

VII Maratona de Programação do IFB

Caderno de Problemas

19 de novembro de 2022



(Este caderno contém 13 problemas)

Comissão Organizadora:

Prof. Daniel Saad Nogueira Nunes (IFB)
Prof. Edson Alves da Costa Júnior (UnB/FGA)
Prof. Guilherme Novaes Ramos (UnB)
Prof. Vinicius Ruela Pereira Borges (UnB)
Alberto Tavares
Enzo Yoshio
Jeremias Moreira Gomes
José Marcos da Silva Leite

Lembretes

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova, entretanto, o mesmo não vale para materiais dispostos eletronicamente.
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída conforme as amostras dos exemplos. Deve-se considerar entradas e saídas padrão;
- Para cada problema, além dos testes públicos, o juiz executará a sua submissão contra uma série de testes secretos para fornecer um parecer sobre a correção do programa.
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. Lembre-se que as soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Utilize a aba *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;

C/C++

- Seu programa deve retornar zero, executando, como último comando, `return 0` ou `exit 0`.

Java

- Não declare ‘`package`’ no seu programa Java.
- Note que a convenção para o nome do arquivo fonte deve ser obedecida, o que significa que o nome de sua classe pública deve ser uma letra maiúscula igual a letra que identifica o problema.

Python

- Tenha cuidado ao selecionar a versão correta na submissão.

Problema A

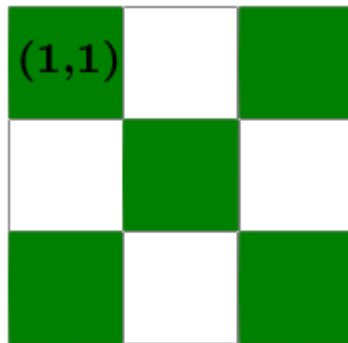
Azulejos

Limite de tempo: 1s

Autor: Vinicius Borges

Edson está construindo sua casa e planeja revestir uma parede com lajotas que possuem as cores da Sociedade Esportiva do Gama, seu clube de coração. Essa parede possui dimensões $N \times M \text{ m}^2$ (altura \times largura), em que cada azulejo possui dimensão $1 \times 1 \text{ m}^2$, sendo ou da cor verde ou da cor branca.

A parede será revestida como se fosse uma malha, em que deve-se alternar as cores dos azulejos. Assim, a cor do azulejo na posição (i, j) deve ser diferente da cor dos azulejos vizinhos na vertical e na horizontal. O azulejo na posição $(1, 1)$ (canto esquerdo superior da parede) sempre recebe a cor verde. A figura abaixo ilustra uma parede 3×3 revestida conforme o processo descrito acima:



Edson sabe que é mais fácil obter azulejos brancos do que azulejos verdes no comércio do Distrito Federal. Para que não falte nenhum tipo de azulejo no revestimento dessa parede, ajude-o a determinar a quantidade total de azulejos verdes que serão utilizados.

Entrada

A primeira linha da entrada contém dois números inteiros N e M ($1 \leq N, M \leq 2 \cdot 10^9$) separados por espaço em branco indicando a altura e a largura da parede, respectivamente.

Saída

Imprima um único número inteiro com a resposta do problema: a quantidade de azulejos verdes que serão utilizados para compor o revestimento da parede.

Exemplo

Entrada	Saída
3 3	5
4 2	4
5 7	18

Problema B

Bolo de FooBar

Limite de tempo: 1s

Autor: Guilherme Ramos

Toda família tem uma receita de bolo que é a melhor do mundo, e os Costajúniors não são exceção. O segredo do sucesso, diz a lenda, é o ajuste gradual da cobertura a medida que ela é preparada. Eles ficam provando e indicando qual dos ingredientes especiais precisa ser adicionado para atingir o ponto ideal. Como a receita não pode sair da família, usam um código secreto para dizer qual a pitada necessária e despistar curiosos.

A código é simples: no início do processo são dados dois números a e b indicando, respectivamente, os ingredientes secretos `foo` e `bar`. Cada um indica o que deve ser adicionado: uma pitada de `foo` se for múltiplo de a , um toque de `bar` se for múltiplo de b , ou nada.

