

**SEMANA DE
INFORMÁTICA**

Maratona de Programação

Caderno de Problemas

UFV, 8 de Novembro de 2020

Instruções:

- Este caderno contém 16 problemas: com páginas numeradas de 1 a 20, não contando esta página de rosto.
- Em todos os problemas, a entrada deve ser lida da *entrada padrão* e a saída deve ser escrita na *saída padrão*.



Universidade Federal de Viçosa

Informações gerais

- A entrada deve ser lida da entrada padrão e a saída escrita na saída padrão
- Todas as linhas, inclusive a última, tanto da entrada quanto da saída, devem ter fim-de-linha
- Sempre que uma linha contém vários valores, eles são separados por um único espaço em branco e nenhum outro espaço deve aparecer (nem linha em branco), tanto na entrada quanto na saída
- O alfabeto inglês é sempre usado, não deve haver acentos, cedilhas, etc, nem na entrada nem na saída

Tempos limites de execução

- Capivid-20 – 0,2s
- Transportando Doce de Leite na Nlogônia – 8 s
- Nas demais – 1 s
- ATENÇÃO! Os tempos foram calibrados para C++ e são os mesmos para todas as linguagens; não há garantia que códigos em outras linguagens executem dentro destes tempos.

"Esta é uma obra de ficção, qualquer semelhança com nomes, pessoas, fatos ou situações da vida real terá sido mera coincidência."

Fontes das figuras ilustrativas usadas nos problemas da prova:

- Senha Super Secreta: https://twitter.com/math_cleon/status/1288607176680976388
- VOIP: adaptada de <https://valkirias.com.br/enfim-capivaras/>
- Capivid-20: https://twitter.com/capybara_man/status/1287035713373052929
- Distanciamento capivaral: <https://www.dinheirorural.com.br/secao/agronegocios/bicho-bom>
- Eu sabia essa com maçãs!: <https://twitter.com/descomplica/status/520725505708802048>
- Por que a Capivara atravessou a lagoa?: Figura da capivara: <http://pt.wikifur.com/wiki/Capivara>, doce de leite: <https://armazemseuluiz.com.br/doce-de-leite-vicosa-800g>
- Por que a Capivara atravessou a lagoa de novo?: <https://primeiroasaber.com.br/2020/06/08/jacare-e-flagrado-na-lagoa-da-ufv/>, de Krisley Freitas
- Capivaras querem doce (de novo)!: Foto extraída do desenho Pica-Pau
- The OC: Uma capivara na mata do Paraíso: https://www.reddit.com/r/aww/comments/fqzw91/capybara_paradise/
- Balanceamento de capivaras nas lagoas: <https://www.pinterest.com/pin/643733340456922784/>
- Amigas: <https://www.warnerbros.it/serietv/genere-commedia/friends/>
- Aterrando a lagoa: Figura da capivara: <http://pt.wikifur.com/wiki/Capivara>, doce de leite: <https://armazemseuluiz.com.br/doce-de-leite-vicosa-800g>
- Classificação de textos de maratonas: adaptada de <https://tvefamosos.uol.com.br/noticias/redacao/2017/03/10/globo-divulga-estreia-de-senhora-do-destino-com-memes-de-nazare-confusa.htm>
- Capivara aravipac está presente: Arquivo pessoal
- Capivarim: <https://conexaoplaneta.com.br/blog/o-dia-do-fotografo-e-da-capivara/a-onca-e-capivara-800/>

Problema A. Senha Super Secreta

Arquivo-fonte: "senha.x", onde x deve ser c, cpp, java ou py

Cansada de ter suas senhas descobertas, sua tia Capivarista te ofereceu R\$200 para criar uma senha super secreta.

Você sabe que as senhas dela eram facilmente descobertas porque eram sempre a idade de um de seus dois filhos ou uma das seguintes contas simples: a soma ou a diferença entre essas idades.

Uma senha super secreta para ela é qualquer valor entre 0 e 100 que não seja nenhum dos mencionados. Mas ela não é capaz de criar uma senha sozinha, então você, como bom sobrinho, aceitou o trabalho.



Entrada

A entrada contém dois números inteiros A e B que são as idades dos dois filhos de Capivarista ($1 \leq A, B \leq 100$).

Saída

Escreva uma senha super secreta. Se houver mais de uma resposta, escreva qualquer uma delas.

Exemplos

Entrada	Saída
3 5	10
14 21	30

Problema B. VOIP

Arquivo-fonte: "voip.x", onde x deve ser c, cpp, java ou py

Sabe-se que capivaras se comunicam por VOIP, enviando mensagens de uma estação para outra, seguindo o curso normal das estações (Verão, Outono, Inverno, Primavera), daí o nome do sistema de comunicação.

Sua tarefa aqui é determinar quanto tempo demora para uma mensagem chegar de uma estação à outra. Por exemplo, uma mensagem enviada no Verão para o Outono demora apenas 1 estação, enquanto uma mensagem enviada no Verão para a Primavera demora 3 estações.

As estações se repetem na mesma ordem ano após ano, então uma mensagem enviada na Primavera para o Verão demora apenas 1 estação.

Obs.: uma mensagem enviada de uma estação para a mesma estação só chega no ano seguinte.

Entrada

A entrada contém apenas dois caracteres, A e B, indicando que uma mensagem é enviada da estação A para a estação B.

Tanto A quanto B estão entre os possíveis no sistema VOIP: V, O, I, P.

Saída

Escreva quantas estações demora para a mensagem chegar da estação de origem à estação de destino.

Exemplos

Entrada	Saída
V O	1
I P	1
O V	3
V V	4



Problema C. Capivid-20

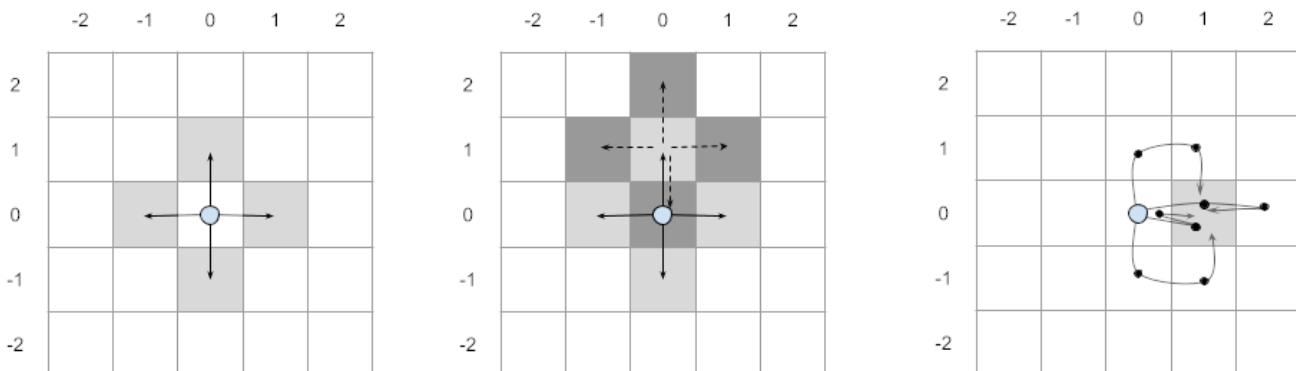
Arquivo-fonte: "capivid.x", onde x deve ser c, cpp, java ou py

As capivaras da UFV são constantemente mordidas por cães, que disputam com elas o espaço da "curva da reta". Elas não têm maiores problemas de saúde após estas mordidas porque são vacinadas contra raiva.

