja; e que ela irá fazer isto todos os dias. Para ter certeza de que esta estratégia irá irritar seu pai por muito tempo, ela pediu ajuda ao namorado programador (você) para calcular o número total de escolhas distintas que ela pode fazer ao selecionar  $K$  filmes e colocá-los em uma ordem específica nas  $K$  posições das prateleiras, sabendo que o estoque da Blockbuster de Randy possui  $N$  filmes, rotulados de 1 a  $N$ , e que há exatamente  $c_i$  cópias do  $i$ -ésimo filme.

### Entrada

A primeira linha da entrada contém os inteiros  $N$  e  $K$ .

As próximas  $N$  linhas contêm os valores  $c_i$ , dados em ordem, começando por  $c_1$ .

### Saída

Imprima uma única linha com o valor pedido no enunciado. Como este valor pode ser muito grande, imprima-o módulo  $10^9 + 7$ .

### Restrições

- $1 \leq N, K, c_i \leq 100$
- $K \leq \sum_{i=1}^N c_i$

### Exemplos

Entrada	Saída
1 2 5	1
5 2 1 1 1 1 1 1	20

## E Eleições em South Park Elementary

Autor: Matheus Pimenta

Estão acontecendo as eleições para mascote na Escola Elementar de South Park! Este ano estão concorrendo os candidatos, da esquerda para a direita na foto, Giant Douche e Turd Sandwich.



Você é o programador da escola e ficou responsável por computar os votos da eleição!

### Entrada

A primeira linha da entrada contém um inteiro  $N$ .

Cada uma das próximas  $N$  linhas descreve um voto. Votos para Douche são representados por `douche`, enquanto votos para Turd são representados por `turd`.

### Saída

Imprima uma linha para cada candidato com o número de votos, como mostram os exemplos.

Em seguida, imprima uma linha com o resultado da eleição. Caso haja mais votos para Douche, a mensagem impressa deve ser **“Giant Douche venceu!”**. Caso haja mais votos para Turd, a mensagem é **“Turd Sandwich eh o novo mascote!”**. Caso haja empate, a mensagem é **“Um novo turno sera necessario.”**.

### Restrições

- $2 \leq N \leq 10^5$

### Exemplos

Entrada	Saída
2 turd douche	Douche: 1 Turd: 1 Um novo turno sera necessario.
3 douche douche douche	Douche: 3 Turd: 0 Giant Douche venceu!
3 turd douche turd	Douche: 1 Turd: 2 Turd Sandwich eh o novo mascote!

## F Fatbeard

Autor: Matheus Pimenta

Fatbeard é um pirata famoso por bolar planos, que de vez em quando dão certo.



Em sua última viagem, Fatbeard roubou um mapa extremamente valioso, criado por um pirata lendário. O mapa inclui todas as ilhas e correntes marítimas do oceano. Mas a informação mais valiosa que o pirata lendário deixou, foram suas estimativas de quanto ouro é possível saquear do arquipélago pelo qual uma dada corrente marítima passa. Especificamente, há  $N$  grandes regiões calmas no oceano, ou seja, há  $N$  grandes regiões por onde não passa nenhuma corrente marítima. Estas regiões estão rotuladas no mapa com os números naturais de 1 a  $N$ . Para cada uma das  $M$  correntes marítimas, o mapa mostra uma estimativa  $G$  de quanto ouro pode ser saqueado do arquipélago pelo qual ela passa, a região calma  $A$  onde ela começa e a região calma  $B$  onde ela termina. Há no máximo uma corrente marítima fluindo de uma região  $A$  para uma região  $B$ .

De posse das informações do mapa lendário, Fatbeard bolou o seguinte plano. Primeiro, ele chegou a uma conclusão razoável: para fazer uma viagem eficiente, isto é, rápida e lucrativa, ele decidiu navegar somente por correntes marítimas. Em seguida, ele raciocinou da seguinte forma. Os outros piratas são mais fortes, mas são muito mais preguiçosos. Eles só escondem seus tesouros em regiões calmas do oceano e só saem de uma região calma se puderem voltar a ela navegando por correntes. Portanto, para evitar combates em que o prejuízo seria garantido, Fatbeard decidiu saquear somente os arquipélagos das correntes pelas quais outros piratas não navegam e esconder cada saque em alguma ilha perto do fim da corrente marítima.