Entrada

A entrada consiste de uma linha com dois inteiros a e b ($0 < a, b \leq 2357$), separados por espaço. A seguir é fornecida uma sequência indeterminada de números inteiros, um por linha ($0 < n \leq 10^9$), indicando o que deve ser acrescentado. A entrada termina com EOF.

Saída

Para cada número dado, apresente o(s) ingrediente(s) a ser(em) acrescentado(s) à receita (em uma linha), se for o caso.

Exemplo

Entrada	Saída
7 11	foo
5	bar
7	foo
9	bar
11	
14	
22	
3 5	foo
1	bar
3	foo
5	bar
6	foobar
10	
15	

Problema C

Ceborinha

Limite de tempo: 1s

Autor: Guilherme Ramos

A turma do Bairro do Limoeiro sempre anda as rusgas com o Capitão Feio. Seu último plano envolvia enganar o Cascão para ajudá-lo a emporcalhar o ambiente, mandando mensagens falsas se passando pelo Cebolinha. Ajude a turminha a não cair nessa cilada!

Dada uma mensagem identificada como vinda do Cebolinha, responda “Bola Ceborinha!”, se achar que for do grande amigo do Cascão, ou “Ta Feio a coisa...” se determinar que veio do Capitão Feio. Por sorte, em seu entusiasmo por ter um mundo mais poluído, esse vilão esqueceu o detalhe mais óbvio: Cebolinha sempre troca “R” por “L”!

Entrada

A entrada consiste de uma única linha contendo não mais que 100 caracteres.

Saída

A saída consiste de uma única linha com a mensagem de resposta do Cascão.

Exemplo

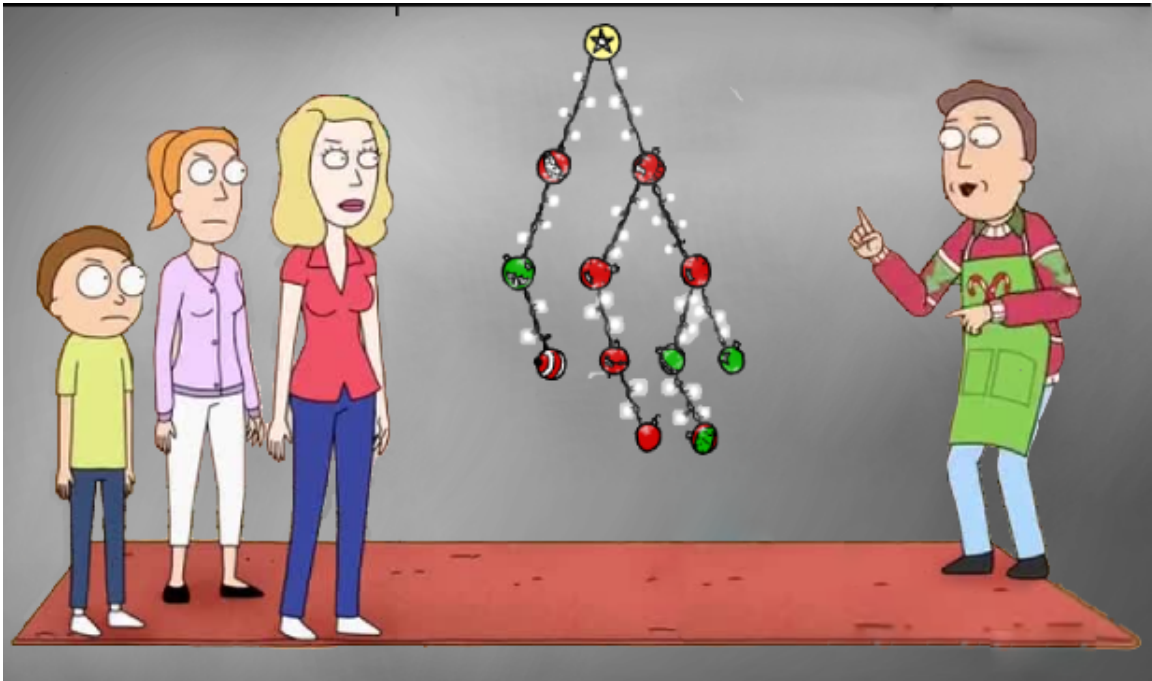
Entrada	Saída
Ola tulma!	Bola Ceborinha!
Pra cima deles Cascao!	Ta Feio a coisa...
Dentuca! Baixinha! Golducha!	Bola Ceborinha!

