



Centro Universitário

II Maratona de Programação IESB



Escrito e preparado por Edson Alves, Rodrigo Guimarães, Vinícius Costa, João Pedro Ferraz, Daniel Saad Nogueira Nunes, Matheus Faria, Felipe Duerno e Patricia Moscariello

A Conta redonda

Limite de Tempo: 2s

O garçom que atendeu uma família durante uma almoço de domingo ficou surpreso com o valor da conta: R\$ 250,00! O espanto não se deu pelo valor em si, mas pelo fato de que a família consumiu R\$ 227,28 e, após o acréscimo dos 10%, a conta fechou em um valor “redondo”, isto é, com as casas dos centavos zeradas.

O estabelecimento computa os 10% da seguinte maneira: uma vez que o valor consumido esteja escrito na forma canônica (isto é, com exatamente duas casas decimais representando os centavos), apaga-se o último dígito do valor e move-se a vírgula uma unidade para a esquerda. No exemplo dado, temos

$$\text{R\$ } 227,28 \longrightarrow \text{R\$ } 22,72$$

Dado o valor de uma conta redonda, em reais, determine, se possível, o valor do consumo e a taxa de 10% cobrada pelo estabelecimento.

Entrada

A entrada consiste em uma série de T ($1 \leq T \leq 1.000$) casos de teste, onde o valor de T é dado na primeira linha da entrada.

Cada caso de teste é representado por uma única linha, contendo o valor da conta redonda R ($1 \leq R \leq 10^4$), em reais.

Saída

Para cada caso de teste imprima, em uma linha, a mensagem “Caso # t : R\$ C + R\$ D ”, onde t é o número do caso de teste (cuja contagem tem início com o número um) e C e D são os valores do consumo e dos 10%, respectivamente, em forma canônica. Caso não seja possível determinar tais valores, imprima a mensagem “Caso # t : impossível”.

Exemplos de entradas	Exemplos de saídas
6	Caso #1: R\$ 0,91 + R\$ 0,09
1	Caso #2: R\$ 10,00 + R\$ 1,00
11	Caso #3: R\$ 27,28 + R\$ 2,72
30	Caso #4: R\$ 90,91 + R\$ 9,09
100	Caso #5: R\$ 227,28 + R\$ 22,72
250	Caso #6: impossível
8084	

B Gamification

Limite de Tempo: 1s

O processo de gamificação de um disciplina agradou os alunos, mas os levou a uma certa confusão: as notas foram substituídas por pontos de experiência, a serem acumulados em duas fases distintas. O resultado final do aluno na disciplina seria dado em função do valor X_P definido abaixo,

$$X_P = \left\lfloor \frac{F_1 + F_2}{2} \right\rfloor,$$

onde F_i é a quantidade de pontos de experiência acumulados na fase i e $\lfloor x \rfloor$ é o maior inteiro menor ou igual a x .

Dados os valores de F_1 e F_2 , determine o X_P correspondente.

Entrada

A entrada consiste em T ($1 \leq T \leq 100$) casos de teste, cujo valor de T é dado na primeira linha da entrada. Cada caso de teste é representado por uma única linha, contendo os valores dos inteiros F_1 e F_2 ($0 \leq F_i < 2^{64}$, $i = 1, 2$), separados por um espaço em branco.

Saída

Para cada caso de teste imprima, em uma linha, a mensagem “Caso t : X_P ”, onde t é o número do caso de teste (cuja contagem tem início com o número um).

Exemplos de entradas	Exemplos de saídas
4	Caso 1: 5
10 0	Caso 2: 4
3 5	Caso 3: 7
7 7	Caso 4: 237
198 277	

C Números de Fermat

Limite de Tempo: 3s

Pierre de Fermat conjecturou que os números da forma

$$F_n = 2^{2^n} + 1$$

seriam primos para qualquer natural n . Embora os cinco primeiros números F_0, F_1, F_2, F_3, F_4 sejam primos, Leonhard Euler mostrou, em 1732, que

$$F_5 = 2^{2^5} + 1 = 2^{32} + 1 = 4294967297 = 641 \times 6700417,$$

provando que a conjectura era falsa. Não há, até hoje, nenhum outro número de Fermat primo conhecido para $n > 4$.

Dados dois naturais n e m , determine o maior divisor comum entre F_n e F_m .