Neste ano, entretanto, uma delas foi mordida por um jacaré e teve sérios problemas de saúde nas horas seguintes. Apesar das análises, pesquisadores da UFV descobriram que ela foi infectada por um novo vírus, batizado de CAPIVID-20.

Antes de ser diagnosticada, a capivara passeou pelo campus e pode ter infectado outras capivaras. Por isto, a Divisão de Saúde entrevistou várias delas para saber onde estiveram nas últimas horas. Mas a capivara diagnosticada com CAPIVID-20 está internada com respirador mecânico e não consegue contar por onde andou após a mordida.

A Divisão de Saúde estima que, após ser mordida na lagoa, a capivara caminhou aleatoriamente pelo campus, parando para descansar em cada hora. O Campus foi dividido em células, como na figura abaixo, sendo a lagoa onde a capivara foi mordida identificada pela célula na coordenada (0,0). A capivara pode ter caminhado em qualquer direção ortogonal (norte, sul, leste, oeste), então na hora 1 ele esteve em alguma das células marcadas na figura da esquerda com igual probabilidade (25% de chance em cada).



Em seguida, ela caminhou novamente de onde estava em qualquer direção ortogonal para alguma célula adjacente e descansou na hora 2. Por exemplo, se na hora 1 ela estava na célula de coordenada (0,1), na hora 2 ela pode ter descansado em alguma das células marcadas de cinza escuro na figura do meio. Note que ela pode inclusive ter voltado para a lagoa.

De forma geral, se a capivara com CAPIVID-20 estava na célula (i, j) na hora t , na hora $t + 1$ ela estava em uma das células $(i - 1, j)$, $(i, j - 1)$, $(i + 1, j)$, $(i, j + 1)$, com igual probabilidade em cada uma delas. Assim, na hora 3, a capivara pode ter estado na célula $(1, 0)$, sendo possível ter chegado nela de 4 formas diferentes (figura da direita).

Para cada capivara entrevistada, calcule a probabilidade de que ela tenha estado ao mesmo tempo na mesma célula que a infectada com CAPIVID-20.

Entrada

A entrada começa com uma linha contendo um inteiro N indicando o número de capivaras entrevistadas ($1 \leq N \leq 100$). Em seguida existem N linhas com as informações dadas por cada capivara. Cada uma destas linhas começa com um inteiro T , indicando a quantidade de informações dadas pela capivara ($1 \leq T \leq 3$), seguido por T triplas de inteiros H, X, Y indicando respectivamente o horário e as coordenadas da célula visitada ($1 \leq H \leq 10, -10 \leq X, Y \leq 10$).

Saída

Escreva N linhas na saída, informando a probabilidade de cada capivara entrevistada ter estado na mesma célula no mesmo horário da capivara com CAPIVID-20. Cada valor deve ser escrito com 3 casas decimais.

Exemplos

Entrada	Saída
3	0.000
2 1 0 2 2 0 1	0.125
2 1 1 1 2 1 1	0.000
1 1 0 0	

Neste exemplo, a primeira capivara esteve na hora 1 na célula (0, 2) e na hora 2 na célula (0, 1). É impossível que a capivara doente tenha estado nestas células nestes horários. No segundo exemplo a capivara ficou parada na célula (1, 1) nas horas 1 e 2. Existe 12,5% de chance da capivara doente ter estado ali em algum destes horários. A última só se lembra de ter estado na lagoa na hora 1. Neste momento a capivara doente não estava mais ali.

Entrada	Saída
3	0.250
2 1 0 0 2 0 0	0.141
1 3 1 0	0.207
3 2 1 1 3 1 1 4 1 1	

Problema D. Distanciamento capivaral

Arquivo-fonte: “distancia.x”, onde x deve ser c, cpp, java ou py

Devido ao surto de CAPIVID-20, as capivaras da UFV precisam respeitar um distanciamento capivaral, que restringe a quantidade de capivaras por m^2 em qualquer ambiente.

Assim, existe um limite no número de capivaras que podem ficar simultaneamente dentro da lagoa. Quando o limite é atingido, as capivaras que desejam entrar na lagoa precisam esperar na margem até que outras decidam sair da lagoa.

Como relaxar na lagoa é o passatempo preferido das capivaras, esta regra acabou tendo um efeito indesejado: o número de roedores esperando às margens da lagoa pode ser muito grande, não respeitando o distanciamento capivaral.

Então, foi feita mais uma regra: há um limite no número de capivaras por m que podem aguardar na margem da lagoa.

Verifique se é possível respeitar as regras.

Obs: capivara sozinha na lagoa sempre respeita a primeira regra, e capivara sozinha na margem sempre respeita a segunda regra.



Figura 1: Capivaras desrespeitando distanciamento

Entrada

A entrada começa com uma linha contendo dois números reais, C e L , representando o comprimento e a largura da lagoa (considere a lagoa retangular). $0 < C, L \leq 1000$.

Em seguida outra linha com dois números reais D e F , indicando respectivamente o máximo de capivaras por metro quadrado dentro da lagoa e o número máximo de capivaras por metro nas margens (o perímetro da lagoa). $0 < D, F \leq 3$.

Por fim, uma terceira linha com um número inteiro N , representando o número de capivaras em um dia de sol na UFV. $N \leq 1.000.000$.

Saída

Escreva uma linha contendo “YES”, se for possível que todas as N capivaras sejam distribuídas na lagoa e na margem, respeitando tanto o limite por metro quadrado dentro da lagoa quanto o limite por metro nas margens, ou “NO” se não for possível respeitar simultaneamente as duas regras.

Exemplos

Entrada	Saída
4.0 4.0 0.3 0.2 6	YES

Neste exemplo, a lagoa tem tamanho 4.0×4.0 e podem coexistir 0.3 capivaras por m^2 dentro dela e 0.2 capivaras por m em seu perímetro. As 6 capivaras podem estar dispostas, por exemplo, 4 dentro da lagoa e 2 fora, sendo ambas regras respeitadas.