Você, que é o talentoso papagaio calculista de Fatbeard, ficou responsável por examinar o mapa e calcular o lucro do trajeto mais lucrativo possível, considerando que o barco de Fatbeard está inicialmente na região calma rotulada por  $F$ .

### Entrada

A primeira linha da entrada contém os inteiros  $N$ ,  $M$  e  $F$ .

As próximas  $M$  linhas representam as correntes marítimas. Para cada corrente são dados os três inteiros  $G$ ,  $A$  e  $B$ .

### Saída

Imprima uma linha com o lucro máximo que Fatbeard pode obter seguindo o plano.

## Restrições

- $1 \leq N \leq 10^6$
- $0 \leq M \leq \min(N(N-1), 10^6)$
- $1 \leq G \leq 10^6$
- $1 \leq F, A, B \leq N$  e  $A \neq B$

## Exemplos

Entrada	Saída
2 1 1 7 1 2	7
6 7 3 70 1 2 50 2 3 60 3 1 30 2 6 30 3 6 20 2 4 25 4 5	45

## G Guitar Queer-O

Autor: Matheus Pimenta

Após comprarem o novo Guitar Hero III: Legends of Rock, Stan e Kyle têm treinado incessantemente para se tornarem astros do rock.



Acontece que a cópia do jogo dos garotos veio cheia de *bugs*! Agora eles precisam da sua ajuda para continuarem o treinamento!

Dada a sequência de combinações de teclas da música e as duas sequências de teclas pressionadas por Stan e Kyle, determine a pontuação total dos garotos.

Cada combinação tem no máximo 5 teclas e pode dar no mínimo -5 e no máximo 5 pontos para cada garoto. Dadas duas combinações de teclas, a combinação correta e a combinação pressionada, cada tecla fixa que aparece simultaneamente em ambas as combinações acrescenta um ponto para o garoto que a pressionou. Além disso, para cada tecla que foi pressionada quando não deveria, ou que não foi pressionada quando deveria, um ponto é subtraído.

### Entrada

A primeira linha da entrada contém um único inteiro  $N$ , o número de combinações de teclas da música.

As próximas  $N$  linhas representam as combinações de teclas que devem ser pressionadas para tocar a música corretamente.

As próximas  $N$  linhas representam as combinações pressionadas por Stan.

As próximas  $N$  linhas representam as combinações pressionadas por Kyle.

Uma combinação de teclas é uma string no formato G-R-Y-B-O. Variações deste formato onde algumas letras são substituídas por - (hífen) também são combinações válidas. Cada letra representa uma tecla.

### Saída

Imprima uma linha com a pontuação total dos garotos.

Caso a pontuação total seja a maior possível, imprima outra linha com a mensagem CONGRATULATIONS, YOU ARE FAGS!.

Caso a pontuação total seja a menor possível, imprima outra linha com a mensagem GAME OVER, YOU SUCK!.

## Restrições

- $1 \leq N \leq 10^5$

## Exemplos

Entrada	Saída
1 --R-Y-B-- --R-Y---0 --R-Y-B-0	2
1 --R-Y-B-- --R-Y-B-- --R-Y-B-0	5
1 --R-Y-B-- --R-Y-B-- --R-Y-B--	6 CONGRATULATIONS, YOU ARE FAGS!
1 --R-Y-B-- G-----0 G-----0	-10 GAME OVER, YOU SUCK!

## H Heaven vs. Hell

Autor: Matheus Pimenta

Mataram o Kenny! E, desta vez, foi Deus.



Deus levou a alma de Kenny ao Céu por ele ter demonstrado grandes habilidades no jogo Heaven vs. Hell, atingindo o nível 60. O jogo foi publicado na Terra por intervenção divina como um esforço para selecionar o Keanu Reeves que vencerá a iminente guerra contra o Inferno. Kenny deve controlar o exército do Céu através do PSP Dourado, o Rei de Todos os PSPs. Você, o cérebro inteligente e raro de Kenny, deve escolher os alvos durante a batalha de maneira a minimizar o dano total que será causado ao Céu.