Entrada

A entrada consiste em T ($1 \leq T \leq 10$) casos de teste, cujo valor de T é dado na primeira linha da entrada. Cada caso de teste é representado por uma única linha, contendo os valores dos inteiros n e m ($0 \leq n, m \leq 10^6$), separados por um espaço em branco.

Saída

Para cada caso de teste imprima, em uma linha, a mensagem “Caso #: d ”, onde t é o número do caso de teste (cuja contagem tem início com o número um) e d é o maior divisor comum entre F_n e F_m . Como o valor de d pode ser muito grande, imprima o resto de sua divisão por $10^9 + 7$.

Exemplos de entradas	Exemplos de saídas
2	Caso 1: 17
2 2	Caso 2: 1
4 5	

D Descendentes de JoH'el

Limite de Tempo: 3s

Há milênios atrás, no planeta Cryptonia, existiu uma grande figura da civilização cryptoniana, JoH'El Gl'mR. O sobrenome dos habitantes de Cryptonia é formado segundo a tradição de seus antepassados:

1. se os pais tem sobrenomes que não contém aspa simples, o sobrenome do filho será a junção do sobrenome do pai e do sobrenome da mãe, separados por uma aspa simples;
2. se ambos pais tem sobrenomes que contém uma aspa simples e a criança é do sexo masculino, o sobrenome dela será formado pela palavra que antecede a aspa, no nome do pai, uma aspa simples, e a palavra que sucede a aspa, no nome da mãe;
3. se ambos pais tem sobrenomes que contém uma aspa simples e a criança é do sexo feminino, o sobrenome dela será formado pela palavra que sucede a aspa, no nome do pai, uma aspa simples, e a palavra que antecede a aspa, no nome da mãe;
4. se um dos pais tem nome com aspa e o outro nome, o sobrenome da criança será o sobrenome, dentre os dois, que contém aspa.

Nos tempos atuais descobriu-se um problema de fertilidade no planeta. Como o tratamento é lento e oneroso, os casais que desejassem ter filhos deveriam entrar em uma fila. Cada casal da fila seria tratado, um por vez, até que nascesse o seu filho único, e o sexo da criança seria escolhido alternadamente, sendo que o primeiro casal da fila teria um menino, o próximo uma menina, o seguinte um menino, e assim por diante. Este processo atrasaria, ao máximo, o fim inevitável da população.

Com este panorama, o Conselho Supremo de Cryptonia tomou três importantes decisões: homenagear JoH'El e batizar o último descendente de Cryptonia com seu primeiro nome; que as crianças oriundas da fila formariam casais, de acordo com a ordem de nascimento, e seriam prometidos em casamento; e que estes casais seriam obrigados, legalmente, a entrar na fila quando atingissem a maioridade.

O plano para tentar manter a civilização viva é mandar o último descendente para o planeta Terra. Visando a confecção de sua nave, é preciso saber de antemão o nome e sobrenome do indivíduo.

Você é um enviado do nosso planeta para Cryptonia a fim de ajudá-los com este problema, criando um programa que, dado os N casais inscritos inicialmente na fila, informe o nome a ser escrito na nave do último descende de JoH'el.

Entrada

A primeira linha da entrada contém o inteiro N ($2 \leq N \leq 2 \times 10^5$), que representa o número de casais inicialmente inscritos na fila. As N linhas seguintes contém, cada uma, um par de sobrenomes P_i e M_i ($1 \leq i \leq N$), separadas por um espaço em branco, onde P_i é o sobrenome do futuro pai e M_i é o sobrenome da futura mãe. Os sobrenomes são compostos por, no máximo, 5 caracteres alfabéticos maiúsculos ou minúsculos, podendo conter ou não uma aspa simples.

Saída

Imprima, em uma linha, o nome e sobrenome do último descendente de Cryptonia.

Exemplos de entradas	Exemplos de saídas
2 gL LkE seLKa Me	JoH'El gL'Me
4 Kl'E1 Lana Jimmy Rut'h Perry Joana Ol'en Lan'g	JoH'El Kl'en
7 SFck jb'U xGjQ Lgd rdox QB a t H wurY RB lZ G E	JoH'El G'wurY

E Contratação de Estagiários

Limite de Tempo: 3s

Uma empresa selecionou R rapazes e M moças para estágio, e agora deve chamá-los para assumir os cargos. Por conta dos trâmites legais, será possível chamar apenas um estagiário por vez.