Entrada	Saída
4.0 4.0 0.2 0.2 7	NO
Entrada	Saída
10.0 4.0 0.3 0.2 15	YES
Entrada	Saída
2.0 4.0 0.2 0.3 5	NO

Problema E. Eu sabia essa com maçãs!

Arquivo-fonte: “fatorial.x”, onde x deve ser c, cpp, java ou py

Considere a equação $n! = a! \times b!$. Para $n, a, b > 1$, até hoje só se conhecem soluções onde n é 10.

Uma professora pediu Capivaristo, um capivaro que estuda no Coluni, para verificar se 3 números satisfazem essa equação. Infelizmente ele não está conseguindo fazer contas tão complicadas.

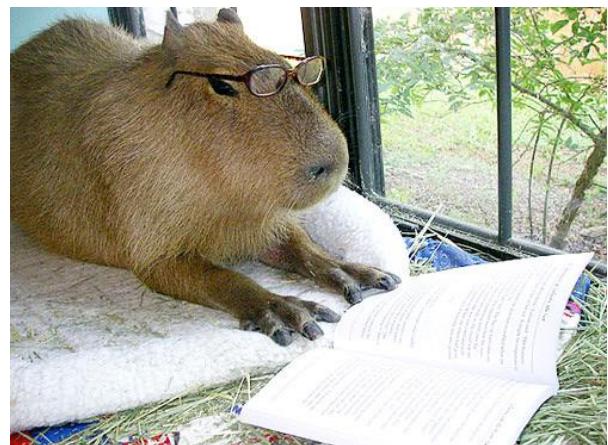
Dados 3 números n, a e b , sua tarefa será verificar se tais números satisfazem a equação ou não, ajudando Capivaristo a resolver seu dever de casa.

Entrada

A entrada contém 3 números inteiros: n, a, b .
($0 \leq n, a, b \leq 2.000.000.000$)

Saída

Escreva uma linha contendo “YES”, se a entrada satisfazer a equação, ou “NO”, caso contrário.



Exemplos

Entrada	Saída
1 1 1	YES
1 2 3	NO

Problema F. Por que a Capivara atravessou a lagoa?

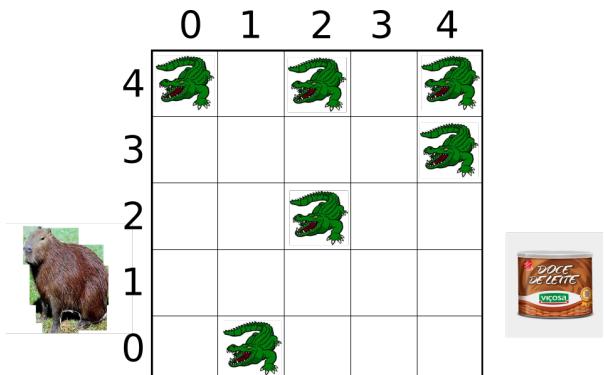
Arquivo-fonte: “atravessou.*x*”, onde *x* deve ser c, cpp, java ou py

Como todos já devem saber, apareceu um jacaré na lagoa da UFV. Já era difícil dar mais coisa errada em 2020, mas agora apareceram ainda mais desses animais!

Uma capivara estava em um dos lagos da UFV (um lago quadrado) e avistou uma lata de doce-de-leite do outro lado da lagoa. Sua idéia seria nadar até o outro lado para comer da iguaria. Porém, há vários jacarés dormindo na água (por algum motivo ainda não descoberto pelos pesquisadores, eles dormem apenas em coordenadas inteiras da lagoa).

Assim, ela contratou a No Bugs para desenvolver um software capaz de ajudá-la a decidir como a lagoa deve ser atravessada (a capivara só sabe nadar em linha reta e não teve a brilhante ideia de contornar a lagoa por terra).

Dadas as coordenadas (x, y) dos jacarés, sua tarefa é encontrar uma coordenada inteira y na qual a capivara pode nadar com segurança em linha reta até o outro lado da lagoa. No exemplo ao lado, ela poderia nadar apenas na coordenada $y = 1$.



Entrada

A entrada começa com uma linha contendo um inteiro N indicando o número de jacarés na lagoa ($0 \leq N \leq 1.000.000$). A seguir, há um número L indicando o tamanho do lado da lagoa ($1 \leq L \leq 1000$).

Então, há N linhas, sendo que a i -ésima linha contém as coordenadas X_i, Y_i do i -ésimo jacaré ($0 \leq X_i, Y_i < L$).

Saída

Escreva uma linha contendo a coordenada inteira y na qual a capivara pode nadar com segurança em linha reta até o outro lado da lagoa. Se houver múltiplas respostas válidas, escreva a menor delas. Se não houver solução, escreva -1 .

Exemplos

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

Entrada	Saída
5	-1
5	
0 0	
0 1	
0 2	
0 3	
0 4	

Problema G. Por que a Capivara atravessou a lagoa de novo?

Arquivo-fonte: “atravessou2.*x*”, onde *x* deve ser c, cpp, java ou py

As capivaras da questão anterior acharam outra lagoa com varias latas de doce do lado oposto da margem. Elas animaram tanto que chamaram vários amigos para fazerem uma festa com muita grama e doce-de-leite.

Porém, encontraram uma pedra no meio do caminho: nessa lagoa, que fica perto do Multiuso/Reitoria, mora um jacaré muito bravo e atravessá-la com segurança para buscar as latas será um desafio.

Os roedores observaram o seguinte: todo dia às 11 da manhã o jacaré sai para almoçar no RU e gasta K minutos para retornar.

Assim, as capivaras planejaram buscar o doce nesse horário, sendo cada animal responsável por buscar uma lata. Porém, outro desafio surgiu: se muitas (mais de duas) capivaras estiverem dentro da lagoa simultaneamente, os cachorros caramelos que vivem na UFV vão começar a latir muito, e isso poderá chamar a atenção do jacaré!

Dados os tempos em que cada capivara gasta para entrar na lagoa e buscar sua lata de doce, determine se será possível que todas consigam buscar a iguaria de forma segura, ou seja, antes que o jacaré volte.

Obs: se uma capivara entra na lagoa exatamente no momento em que outra sai, deve-se considerar que nesse momento haverá apenas uma delas dentro d'água. De forma similar, se as capivaras saírem da água exatamente na hora que o jacaré voltar, elas não estarão ameaçadas.

Entrada

A entrada começa com o número de casos de teste (entre 1 e 30).

Cada caso de teste começa com dois inteiros: o número de capivaras N e o tempo K (em minutos) que o jacaré gasta para almoçar ($1 \leq N \leq 100, 0 \leq K \leq 2.000.000$). Por fim, há N inteiros (entre 0 e 100) cada um descrevendo o tempo (em minutos) que cada capivara gasta para buscar o doce.