Uma série de  $N$  demônios, rotulados de 0 a  $N - 1$ , entram Céu a dentro, um por minuto. O rótulo de um demônio indica também o instante de tempo, medido em minutos, em que ele entra no Céu. Por exemplo, o demônio rotulado por 0 é o primeiro a chegar; o demônio 1 entra no Céu após decorrido um minuto de batalha; o demônio 2 entra no Céu após decorridos dois minutos de batalha; e assim por diante.

Todos os demônios marcham em direção à ICPC — Incrível Cidadela de Proteção Celestial — que Kenny precisa defender. O  $i$ -ésimo demônio entra no Céu a uma distância de exatamente  $T_i$  metros da ICPC e se move em direção a ela com uma velocidade constante de  $v_i$  metros por minuto. Ao chegar no ponto onde a ICPC se encontra, o demônio pára de se mover. Portanto, a distância  $s_i$  do  $i$ -ésimo demônio até o ponto em que ele entrou no Céu, em função do tempo  $t$  que já se passou desde o início da batalha, é descrita da seguinte forma:

$$s_i(t) = \begin{cases} v_i(t - i) & \text{se } t \in [i, i + T_i/v_i) \\ T_i & \text{se } t \in [i + T_i/v_i, \infty) \end{cases}$$

Estima-se que a quantidade de dano por minuto  $d_i$  causada ao Céu pelo  $i$ -ésimo demônio é diretamente proporcional à distância em que ele se encontra do local pelo qual ele entrou no Céu. Em outras palavras,

$$d_i(s_i) = D_i s_i, \forall s_i \in [0, T_i]$$

onde  $D_i$  é uma constante de proporcionalidade, ou seja,  $D_i$  é uma medida da força de destruição do  $i$ -ésimo demônio.



Combinando as duas últimas equações, temos que a quantidade de dano por minuto causada ao Céu pelo  $i$ -ésimo demônio, em função do tempo, é dada por:

$$d_i(t) = \begin{cases} 0 & \text{se } t \in [0, i) \\ D_i v_i (t - i) & \text{se } t \in [i, i + T_i/v_i) \\ D_i T_i & \text{se } t \in [i + T_i/v_i, \infty) \end{cases}$$

Logo, o dano causado no intervalo  $[t_a, t_b]$  é dado pela integral:

$$\int_{t_a}^{t_b} d_i(t) dt$$

Kenny ataca da seguinte maneira. Primeiro, ele escolhe como alvo um dos demônios que já estão no Céu naquele instante (claro). Em seguida, ele comanda que o exército do Céu conjure um ataque de raios divinos sobre o alvo. O ataque leva  $C$  minutos para ser carregado. Enquanto o ataque está sendo carregado, Kenny não pode fazer nada. O alvo é desintegrado instantaneamente assim que o ataque o atinge. Kenny precisa escolher e comandar a destruição do próximo alvo no instante imediatamente seguinte à destruição do alvo anterior. Afinal, isto é uma guerra! Não há tempo a perder! É preciso defender a ICPC! Por fim, a capacidade de Kenny para prever o futuro é limitada. Ele só consegue calcular o que vai acontecer nos próximos  $2C$  minutos, contando a partir do instante atual.

## Entrada

A primeira linha da entrada contém os inteiros  $N$  e  $C$ .

As próximas  $N$  linhas descrevem os  $N$  demônios que vão invadir o Céu, por ordem de chegada. Cada linha contém os inteiros  $T_i$ ,  $v_i$  e  $D_i$ .

## Saída

Imprima uma única linha com o dano mínimo que pode ser causado ao Céu.