Problema D

Discussão de Natal

Limite de tempo: 2s

Autor: Jeremias Gomes



Pensando cedo no Natal, Djerry resolveu montar, pela primeira vez, a árvore de natal da família. Como ele nunca tinha feito isso antes, resolveu consultar na internet interdimensional e, no final, acabou montando sem querer uma árvore binária.

Ao chamar a família para mostrar o seu trabalho, Djerry os posicionou de um lado e ficou do outro. O único problema é que eles estão em um mundo bidimensional e, estando de um lado, só é possível enxergar um dos enfeites de cada nível da árvore pois eles obstruem a visão dos demais que estão no mesmo nível. Por causa disso, o que a família de Djerry vê da árvore não são, necessariamente, os mesmos enfeites da árvore que Djerry vê.

Dessa forma, Djerry pediu a sua ajuda para, dada a constituição da árvore construída por ele, informar quais são as visões da árvore de sua família e dele mesmo, quando olhada pelos seus respectivos lados.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 2 * 10^5$), que indica o número de enfeites da árvore.

Cada uma das N linhas seguintes contém três inteiros B ($1 \leq B \leq N$), E e D ($0 \leq E, D \leq N$), separados por um espaço em branco, que indicam o índice do enfeite (onde a estrela, ou topo da árvore, sempre é o índice 1) e as conexões a esquerda e a direita desse enfeite. Se uma conexão não existir, será representada por 0.

Saída

A primeira linha da saída deve conter um inteiro V , que indica o número de enfeites que podem ser visualizados tanto pela família, quanto por Djerry.

A segunda linha deverá conter V inteiros, em qualquer ordem, separados por um espaço, com os índices dos enfeites da árvore que podem ser vistos pela família de Djerry.

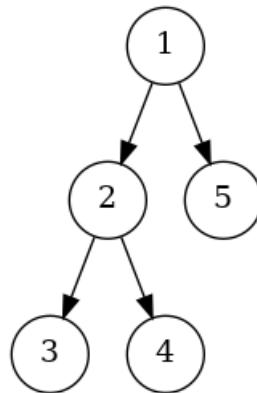
A terceira linha deverá conter V inteiros, em qualquer ordem, separados por um espaço, com os índices dos enfeites da árvore que podem ser vistos por Djerry.

Exemplo

Entrada	Saída
5	3
1 2 5	1 2 3
2 3 4	1 5 4
3 0 0	
4 0 0	
5 0 0	
12	5
1 4 2	1 4 3 5 10
2 6 8	1 2 8 11 12
3 0 5	
4 3 0	
5 0 0	
6 0 7	
7 0 10	
8 9 11	
9 0 12	
10 0 0	
11 0 0	
12 0 0	
4	3
1 3 4	1 3 2
2 0 0	1 4 2
3 0 0	
4 0 2	