Como o resultado do processo de seleção era apenas aprovado ou reprovado, sem nenhum critério de ranqueamento, a empresa precisa gerar uma lista com a ordem de chamada dos estagiários. Ela também não quer favorecer nenhum gênero em particular, de modo que a lista deve conter, no máximo, K pessoas de mesmo sexo consecutivamente.

Auxilie a empresa, escrevendo um programa que contabilize o número total de listas válidas que podem ser formadas, respeitando o critério imposto pela empresa.

Entrada

A entrada consiste em T ($1 \leq T \leq 10$) casos de teste, cujo valor está na primeira linha. Cada caso de teste é representado por uma única linha, contendo os valores de R , M e K ($1 \leq R, M, K \leq 100$), separados por um espaço em branco.

Saída

Para cada caso de teste imprima, em uma linha, a mensagem “Caso t : L ”, onde t é o número do caso de teste (cuja contagem tem início com o número um) e L é o número de listas distintas que podem ser formadas, segundo o critério estabelecido pela empresa. Como este número pode ser muito grande, imprima o resto de sua divisão por $10^9 + 7$.

Exemplos de entradas	Exemplos de saídas
5	Caso 1: 2
1 2 1	Caso 2: 8
2 2 1	Caso 3: 84
3 2 2	Caso 4: 11520
3 5 2	Caso 5: 35712
4 4 3	

F ganhando Dinheiro

Limite de Tempo: 3s

Moscariello é um geógrafo e matemático que decidiu dividir o mapa-múndi em 4 quadrantes para formalizar algumas teorias. Uma delas diz que, dada a matriz

$$Q = \begin{bmatrix} q_1 & q_2 \\ q_3 & q_4 \end{bmatrix},$$

que representa quantidade atual de carbono no ar, em moscariellos cúbicos (mc^3), em cada quadrante q_i , é possível determinar a quantidade de dinossauros d_i que existiram em cada quadrante, em unidades de milhares, utilizando a fórmula

$$D = aQQ^{-1} + J,$$

onde a é o coeficiente arqueológico e J é a matriz que contém as constantes jurássicas j_i de cada quadrante.

Dados os valores de a , Q , J , auxilie Moscariello a provar a sua teoria e ganhar bastante dinheiro com petróleo, determinando o quadrante com maior quantidade de dinossauros.

Entrada

A entrada consiste em N ($1 \leq N \leq 1000$) casos de teste, cujo valor está na primeira linha. Cada teste é representado por três linhas: a primeira contém o valor de a ($1 \leq a \leq 100$); a segunda linha contém os valores q_i ($1 \leq q_i \leq 100$), separados por um espaço em branco; e a terceira linha contém os valores j_i ($1 \leq j_i \leq 100$), também separados por um espaço em branco, com $1 \leq i \leq 4$.

Saída

Para cada caso de teste imprima, em uma linha, a mensagem “Caso t : q ”, onde t é o número do caso de teste (cuja contagem tem início com o número um) e q é o quadrante com o maior número de dinossauros. Caso exista mais de um quadrante com o valor máximo, imprima todos eles, em ordem crescente, separados por um espaço em branco. Caso a teoria de Moscariello seja inválida (isto é, não seja possível determinar o número de dinossauros para os valores indicados), imprima a mensagem “Caso t : teoria invalida”.

Exemplos de entradas	Exemplos de saídas
3	Caso #1: 4
5	Caso #2: teoria invalida
1 2 3 4	Caso #3: 1 2 3 4
1 2 3 4	
3	
1 2 1 2	
1 2 1 2	
1	
1 2 2 1	
1 2 2 1	

G Lentorianos

Limite de Tempo: 2s

Os lentorianos são o povo do planeta Javax (em algum momento de sua história, se perdeu o motivo pelo qual eles não se chamam javaxianos). Estas criaturas muito organizadas, definiram áreas de mineração por todo o planeta e distribuíram entre as mineradoras de Javax.

Os lentorianos cavam o terreno com picaretas de péssima qualidade que quebram ao bater em uma pedra, pois só há uma loja de picaretas no planeta (quem diria, certo?!). A loja de picaretas possui um estoque inacabável de picaretas e vende (inclusive fiado) às mineradoras, ao preço de uma jóia.