Saída

Para cada caso de teste, escreva uma linha contendo “YES” se for possível buscar todas as latas de doce em segurança e “NO”, caso contrário.

Exemplos

Entrada	Saída
<pre>2 2 1 1 3 4 4 3 1 2 2</pre>	<pre>NO YES</pre>

No segundo exemplo, a primeira e a terceira capivara podem entrar na lagoa simultaneamente. Quando a terceira voltar (no tempo 2), a quarta poderá entrar (ela voltará no tempo 4). Quando a primeira voltar (no tempo 3) a segunda poderá entrar (e voltar no tempo 4). Assim, as últimas a saírem da lagoa sairão justamente no momento que o jacaré estiver voltando do almoço. Obs: há outras soluções válidas nesse exemplo.

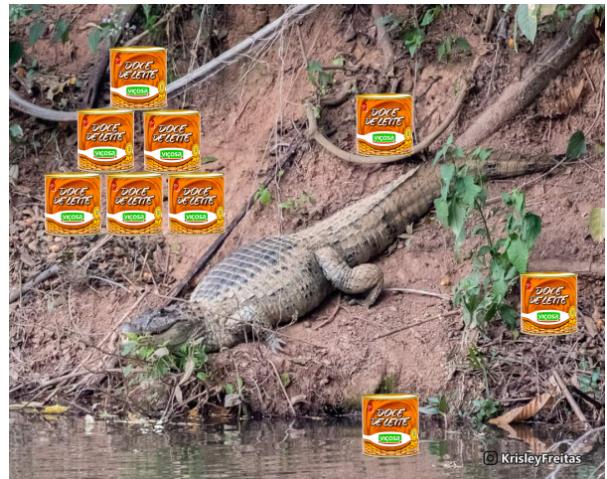


Figura 2: Jacaré da Lagoa da UFV. Adaptação da foto de: Krisley Freitas

Problema H. Capivaras querem doce (de novo)!

Arquivo-fonte: “doce.x”, onde x deve ser c, cpp, java ou py

Duas capivaras da UFV encontraram duas latas de doce de leite na beirada da lagoa (deixadas por alguns estudantes do DPI). Como tinham quantidade diferente de doce, seria injusto uma comer mais doce que outra.

O jacare Zé, apesar de ser o terror das capivaras, apareceu para ajudá-las e propôs o seguinte: “caso o peso de uma lata seja divisível por 3, comerei $\frac{2}{3}$ da lata e deixarei $\frac{1}{3}$ para vocês; se for divisível por 2, comerei metade; e se for divisível por 5, comerei $\frac{4}{5}$, deixando $\frac{1}{5}$ para vocês.”

Sabendo que o Zé é muito espertalhão, as capivaras aceitaram a proposta, mas pediram para ele fazer o proposto acima no número mínimo de passos (elas querem ficar livres dele o mais rápido possível).



Entrada

A entrada começa com uma linha contendo dois inteiros A e B separados por um espaço em branco ($0 \leq A, B \leq 2.000.000.000$). Tais inteiros representam os pesos (em gramas) das duas latas de doce.

Saída

Escreva uma linha contendo o número mínimo de passos para que o peso das duas latas fique igual. Se isso não for possível, escreva -1.

Exemplos

Entrada	Saída
12 2	2
Entrada	Saída
14 8	-1

Problema I. The OC: Uma capivara na mata do Paraíso

Arquivo-fonte: "theoc.x", onde x deve ser c, cpp, java ou py

Como muitos devem saber, a UFV possui uma mata em uma região de Viçosa chamada Paraíso. Tal região é uma reserva de Mata Atlântica e também utilizada para pesquisas e aulas.

Ryan, uma capivara que morava na lagoa da UFV, decidiu se mudar para tal mata com o objetivo de ter uma vida mais pacata de interior e aproveitar melhor a vida nesse local. Após a mudança, se lembrou de uma coisa muito importante: ele havia se matriculado em uma disciplina da UFV (Organização de Computadores) e teria sua última prova amanhã. Ajude Ryan a voltar para UFV e fazer sua prova de OC.

Infelizmente Ryan deverá voltar à pé para a UFV e, nisso, passar por diversas trilhas (de mão única) que ligam o Paraíso à UFV. Trechos das trilhas se conectam em vários pontos. Porém, há um grande problema: vários jacarés vieram para a UFV e muitos habitam tais trilhas. Sendo assim, Ryan deverá lutar contra tais jacarés para conseguir atravessar cada trecho.

Ryan possui K pontos de ataque e V pontos de vida. Ao lutar com um jacaré, esses pontos são reduzidos. Ajude Ryan a chegar à UFV no menor tempo possível sem que seus pontos de ataque e vida restantes fiquem negativos.



Entrada

A entrada começa com uma linha contendo um inteiro N indicando o número de junções de trilhas e um inteiro T indicando o número de trilhas ($1 \leq N \leq 100, 0 \leq T \leq 1000$). A mata do paraíso está na junção 0 e a UFV está na junção $N - 1$. A seguir, há dois inteiros K e V indicando os pontos de ataque e vida que Ryan terá inicialmente ($0 \leq K, V \leq 20$).

A seguir, há T linhas, representando as trilhas, sendo que cada linha i contém quatro inteiros $A_i B_i D_i K_i V_i$ ($0 \leq A_i, B_i < N, A_i \neq B_i; 0 \leq K_i, V_i \leq 20; 0 \leq D_i \leq 1.000.000$). Isso indica que a i -ésima trilha conecta a junção A_i à junção B_i , que gasta-se D_i minutos para atravessá-la e que lutar contra os jacarés presentes nela gastará K_i pontos de ataque e V_i pontos de vida.

Saída

Escreva uma linha contendo 2 inteiros indicando, respectivamente, quanto tempo Ryan gastará, no mínimo para chegar à UFV e qual a quantidade máxima (somada) de vida e ataque restante que sobrará nele (caso decida chegar gastando esse tempo). Se não for possível chegar à UFV considerando as restrições do problema, imprima $-1 - 1$.

Exemplos

Entrada	Saída
<pre>3 3 5 6 0 1 1 3 2 1 2 1 3 2 0 2 5 3 5</pre>	<pre>5 3</pre>

Problema J. Balanceamento de capivaras nas lagoas

Arquivo-fonte: “balanceamento.x”, onde x deve ser c, cpp, java ou py

Devido ao medo do CAPIVID-20, o reitor da UFV decidiu balancear melhor o número de capivaras habitando cada lagoa de modo a evitar superlotações em algumas delas (isso não será tão trabalhoso, já que o jacaré também está ajudando a resolver esse problema).