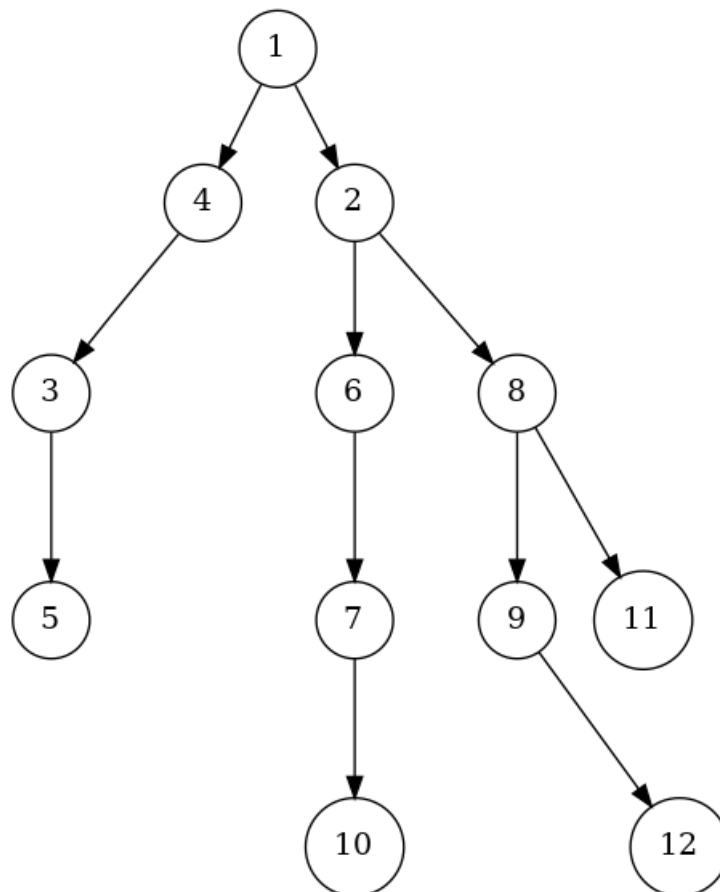
Notas

No primeiro caso de teste, a árvore construída por Djerry é a seguinte:



Nessa árvore, a família de Djerry tem visão apenas para os enfeites 1, 2 e 3, enquanto Djerry tem visão apenas para os enfeites 1, 5 e 4.

Já no segundo caso de teste, a família de Djerry tem visão para os enfeites 1, 3, 4, 5 e 10, enquanto Djerry tem visão para 1, 2, 8, 11 e 12.



Problema E

Empreitada Perigosa

Limite de tempo: 3s

Autor: Vinicius Borges

Blayton é um repórter que gosta de se arriscar e fazer reportagens que envolvem atividades radicais e extremas. Sua próxima meta é realizar uma longa caminhada que durará D dias seguidos, sem interrupções para descanso. Com isso, Blayton terá que se abster de alimentar, mas **nunca fica mais que um dia sem se alimentar**.

A alimentação de Blayton é organizada da seguinte maneira: ele organiza N kits de comida, sendo que cada um é totalmente consumido em um dia que Blayton escolhe se alimentar.

Ajude Blayton no planejamento dessa aventura arriscada. Determine a quantidade de estratégias de alimentação que ele pode adotar para completar com sucesso sua aventura, sem correr o risco de ficar dois dias consecutivos sem se alimentar.

Entrada

A primeira linha contém dois números inteiros D e N ($2 \leq D \leq 30, 1 \leq N \leq D$), indicando a quantidade total de dias seguidos que dura a aventura de Blayton e a quantidade de kits de comida preparados por Blayton antes da sua aventura começar, respectivamente.

Saída

Imprima um número inteiro com a resposta para o problema – a quantidade total de estratégias de alimentação que Blayton pode adotar para completar o percurso, sem correr o risco de ficar dois dias consecutivos sem se alimentar.

Exemplo

Entrada	Saída
3 2	4
5 3	7
6 2	0

Notas

No primeiro exemplo, para os três dias de aventura, existem quatro possíveis estratégias de alimentação. Suponha que “A” seja um dia que Blayton decide se alimentar e “X” um dia que ele não decide se alimentar. As estratégias de alimentação para sua aventura são: $\{A - X - A\}$, $\{X - A - A\}$, $\{A - A - X\}$ e $\{X - A - X\}$.

No segundo exemplo, as estratégias de alimentação são $\{X - A - X - A - X\}$, $\{X - A - A - X - A\}$, $\{X - A - A - A - X\}$, $\{X - A - X - A - A\}$, $\{A - X - A - A - X\}$, $\{A - X - A - X - A\}$ e $\{A - A - X - A - X\}$.

No terceiro exemplo, não é possível elaborar nenhuma estratégia de alimentação para que Blayton complete a jornada com 2 kits, dada as restrições do problema.

Problema F

Fantástica Fábrica de Fibonacci

Limite de tempo: 2s

Autor: Daniel Saad

A Fantástica Fábrica de Fibonacci é responsável pela maior produção de chocolates do mundo. Para produzir os chocolates, eles utilizam uma fórmula mágica, em que os ingredientes são despejados em quantidades presentes na sequência de Fibonacci. Por exemplo, para produzir um chocolate de 15 gramas, podem ser utilizados ingredientes de peso 2, 8 e 5.