## Restrições

- $1 \leq N \leq 3 \cdot 10^3$
- $1 \leq C \leq 10$
- $1 \leq T_i, v_i, D_i \leq 10^4$
- $T_i$  é múltiplo de  $v_i$
- $D_i$  é par

## Exemplos

Entrada	Saída
3 3 10 5 4 12 4 2 9 3 2	263
3 3 10 5 4 12 4 2 9 3 8	416



# I Intergaláticos Cometas e a Princesa do Canadá

Autor: Matheus Pimenta

Esta semana, Cartman comentou em seu canal do YouTube — Cartman Braaaaa — uma partida do mais novo jogo lançado por Terrance e Phillip: Intergaláticos Cometas e a Princesa do Canadá (ICPC). No jogo, Terrance e Phillip sobem em um cometa para atravessarem o espaço na missão de resgatar a princesa do Canadá, que foi raptada por aliens. Cartman fez um acompanhamento emocionante da fase final do jogo e, ao final do vídeo, propôs uma enquete para seus seguidores. “Qual vocês acham que é a menor quantidade de energia no início da fase necessária para vencer o jogo???” Você odeia Cartman, mas seu irmãozinho mais novo, que acompanha o canal e reconhece suas habilidades como programador, te pediu ajuda para responder a pergunta. Ajude-o!



Na última fase do jogo, o cometa de Terrance e Phillip inicia com a energia cinética adquirida pelo jogador nas fases anteriores. Durante a perseguição, o cometa perde energia ao longo do trajeto. Isto acontece porque a nave alienígena que raptou a princesa começa a deixar um rastro atmosférico gigantesco que impõe resistência ao movimento do cometa. Felizmente, junto com a atmosfera criada pela nave, são criados também alguns locais com resíduos inflamáveis que incendeiam a cauda do cometa, fornecendo a ele uma grande quantidade de energia cinética instantaneamente. Se a energia do cometa se anular (ficar menor ou igual a zero) *estritamente antes* de atingir a nave, ele ficará parado para sempre e o jogador perde o jogo. Caso contrário, o jogador vence.

Estudando o jogo, você descobriu quantos quilômetros tem a trajetória da última fase e os rotulou começando pelo número 1. Além disso, com sua grande astúcia de programador, você percebeu que o jogo é programado para que  $N$  eventos fixos sempre afetem o cometa na última fase, independentemente de como o jogador controlar o cometa. Cada um dos  $N$  eventos pode estar em um dos formatos a seguir:

- 1  $Q$   $J$ : Indica mudança na resistência da atmosfera deixada pela nave alienígena. A partir do *início* do quilômetro  $Q$ , a energia cinética irá decair exatamente  $J$  Joules por quilômetro.
- 2  $Q$   $J$ : Exatamente no *final* do quilômetro  $Q$ , há vários resíduos inflamáveis tais que se o cometa se aproximar de um deles, exatamente  $J$  Joules de energia cinética são transferidos ao cometa.
- 3  $Q$ : O cometa alcança a nave alienígena exatamente no *final* do quilômetro  $Q$  e Terrance e Phillip resgatam a princesa do Canadá.

Com todas estas informações que você já reuniu sobre a última fase do jogo ICPC, calcule a menor quantidade inteira de energia cinética que o jogador deve adquirir para o cometa antes de ir para a fase final do jogo, de modo que seja possível finalmente resgatar a princesa do Canadá.

## Entrada

A primeira linha da entrada contém o inteiro  $N$ .

As próximas  $N$  linhas representam os eventos que acontecem na última fase do jogo ICPC, com os formatos descritos no enunciado. O primeiro evento é sempre da forma  $1\ 1\ J$ , indicando a primeira resistência atmosférica que o cometa enfrenta. Somente o último evento é da forma  $3\ Q$ . Os  $N-2$  eventos intermediários podem ser dos tipos 1 e 2 e estão em ordem estritamente crescente de  $Q$ . Os valores  $Q$  e  $J$  são sempre inteiros.

## Saída

Imprima uma linha com a menor energia cinética inteira que permita um jogador vencer o jogo ICPC.

## Restrições

- $2 \leq N \leq 10^4$
- $1 \leq Q \leq 10^5$
- $0 \leq J \leq 10^5$

## Exemplos

Entrada	Saída
2 1 1 0 3 10	1
2 1 1 1 3 10	10
3 1 1 2 2 5 5 3 10	15
3 1 1 2 1 6 1 3 10	15

## J Jornada em Azeroth

Autor: Matheus Pimenta

Stan está viciado em World of Warcraft. Vamos torcer para que, no futuro, ele troque para o vício de treinar para a Maratona de Programação, que é o único vício saudável, certo?!