Para isso, um carro foi contratado para transportar capivaras de uma lagoa para outra. Porém, surgiu um desafio: preocupado com tal transporte, pois poderia espalhar a doença pelo Campus, um professor do Departamento de Veterinária sugeriu que as capivaras sejam transportadas apenas por distâncias bem curtas.

Após um tempo de estudo junto com o Departamento de Agrimensura, os pesquisadores conseguiram determinar entre quais pares de lagoas as capivaras podem ser transportadas sem que haja muito risco de contaminação. Observe que, se uma capivara pode ser transportada da lagoa A para a B em segurança e também é possível transferir capivaras de B para C em segurança, isso não significa que a capivara de A poderia ser transportada para C (a garantia de segurança vale apenas para as capivaras que estavam originalmente na lagoa).

Sua tarefa é determinar se será possível transportar as capivaras de modo a atingir exatamente o balanceamento desejado.

Entrada

A primeira linha da entrada começa com a quantidade de lagoas L ($0 \leq L \leq 100$) e a quantidade de transportes viáveis entre as lagoas T ($0 \leq T \leq 300$).

A seguir, há L inteiros (cada um entre 0 e 100) representando quantas capivaras atualmente há em cada lagoa. Na próxima linha, há mais L inteiros (entre 0 e 100) indicando a quantidade desejada de capivaras nas lagoas após o rebalanceamento.

Então, há T linhas indicando as possibilidades de transportes que podem ser feitos com segurança. Cada uma dessas linhas contém dois inteiros A, B ($1 \leq A, B \leq L, A \neq B$) indicando que o transporte entre as lagoas A e B (e vice-versa) pode ser feito em segurança. Não haverá repetição nos pares AB .

Saída

Escreva uma linha contendo “YES” se for possível fazer o rebalanceamento de forma segura e “NO” caso contrário.

Se a saída for “YES”, escreva em seguida L linhas, cada uma contendo L inteiros. O inteiro na linha i , coluna j indicará quantas capivaras deverão sair da lagoa i e serem transportadas para a lagoa j . Os da linha i , coluna i indicarão quantas capivaras permanecerão na lagoa i . Se houver mais de uma saída possível, escreva qualquer uma delas.

Exemplos

Entrada	Saída
<pre>3 1 1 1 2 1 2 1 1 2</pre>	NO

Entrada	Saída
<pre>3 2 1 1 2 1 2 1 1 2 2 3</pre>	<pre>YES 1 0 0 0 1 0 0 1 1</pre>



Figura 3: Carro contratado pela UFV

Problema K. Amigas

Arquivo-fonte: “amigas.*x*”, onde *x* deve ser c, cpp, java ou py

Como todos sabem na UFV há várias capivaras. Elas vivem em diferentes locais do Campus e todo dia de manhã caminham para fazer seus importantes compromissos. Além de comer muita grama e embelezar o campus, há alguns locais que tais capivaras frequentam: um grupo de amigos adora tomar café na cafeteria da BBT, uma delas precisa ir todo dia trabalhar em um Museu (como paleontóloga), outra é ator, etc.

Uma característica peculiar de tais roedores é que eles não possuem uma memória tão boa: conseguem ir aos compromissos apenas se precisarem passar por no máximo duas ruas nos trajetos (e sempre usam o menor caminho possível). Felizmente, como todos sabem, o Campus da UFV forma uma grade (onde a distância entre cada par de cruzamentos vizinhos é sempre a mesma) e, assim, é muito fácil fazer tais trajetos.

Porém, há um desafio: devido à constante presença de motoqueiros malucos no Campus, o reitor decidiu transformar (apenas durante o dia) todas vias em mão única (inclusive para pedestres e capivaras) e, com isso, tentar reduzir os problemas de segurança. Note que uma via é composta por uma linha ou coluna (inteira) do mapa. Isso fez com que fique impossível para algumas capivaras irem a alguns lugares fazendo aquele tipo de rota que usa no máximo 2 vias.

Como os roedores são muito queridos na UFV, pediram aos funcionários que proponham mudanças na mão das vias de modo a permitir que todas as capivaras continuem podendo caminhar até seus compromissos. Note que elas precisam apenas chegar aos locais de destino – como sempre voltam à noite (quando as vias voltam a ter mão dupla), a volta nunca será um problema.

Dados os compromissos de várias capivaras, determine se será possível para a UFV orientar as vias de mão única de modo que elas consigam fazê-los.

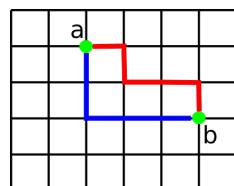


Figura 4: Exemplo com duas rotas entre *a* e *b* em um mapa com 6 linhas e 7 colunas: a rota em vermelho não pode ser feita por uma capivara (passa por 4 vias); já a azul pode, pois é o caminho mínimo (comprimento 5) e passa por no máximo 2 vias.

Entrada

A entrada pode conter múltiplos casos de teste e termina com o final de arquivo.

Cada caso de teste começa com três inteiros L, C, M representando o número de linhas e colunas do mapa da UFV ($1 \leq L, C \leq 50$) e compromissos ($0 \leq M \leq 2000$). A seguir, há M linhas, cada uma representando o trajeto que uma capivara deverá fazer.

Cada trajeto é representado por 4 inteiros $l_0 c_0 l_1 c_1$ ($0 < l_0, l_1 \leq L, 0 < c_0, c_1 \leq C$), sendo que $l_0 c_0$ indicam a linha e a coluna onde a capivara mora e $l_1 c_1$ indicam a linha e a coluna do local para onde a capivara deverá ir pela manhã (no exemplo acima, supondo que a capivara mora no ponto *a* e o ponto 1, 1 está no canto inferior esquerdo da tela, então $l_0 = 5$ e $c_0 = 3$). Note que todas capivaras moram em esquinas e seus trajetos também terminam em esquinas.

Saída

Imprima uma linha contendo “YES” se for possível orientar as ruas de modo que todas capivaras consigam fazer seus trajetos e “NO” caso contrário.

Exemplos

Entrada	Saída
4 4 2 1 1 3 2 3 2 1 1 10 10 3 1 1 1 10 10 1 10 10 10 10 1 1	YES NO

Problema L. Aterrando a lagoa

Arquivo-fonte: "aterrando.x", onde x deve ser c, cpp, java ou py

Conforme informado em outras questões, a UFV possui uma lagoa retangular com jacarés em vários pontos (com coordenadas inteiras).

As capivaras sempre ficam preocupadas em atravessar tal lagoa e, em geral, atravessam apenas em linha reta (elas não conseguem mudar de direção dentro d'água) da esquerda para a direita, somente partindo de uma posição inteira em linhas sem jacarés (chamamos tais linhas de "linhas seguras"). Na figura ao lado, por exemplo, a única opção de travessia segura seria a linha 1.