Uma linha especial de chocolates dessas fábricas possui um convite dourado cujo detentor poderá visitar a fábrica para conhecer as suas instalações. Estes chocolates possuem uma propriedade especial: além das quantidades de ingredientes utilizados pertencerem à sequência de Fibonacci, estas quantidades nunca são consecutivas, com respeito à sequência.

Buscando minimizar custos, o administrador da fábrica, William, quer saber o número mínimo de ingredientes para fabricar os chocolates que possuirão o convite dourado.

Entrada

A primeira linha da entrada possui um inteiro n ($1 \leq n \leq 10^5$), que representa o número de consultas a serem respondidas.

As próximas n linhas possuem, cada, um inteiro q_i ($1 \leq q_i \leq 10^{18}$), que indica o peso do chocolate especial que deve ser fabricado.

Saída

Para cada consulta, imprima uma linha com o número mínimo de ingredientes utilizados para fabricação do chocolate especial.

Exemplo

Entrada	Saída
3	1
1	1
5	2
15	

Problema G

Gato na Caixa

Limite de tempo: 1s

Autor: Alberto Tavares



Miau

O gato Clark está preso em uma caixa! Seu dono acaba de colocar ração em seu potinho, então o gato procura uma maneira de sair da caixa o mais rápido possível, e é seu trabalho ajudá-lo.

Mais formalmente, a caixa é representada por um vetor h_1, h_2, \dots, h_N de tamanho N tal que $h[i]$ representa a altura da seção i , e a distância horizontal entre as posições i e j é $|i - j|$. Seja p a altura que Clark consegue pular. O gato conseguirá sair da caixa se ele estiver em uma posição i tal que $p \geq h[i]$.

Para cada uma das q consultas no formato $x \ p$, onde x é um inteiro representando a posição inicial de Clark e p é um inteiro representando a altura máxima que ele pula, ajude-o e diga a menor distância horizontal que ele deverá andar para sair da caixa.

Entrada

A primeira linha de entrada contém dois inteiros N e Q ($2 \leq N \leq 10^5$, $1 \leq Q \leq 10^5$) — o tamanho da caixa e quantidade de consultas, respectivamente.

A segunda linha contém N inteiros $h_1 \dots h_N$ ($1 \leq h[i] \leq 10^9$) — as alturas das seções da caixa.

As próximas Q linhas contém dois inteiros x, p ($1 \leq x \leq n$, $1 \leq p \leq 10^9$) — as Q consultas a serem respondidas.

Saída

Para cada uma das Q consultas, imprima a menor distância horizontal que Clark terá que andar para sair da caixa, ou “Miau” caso seja impossível.

Exemplo

Entrada	Saída
6 5	0
8 5 6 5 3 2	1
3 6	4
3 5	5
1 3	Miau
1 2	
5 1	

Problema H

Hahtlehtihcoh

Limite de tempo: 5s

Autor: Daniel Saad

O time do Athletico Paranaense passou por uma reformulação de sua marca em 2018. O seu nome foi alterado de *Clube Atlético Paranaense* para *Club Athletico Paranaense*. O escudo também passou por mudanças, agora com um aspecto mais moderno.

Com as reformulações, a equipe do Athletico também subiu de rendimento e voltou a figurar como protagonista do futebol brasileiro, ganhando vários títulos e disputando outros.

Estas mudanças, no entanto, geraram muitas dúvidas nas cabeças dos torcedores dos outros times, pois eles nunca sabiam onde colocar o “h” do Athletico. Aproveitando da situação, Dorival resolveu *zoar* o seu amigo Athleticano Felipe e resolveu instalar um malware, chamado Malvadão, no computador do amigo, que, aleatoriamente, inseria a letra “h” após uma vogal.

Replique o malware de Dorival e, dada uma palavra qualquer, liste todas as possibilidades da inserção da letra “h” após uma vogal em ordem lexicográfica.

Entrada

A entrada consiste de uma única palavra S ($1 \leq |S| \leq 20$) sobre o alfabeto $\{a, \dots, z\}$. É garantido que S não possui mais do que 10 vogais.

Saída

Imprima, em ordem lexicográfica, e sem repetições, todas as possibilidades do Malvadão prejudicar Felipe.