Acontece que Randy não deixa Stan se concentrar! Então ele pediu a sua ajuda para completar uma *quest*.



O mundo de Warcraft, Azeroth, é composto por  $N$  cidades, rotuladas de 1 a  $N$ . Há também  $M$  regiões perigosas, como florestas, pântanos, cavernas, desertos, montanhas, planícies, mares, lagos e rios. Cada uma destas  $M$  regiões perigosas conecta um par de cidades distintas e não há mais de uma região conectando um mesmo par de cidades. Além disso, ao atravessar uma destas regiões, o personagem de Stan precisa gastar uma certa quantidade de *magic points* para lidar com os monstros que estão no caminho.

A *quest* funciona da seguinte maneira. Stan, que está inicialmente na cidade  $S$ , precisa buscar uma gema sagrada na cidade  $G$  e levá-la até o seu templo, que fica na cidade  $T$ . Acontece que há  $K$  gemas profanas escondidas em  $K$  cidades distintas, rotuladas por  $P_i$ ,  $1 \leq i \leq K$ ,  $1 \leq P_i \leq N$ . Se Stan entrar em uma destas cidades com a gema sagrada, o poder sagrado que vive na gema sairá de dentro dela. Mas Stan tem a opção de guardar a gema sagrada no armazém de qualquer cidade onde não houver uma gema profana e buscá-la mais tarde, pois ele pode entrar em qualquer uma das  $K$  cidades para encontrar e destruir a gema profana que lá foi escondida.

Calcule a quantidade mínima de *magic points* necessária para completar a *quest*.

### Entrada

A primeira linha da entrada contém os inteiros  $N$ ,  $M$  e  $K$ .

A segunda linha contém os inteiros  $S$ ,  $G$ ,  $T$  e os  $K$  inteiros  $P_i$ .

As próximas  $M$  linhas descrevem as  $M$  regiões perigosas. Cada linha contém os inteiros  $A$ ,  $B$  e  $M_p$ , indicando que a região conecta as cidades  $A$  e  $B$  e custa  $M_p$  *magic points* para ser atravessada.

### Saída

Imprima uma única linha com a resposta.

## Restrições

- $2 \leq N \leq 20$
- $1 \leq M \leq N(N-1)/2$
- $0 \leq K \leq \min(N - |\{S, G\}|, 10)$
- $1 \leq S, G, T, P_i, A, B \leq N$ ,  $G \neq T$ ,  $P_i \neq S$  e  $P_i \neq G$ , para todo  $i$
- $1 \leq M_p \leq 10^4$

## Exemplos

Entrada	Saída
3 2 1	320
1 3 2 2	
1 2 100	
1 3 10	

## K Kyle Schwartz

Autor: Matheus Pimenta

O primo chato de Kyle está em South Park novamente e, desta vez, ele propôs um jogo a Kyle.



— Kyle 2, eu vou escolher um número primo  $p$  e dois números  $a$  e  $g$  maiores que zero e menores que  $p$ . Depois eu vou multiplicar  $a$  por  $g$  várias vezes e vou te dizer o resto desse produto na divisão por  $p$ . Você tem que me dizer quantas vezes eu multipliquei!

— Cara, mas esse jogo é muito chato! Só existem  $p$  restos possíveis quando se divide por  $p$ ! Se o resto se repetir enquanto você estiver multiplicando, como eu vou saber???

— Eu prometo que não vou deixar isso acontecer!

Parece que Kyle não vai deixar Kyle 2 em paz. Ajude Kyle 2 a se livrar de seu primo!

### Entrada

A única linha da entrada contém quatro inteiros  $a$ ,  $b$ ,  $g$  e  $p$ , onde  $b = ag^k \pmod{p}$ , para algum  $k \geq 0$ .

### Saída

Imprima uma única linha com o valor de  $k$ .

### Restrições

- $1 \leq a, b, g < p < 10^9$  e  $p$  é primo

### Exemplos

Entrada	Saída
8 8 2 13	0
1 8 2 13	3
1 9 2 13	8
9 6 2 13	9

EXECUTIVE PRODUCERS  
TREY PARKER  
MATT STONE