Para facilitar a vida desses simpáticos roedores, os alunos da UFV decidiram fazer uma vaquinha para aterrinar algumas colunas da lagoa (juntamente com os jacarés que moram em tais colunas). Devido a dificuldades técnicas, o dono da empresa de tratores que foi contratada só consegue aterrinar colunas a partir da borda da lagoa (ou seja, para aterrinar a coluna 2 ele precisa também aterrinar 0 e 1 ou 4 e 3). A empresa também consegue aterrinar ambos os lados: por exemplo, pode aterrinar 0 partindo da esquerda e, a seguir, aterrinar 4 e 3 partindo do outro lado.

Antes de aterrinar a belíssima lagoa, os alunos gostariam de fazer um estudo para descobrir quantas combinações únicas de linhas seguras é possível gerar na lagoa. Sua tarefa será fazer esse cálculo. Na lagoa da figura do exemplo as 9 combinações possíveis são:

- 1 : se nada for aterrado, apenas a linha 1 será segura.
- 1, 3: essa combinação pode ser obtida aterrando com qualquer uma das opções de aterrimento: (4), (4, 3) e (0, 4, 3). (nos próximos exemplos pode haver mais opções de aterrimento que levam a eles – vamos mostrar apenas uma de cada).
- 0, 1: obtido aterrando (0, 1)
- 0, 1, 2: obtido aterrando (0, 1, 2)
- 1, 2, 3: obtida aterrando (2, 3, 4)
- 0, 1, 3: obtido aterrando (0, 1, 4)
- 0, 1, 2, 3: obtido aterrando (1, 2, 3, 4)
- 1, 2, 3, 4: obtido aterrando (0, 2, 3, 4)
- 0, 1, 2, 3, 4: obtido aterrando (0, 1, 2, 4)

Note que, por exemplo, nenhuma possibilidade de aterrimento deixaria apenas as linhas 1, 2 seguras; a única forma de fazer isso seria aterrando apenas a coluna 2, mas isso não é possível sem aterrinar outras colunas.

Como a lagoa pode ter muitas colunas, ao mapeá-las os alunos decidiram usar uma representação mais compacta: cada coluna é codificada nos bits de um número inteiro (exemplo: a coluna 2 ao lado seria codificada no número $10100 = 20$)

Entrada

A entrada começa com uma linha contendo um inteiro L indicando o número de colunas da lagoa ($1 \leq L \leq 100,000$). A seguir, há L números (cada um entre 0 e 1 milhão), cada um representando uma coluna da lagoa codificada em um número binário.

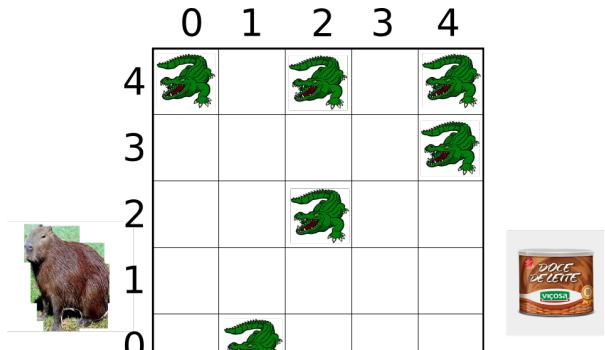
Saída

Escreva uma linha contendo o número de combinações distintas possíveis que poderão ser obtidas.

Exemplos

Entrada	Saída
5 16 1 20 0 24	9

Obs.: este é o exemplo do enunciado.



Problema M. Classificação de textos de maratonas

Arquivo-fonte: "machinelearning.x", onde x deve ser c, cpp, java ou py

Capivarista, um estudante do DPI, sempre participa de competições de programação. Ele possui uma pasta em seu computador onde guarda os PDFs de todos problemas que já resolveu (isso é um troféu para ele!).

Recentemente, Capivarista teve a ideia de separar os problemas que foram resolvidos em competições da Semana de Informática (seus favoritos) para estudá-los novamente. Porém, como ele já resolveu uma quantidade imensa de problemas, decidiu desenvolver um algoritmo de classificação usando *Machine Learning* para realizar tal tarefa de forma automatizada e elegante.

A Figura 5 e as fórmulas abaixo ilustram a estratégia que Capivarista desejava adotar.

Após estudar o problema, o estudante descobriu que precisaria comprar uma máquina de Turing para processar os dados e, como não conseguiu encontrar nenhuma nas lojas de Viçosa, ele pediu ajuda para sua colega Capivarista, que disse: "Deixe de ser bobo, Capivarista! Todo problema da SI tem capivara como tema, problemas de outras competições não!. Basta, então, pegar o texto e procurar por capivara, capivaro, capivarista ou capivaristo no PDF! KISS – Keep It Simple, Stupid!"

Apesar de saber tudo sobre *Machine Learning*, Capivarista não conseguiu pensar em uma forma de pesquisar por algumas palavras em um texto. Ajude-o a implementar um algoritmo que faça isso!

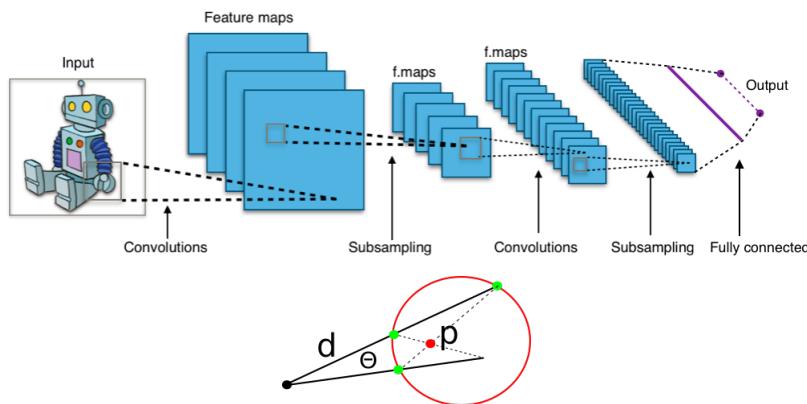
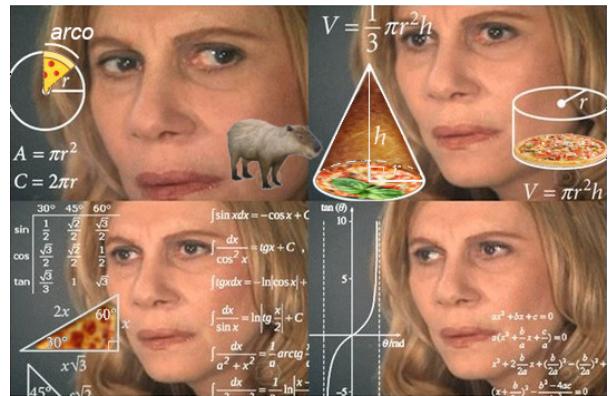