Exemplo

Entrada	Saída
athletico	ahthlehtico ahthlehticoh ahthlehtihco ahthlehtihcoh athletico athleticoh athletihco athletihcoh athlehtico athlehticoh athlehtihco athlehtihcoh athletico athleticoh athletihco athletihcoh

Problema I

Iluminação Pública

Limite de tempo: 1s

Autor: Edson Alves

As casas da rua principal do município Internet Funciona Bem estão dispostas em uma linha reta. A iluminação pública desta rua é feita por meio de postes de luz inteligentes, que estão conectados a uma central, a qual pode ajustar o raio R de alcance dos postes. Se um poste que está localizado na posição x e a central utilizar um raio r , ele será capaz de iluminar as casas que estão localizadas no intervalo $[x - r, x + r]$.

O prefeito deseja, a título de economia de energia, determinar o valor mínimo de R para o qual todas as casas sejam devidamente iluminadas. Auxilie o prefeito encontrando este valor.

Entrada

A primeira linha da entrada contém o inteiro N ($1 \leq N \leq 2 \times 10^5$), que indica o número de casas na rua.

A segunda linha contém N inteiros x_i ($1 \leq x_1 < x_2 < \dots < x_N \leq 10^9$), separados por um espaço em branco, indicando a localização da i -ésima casa.

A terceira linha da entrada contém o inteiro M ($1 \leq M \leq 2 \times 10^5$), que indica o número de postes de luz.

A quarta linha contém M inteiros p_j ($1 \leq p_1 < p_2 < \dots < p_M \leq 10^9$), separados por um espaço em branco, indicando a localização do j -ésimo poste. Um poste pode compartilhar uma mesma localização com uma casa: isto significa que o poste está em frente à residência.

Saída

Imprima, em uma linha, o valor mínimo de R tal que todas as casas da rua sejam iluminadas.

Exemplo

Entrada	Saída
3	2
2 5 10	
2	
4 9	
5	3
2 3 4 5 6	
2	
1 7	
1	999
1	
1	
1000	

Notas

No primeiro caso, o poste 1 iluminará as casas 1 e 2, enquanto que o poste 2 iluminará a casa 3.

No segundo caso, ambos postes iluminarão a casa localizada em 4.

No terceiro caso, o único poste da rua é capaz de iluminar todo o intervalo $[1, 1999]$, o qual inclui a única casa da rua.

Problema J

Jardinagem

Limite de tempo: 2s

Autor: Alberto Tavares

Você é um jardineiro e vai participar da final brasileira da competição da Sociedade Brasileira de Cactos (SBC). Para participar do evento, você deve mandar uma foto de algum cacto cultivado por você. Nesta edição de 2022 a organização do evento requer que os participantes enviem uma **única** foto!

O seu cacto é representado por um polígono convexo de N vértices p_1, p_2, \dots, p_n , tais que p_i e p_j possuem uma aresta entre si quando $|i - j| = 1$, ou $j = n$ e $i = 1$.

Este ano você conseguiu cultivar um cacto muito grande! Para passar esta impressão para os juízes, você deve calcular o maior perímetro que pode aparecer em uma única foto.

Considere que a abertura da sua câmera possui um ângulo de 180 graus, e você pode tirar a foto de qualquer distância do polígono.

Entrada

A primeira linha contém um único inteiro N ($3 \leq N \leq 10^5$) — a quantidade de vértices do polígono.

Cada uma das próximas N linhas contém dois inteiros x_i e y_i ($0 \leq x_i, y_i \leq 10^9$) — as coordenadas x e y do i -ésimo vértice do cacto.

É garantido que não há duas arestas adjacentes colineares.

Saída

Imprima um único número real — o maior comprimento possível para o perímetro contido em uma única foto.

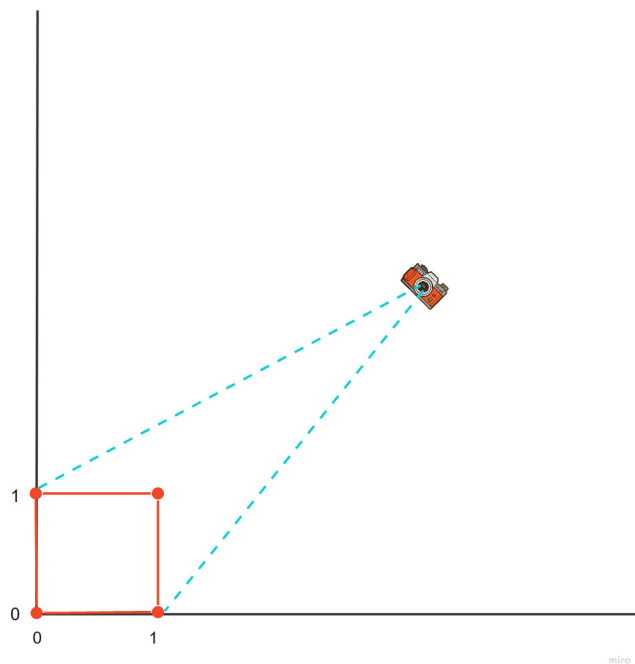
A resposta será considerada correta se a diferença absoluta de sua resposta com a resposta do juiz é menor que 10^{-6} .

Exemplo

Entrada	Saída
4	2.00000000
0 0	
1 0	
1 1	
0 1	
6	25.01113546
4 0	
7 0	
10 1	
10 10	
2 11	
0 9	
4	50.66861157
0 7	
23 8	
23 16	
4 21	

Notas

Uma possibilidade para o caso de teste 1 é



dando perímetro 2, que é o maior possível.

Problema K

Kaindo Raio

Limite de tempo: 2s

Autor: Alberto Tavares

Hoje é dia 19 de novembro e Vinicius está indo ao IFB para a maratona. Porém, ao ver as notícias, Vinicius percebe que a meteorologia prevê que haverá chuva de raios neste dia!

Para cada raio i , Os cientistas previram que ele começará na posição $(0, h_i)$, e se moverá da seguinte forma: em cada instante, se o raio está na posição (x, y) , então ele tem p por cento de probabilidade de ir para $(x - 1, y - 1)$ e $(1 - p)$ por cento de ir para $(x + 1, y - 1)$. O raio colide com o chão quando sua altura y se torna 0.

No i -ésimo raio, Vinicius estima que estará andando na posição $(v_i, 0)$ quando ele atingir o chão. Você deve calcular a probabilidade de cada raio i cair em cima de Vinicius. Como a chuva atrapalha Vinicius a enxergar onde está o ponto decimal, imprima a resposta módulo $10^9 + 7$ (Veja seção de output e notas).

Entrada

A primeira linha de entrada contém um único inteiro n ($1 \leq n \leq 10^5$) — a quantidade de raios que cairão hoje.

Cada uma das próximas n linhas contém três inteiro h_i p_i v_i ($1 \leq h_i \leq 10^7$, $0 \leq p_i \leq 100$, $-10^9 \leq v_i \leq 10^9$) — a altura do i -ésimo raio, a probabilidade deste raio ir para a esquerda (**em por cento**) e a posição de Vinicius quando o raio i atingir o chão, respectivamente.

Saída

Para cada raio i , imprima um único inteiro — a probabilidade do raio i atingir Vinicius, módulo $10^9 + 7$. Isto é, considerando que a resposta é uma fração p/q com $q \neq 0$, imprima $pq^{-1} \pmod{10^9 + 7}$.

Exemplo

Entrada	Saída
6	375000003
4 50 0	398437503
4 25 0	0
6 25 8	0
10 100 10	1
10 0 10	421624333
1234567 65 -741	

Notas

No primeiro exemplo do caso de teste, a resposta é 6/16. Temos $6 \cdot 16^{-1} = 375000003 \pmod{10^9 + 7}$, então a resposta deste exemplo é 375000003.

No segundo exemplo, a probabilidade é 54/256, e $54 \cdot 256^{-1} = 398437503 \pmod{10^9 + 7}$.

Pode ser provado que a resposta sempre pode ser escrita como uma fração p/q com $q \neq 0$.

Problema L

Lógica do Talvez

Limite de tempo: 1s

Autor: Edson Alves

Onofre é muito indeciso, o que o levou a julgar cada fato como “verdadeiro” (V), “falso” (F) ou “talvez” (T). Esta hábito o fez desenvolver uma “Lógica do Talvez”, onde as operações de conjunção (“e” lógico, cujo símbolo é &) e disjunção (“ou” lógico, cujo símbolo é |) funcionam normalmente para os valores lógicos verdadeiro e falso, e se comportam da seguinte maneira no que diz respeito ao talvez:

- talvez e falso é falso; talvez e verdadeiro é talvez; talvez e talvez é talvez;
- talvez ou falso é talvez; talvez ou verdadeiro é verdadeiro; talvez ou talvez é talvez.