Figura 5: Ideia do algoritmo (fonte da fig. de cima: Wikipedia)

$$\int_V x^3 \pi \log(n) \times \begin{bmatrix} 3.14 & 1 & 1 & 1 & 0 & 0 & 7 \\ 0 & 0 & 1 & 1 & x & 0 & 0 \\ 0 & x_3 & 0 & 1 & 1 & 2 & 0 \\ 0 & x_5 & 0 & 1 & 1 & 0 & 2 \\ 0 & x_2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 5 \\ 1 & 1 & 0 & 0 & 3 & 0 & 9 \end{bmatrix} dx + \int_a^\pi x^3 \pi \log(y) \times \begin{bmatrix} 3.14 & 1 & 1 & 1 & 0 & 7 & 7 \\ 0 & 0 & 1 & 1 & x & 0 & 0 \\ 0 & y_3 & 0 & 1 & 1 & 2 & 0 \\ 0 & y_5 & 0 & 1 & 1 & 0 & 2 \\ 0 & y_2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 5 \\ 1 & 1 & 0 & 0 & 6 & 0 & 9 \end{bmatrix} dy = 2.71\pi \times \prod_{i=1}^{314.15} y_i \times x_i$$

Entrada

A entrada é composta com um texto (sem acentos) extraído do PDF de um problema (o texto termina em final de arquivo). Ela pode conter várias linhas, mas palavras nunca começam em uma linha e terminam em outra. Cada linha terá no máximo 1000 caracteres.

Saída

Escreva "YES" se a entrada contiver pelo menos uma das 4 strings típicas das maratonas da Semana de Informática (capivara, capivaro, capivarista e capivaristo) e "NO", caso contrário.

Exemplos

Entrada	Saída
Capivaristo, um estudante do DPI, sempre participa de competicoes de programacao. Ele possui uma pasta em seu computador onde guarda os PDFs de todos problemas que ja resolveu (isso e um trofeu para ele). Recentemente, Capivaristo teve a ideia de separar os problemas que foram resolvidos em competicoes da Semana de Informatica (seus favoritos) para estuda-los novamente. Porem, como ele ja resolveu uma quantidade imensa de problemas, decidiu desenvolver um algoritmo de classificacao com Machine Learning para realizar tal tarefa.	YES
Entrada	Saída
Mike, um estudante do DPI, sempre participa de competicoes de programacao. Ele possui uma pasta em seu computador onde guarda os PDFs de todos problemas que ja resolveu (isso e um trofeu para ele).	NO
Entrada	Saída
jacarecapivaracarrapato PD da subsequencia	YES
Entrada	Saída
capi vara ca pi va ra	NO

Problema N. Capivara aravipac está presente

Arquivo-fonte: "palindrome.x", onde x deve ser c, cpp, java ou py

A alfabetização das capivaras possui uma característica interessante: algumas delas aprendem a ler normalmente, enquanto outras leem apenas de trás para frente.

A UFV contratou o professor Capivarista para ministrar CAP100. Porém, como não sabem a orientação na qual tal professor lê, decidiram matricular em sua turma apenas alunos cujos nomes sejam palíndromes, isto é, que podem ser lidos tanto da esquerda para a direita como vice-versa (exemplo: Natan). Assim, evitam problemas quando ele for ler a chamada.

Como tais nomes não são muito comuns, decidiram aceitar também nomes “quase palíndromes”, isto é, que “quase” podem ser lidos em qualquer orientação. Mais especificamente, um nome será aceito se no máximo uma das letras diferir da letra que estiver na posição correspondente caso o nome seja lido de forma reversa. Por exemplo, “O mala nada na lama” seria um nome válido, já que apenas a primeira letra difere da correspondente (a última). “Abc cde”, por outro lado, não deveria ser aceito, pois ‘a’ é diferente de ‘e’ e ‘b’ é diferente de ‘d’.

Dado o nome de uma capivara, verifique se ela poderá ser matriculada em CAP100.

Curiosidade que apenas participantes da Maratona SI aprendem: aibofobia é medo de palíndrome. Capivaras com aibofobia deveriam evitar CAP100.

Entrada

A entrada contém apenas uma linha contendo o nome de uma capivara. O nome pode conter letras (sem acentos) e espaços em branco. A linha terá pelo menos uma letra e no máximo 100.000 caracteres.

Saída

Imprima uma linha contendo “YES” se o nome for quase palíndrome e “NO” caso contrário.

Exemplos

Entrada	Saída
Aibofobia	YES
Anotaram a data da maratona	YES
Ane	YES
Oberlan Romao	NO
O mala nada na lama	YES
Jacare Capivaristo	NO



Figura 6: Mesa do RU reservada para alunos de CAP100

Problema O. Transportando Doce de Leite na Nlogônia

Arquivo-fonte: “transporte.x”, onde x deve ser c, cpp, java ou py

A Nlogônia, um país completamente plano, possui uma rede de transporte formada por vários túneis retos e sem qualquer acente/declive (ou seja, paralelos ao plano da Nlogônia) ligando cidades. Algumas cidades, porém, possuem dispositivos de teletransporte que permitem que se possa viajar instantaneamente entre elas. Para se teletransportar entre duas cidades é preciso que as máquinas de teletransportes desses dois locais estejam conectadas por cabos de fibra ótica wireless (FOW).

A figura ao lado ilustra uma rede de transporte vista de cima (túneis se cruzando na figura não se interceptam, já que ficam em “andares” diferentes do solo). Tal rede conecta as cidades a, b, c, d, e, f, g, h (em verde) e é formada pelos túneis (linhas pretas) e conexões FOW (linhas tracejadas em azul). Para viajar de a até b , por exemplo, gasta-se 5 minutos. Como há uma conexão FOW de b para c , pode-se viajar instantaneamente entre as duas cidades.

Para reduzir os tempos de transporte, a engenheira Capivara decidiu construir máquinas de teletransporte no cruzamento entre alguns túneis (nesse caso seria fácil passar os cabos FOW: bastaria cavar na vertical até atingir o outro túnel). Porém, como os sistemas de GPS existentes na Nlogônia só conseguem medir coordenadas inteiras com precisão, só conseguirão instalar as novas máquinas em cruzamentos que ocorrem nesses tipos de coordenadas. Assim, na figura de exemplo foram construídas máquinas no ponto de cruzamento X (de coordenadas inteiras, não mostradas), mas não no cruzamento entre ab e cd (pois as coordenadas de tal cruzamento não são inteiras).