Além disso, em uma expressão lógica, a conjunção tem precedência sobre a disjunção.

Este mecanismo de raciocínio confunde os amigos de Onofre, que têm dificuldades para seguir esta lógica. Auxilie os amigos, escrevendo um programa que receba uma expressão válida, composta pelos valores lógicos F, T e V e os pelos operadores & e |, e que determine o seu resultado.

Entrada

A primeira linha da entrada contém o valor do inteiro N ($1 \leq N \leq 2 \times 10^5$), que indica o número de elementos da expressão.

A segunda linha da entrada contém $2N - 1$ elementos, separados por um espaço em branco, dispostos da seguinte forma: $a_1 \ b_1 \ a_2 \ b_2 \ \dots \ b_{N-1} \ a_N$, onde $a_i \in \{\text{F}, \text{T}, \text{V}\}$, para todo $i \in [1, N]$, e $b_j \in \{\&, |\}$, para todo $j \in [1, N - 1]$.

Saída

Imprima, em uma linha, o resultado da expressão.

Exemplo

Entrada	Saída
3 V T & F	V
4 T F & V T	T
2 F & T	F

Notas

No primeiro caso, usando a precedência da conjunção sobre a disjunção, vale que $V \mid T \& F = V \mid (T \& F) = V \mid F = V$.

No segundo caso, $T \mid F \& V \mid T = T \mid (F \& V) \mid T = T \mid F \mid T = (T \mid F) \mid T = T \mid T = T$.

Problema M

Make

Limite de tempo: 1s

Autor: Daniel Saad

O utilitário GNU/Make automatiza todo o processo de construção de artefatos computacionais em sistemas *UNIX like*. Ele consegue resolver dependências e determinar a ordem com a qual as regras devem ser executadas para que os pré-requisitos de outras regras sejam satisfeitos. É claro que, se houver uma dependência circular entre as regras, o utilitário não conseguirá resolvê-la, apesar de informar ao usuário que existe essa dependência.

As regras que serão resolvidas pelo GNU/Make devem ser escritas em um arquivo, chamado *Makefile*, o qual possui a seguinte sintaxe:

```
nome_da_regra: pre_req_1 pre_req_2 ... pre_req_n
```

Isto é, cada regra tem um nome, também chamado objetivo, e é separada dos pré-requisitos por um sinal de dois-pontos. Em seguida, os pré-requisitos são separados por um espaço. Cada regra só pode ser executada se os pré-requisitos foram cumpridos, isto é, se outras regras, que tinham como objetivos os pré-requisitos da regra a ser executada, foram cumpridas.

Dado um arquivo Makefile, determine, caso possível, uma ordem na qual as regras devem ser cumpridas.

Entrada

A primeira linha da entrada possui um inteiro n ($1 \leq n \leq 10^3$), que indica o número de objetivos.

As próximas n linhas descrevem uma regra conforme o formato exposto no enunciado.

É garantido que, para cada pré-requisito, existe uma regra com o mesmo objetivo, e que todos os nomes de objetivos estão limitados a 20 caracteres sobre o alfabeto das letras minúsculas.

Saída

Imprima uma linha com uma possível ordem de cumprimento dos objetivos. Os objetivos devem estar separados por um espaço. No caso de uma dependência circular entre as regras, imprima “impossible”. Se existir mais de uma possível ordem para satisfazer as regras, qualquer uma pode ser impressa.

Exemplo

Entrada	Saída
3 program: depa depa: depb depb:	depb depa program
3 a: b c b: c:	b c a
3 a: b b: c c: a	impossivel

Notas

No primeiro exemplo, a única possibilidade é satisfazer a regra com objetivo **depb** primeiro para então satisfazer a regra com o objetivo **depa** e finalmente satisfazer a regra com objetivo **program**.

No segundo exemplo, uma possível resposta é “**b c a**”. Outra possível resposta é “**c b a**”.

No terceiro exemplo, existe uma dependência circular.