Agora, suponha que o cruzamento de cd e ef no ponto X se dá exatamente no meio deles, para viajar de a para h , o melhor caminho consiste em ir de a até b (gastando 5 minutos), teletransportar de b para c , ir de c até X (gastando 10 minutos – a velocidade nos túneis é sempre constante e, portanto, o tempo é proporcional ao tamanho do trajeto), se teletransportar para o túnel ef pelo ponto X, ir de X até f (gastando 5 minutos) e, finalmente, se teletransportar de f até h . O tempo total seria 20 minutos. (se fizesse o trajeto $abcdfh$ gastaria 25 minutos)

O GPS usado na Nlogônia tem precisão de 0.000001. Assim, se o cruzamento entre dois túneis ocorrer nas coordenadas (3.000009, 5.000005) tais coordenadas seriam consideradas inteiras.

Um caminhão de doce-de-leite está em uma cidade e deseja transportar a preciosa carga para outra cidade da Nlogônia. Devido à curta validade de tal produto, o motorista pediu para você calcular o tempo mínimo necessário para tal percurso.

Entrada

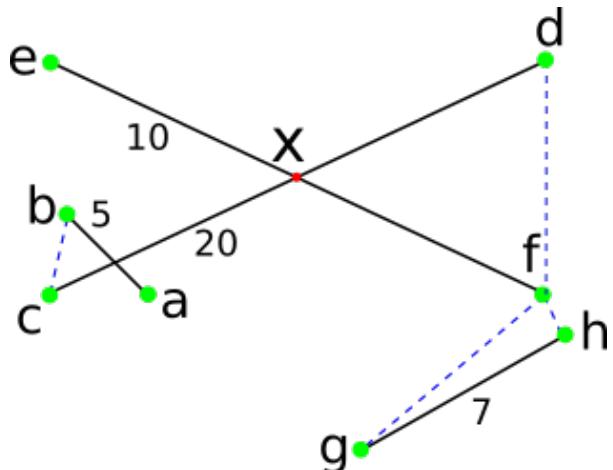
A entrada começa com uma linha contendo um inteiro N indicando o número de túneis ($0 \leq N \leq 20000$). A seguir, há N linhas, cada uma contendo 4 números reais $x_1y_1x_2y_2$ e um inteiro t . Isso representa um túnel conectando as cidades nas coordenadas (x_1, y_1) à cidade nas coordenadas (x_2, y_2) ($0 \leq x_1, y_1, x_2, y_2 \leq 1000$). Levam-se t minutos para percorrer esse túnel completamente. Cada cidade aparece apenas na extremidade de um túnel e as duas extremidades de um túnel nunca possuem as mesmas coordenadas. Assuma que os identificadores das cidades são gerados com base na ordem em que elas aparecem na entrada (ou seja, a cidade 0 é a primeira extremidade do primeiro túnel, a cidade 1 a segunda extremidade, a cidade 2 a primeira extremidade do segundo túnel e assim por diante).

Então, há um número F ($0 \leq F \leq 1000000$) indicando o número de conexões FOW inicialmente existentes entre as cidades. Em seguida há F linhas, cada uma contendo dois inteiros a e b ($0 \leq a, b < 2N$) indicando que há uma conexão FOW entre as cidades a e b .

Por fim, há dois inteiros C e D ($0 \leq C, D < 2N$) e seu objetivo é calcular o tempo mínimo de percurso entre as cidades C e D .

Saída

Escreva uma linha contendo o tempo mínimo, em minutos, necessário para ir de C até D . A saída deve ser impressa com 2 casas de precisão. Se não houver rota entre as duas cidades, escreva -1.



Exemplos

Entrada	Saída
4 12 10 11 13 5 10 10 30 30 20 10 30 30 10 10 22 2 31 8 7 4 1 2 3 5 5 6 5 7 0 7	20.00
4 12 10 11 13 5 10 10 30 30 20 10 30 30 10 20 22 2 31 8 7 2 1 2 3 5 0 7	-1

Problema P. 水豚语

Arquivo-fonte: "capivarim.x", onde x deve ser c, cpp, java ou py

Capivarim (水豚语, em capivarim) é o idioma das capivaras. Note que capivarim é semelhante ao mandarim. Assim como o mandarim, aproximadamente 5 em 6 capivaras não conseguem ler capivarim. Então foi criado o *capim*, um sistema de transliteração que mapeia o sistema de escrita do capivarim em caracteres do alfabeto português, o que facilitou muito a entrada das capivaras na UFV.

Na escrita em capim, todos os caracteres do capivarim são escritos como uma consoante seguida de uma vogal, e um acento sobre a vogal indicando um dos 4 tons do capivarim. A tabela abaixo lista alguns exemplos.

Capivarim	Capim	Português
媽	mã	mãe
麻	má	grama
馬	mâ	doguinho caramelô
罵	mà	maldito

Quem nunca ouviu uma capivara desesperada gritando 媽馬罵麻!? Usando capim é bem mais fácil ler esta frase, que significa "Mãe, doguinho caramelô maldito na grama!".

Assim, em capim, com uma consoante e uma vogal, podem ser representadas 4 palavras diferentes de um caractere, como mostra a tabela para 'm' e 'a'.

Já com uma consoante (m) e duas vogais (a, e) podem ser feitas 8 palavras: mā, má, mâ, mà, mē, mé, mē, mè.

Em capivarim também existem palavras com mais de um caracter. Com uma consoante e uma vogal podem ser feitas 16 palavras em capim de dois caracteres capivarim. Para (m) e (a), por exemplo, existem māmā, māmá, māmâ, mâmâ, ..., mâmâ.

Assim, para uma consoante e uma vogal, existe um total de 20 palavras em capim com até 2 caracteres capivarim (são 4 de 1 caractere e 16 de 2). Para uma consoante e duas vogais existem 72 palavras em capim com até 2 caracteres capivarim.

Obs.: Note que não existe em capim a palavra 'mãe', pois todo caracter capivarim é transliterado em capim sempre com uma consoante seguida de uma vogal com tom, ou sequência de consoante + vogal com tom.

Entrada

A entrada tem uma única linha, contendo três inteiros C , A , P , que indicam respectivamente o número de consoantes, vogais e máximo de caracteres de uma palavra em capim ($1 \leq C \leq 21, 1 \leq A \leq 5, 1 \leq P \leq 20$).

Saída

Escreva uma linha contendo o número de palavras possíveis em capim com os dados da entrada. Como o número de palavras pode ser muito grande, escreva o resultado *mod* 1.000.000.007.

Exemplos

Entrada	Saída
1 1 1	4
Entrada	Saída
1 2 2	72
Entrada	Saída
3 2 3	14424
Entrada	Saída
10 5 5	608037953