Este ano várias subáreas de uma área de mineração estão sendo avaliadas e você foi contratado para escrever o software que faz tal avaliação, retornando o lucro resultante de mineração de uma dada subárea (número de jóias encontradas menos o número de picaretas gastas), sabendo que a área de mineração é retangular e que cada uma das posições da área será minerada.

O software também deve ser capaz de fazer simulações, permitindo que o usuário modifique uma posição qualquer do mapa da área de mineração, de modo que ela fique ou vazia, ou com uma pedra, ou com uma jóia.

Deve ser assumido que, no início da mineração, nenhuma mineradora possui picaretas, que cada posição da área será minerada, e que a picareta não quebra ao se minerar ou terra ou uma jóia.

Entrada

A primeira linha da entrada contém dois inteiros N e M ($1 \leq N, M \leq 3000$), separados por um espaço em branco, indicando o número de linhas e colunas da área de mineração. As N linhas seguintes contém, cada uma, as informações sobre a i -ésima linha ($1 \leq i \leq N$) da área de mineração: M símbolos a_{ij} ($1 \leq j \leq M$), que indicam o que há na posição (i, j) : pedra (#), jóia (*) ou terra (.).

A linha seguinte contém um inteiro C ($1 \leq C \leq 20000$), que indica o número de comandos que serão inseridos no software. Os comandos possíveis são:

- 1 $x y$: a posição (x, y) passa a conter uma jóia;
- 2 $x y$: a posição (x, y) passa a conter uma pedra;
- 3 $x y$: a posição (x, y) passa a conter terra;
- 4 $x y z w$: retorna o lucro da mineração da área delimitada por (x, y) (canto superior esquerdo) e (z, w) (canto inferior direito). É assumido que a origem está no ponto $(1, 1)$ (canto superior esquerdo), e a área se estende até (N, M) (canto inferior direito).

Pode-se assumir que $1 \leq x \leq z \leq N$ e que $1 \leq y \leq w \leq M$.

Saída

Para cada comando que retorna o lucro imprima a mensagem “Query q : L ”, onde q é a quantidade de vezes que tal comando foi solicitado até o momento da impressão e L é o lucro obtido com a mineração da área indicada, com as informações contidas no mapa após o processamento de todos os comandos anteriores.

Exemplos de entradas	Exemplos de saídas
4 4	Query 1: 4
*..#	Query 2: 2
**..	Query 3: 3
***#	Query 4: 0
*..#	Query 5: -1
8	
4 1 1 4 4	
1 1 2	
2 1 1	
3 2 1	
4 1 1 4 4	
4 3 1 3 3	
4 1 3 1 3	
4 1 3 2 4	

H Tradução Proteica

Limite de Tempo: 1s

O RNA mensageiros (mRNA) são sequências biológicas compostas das bases nitrogenadas Adenina, Uracila, Citosina e Guanina.

Algumas regiões do mRNA participam do processo de síntese de cadeias de aminoácidos, ou em outras palavras, proteínas. Estas regiões são compostas de códons, sequências de três bases nitrogenadas, e cada códon participa da síntese de um aminoácido. Alguns códons são especiais, pois sinalizam o início e o fim de uma região codificadora. A grosso modo o códon AUG define o início de uma região codante (ou o aminoácido Metionina se ele ocorrer dentro de uma região codante) e os códons UAA, UAG e UGA sinalizam o fim de uma região codante.

Outra questão importante são os quadros de leitura, isto é a forma de enxergar os códons de uma molécula de mRNA. Por exemplo, as triplas do mRNA AGGTGACACCGCAAGCCTTATATTAGC podem ser lidas como:

- AGG TGA CAC CGC AAG CCT TAT ATT AGC
- A GGT GAC ACC GCA AGC CTT ATA TTA GC
- AG GTG ACA CCG CAA GCC TTA TAT TAG C

Faça um programa que identifique a maior cadeia de aminoácidos que poderia ser formada a partir de um mRNA dentre todos os quadros de leitura possíveis. Considere a simplificação de que os códons que identificam o início de uma região codante sintetizam aminoácidos, mas que os códons de parada não sintetizam.

Entrada

A entrada consiste de uma linha contendo uma string S ($6 \leq |S| \leq 10^6$) composta pelos símbolos A, U, C ou G, indicando uma molécula de mRNA. Esta string sempre vai conter ao menos um códon de início e fim.

Saída

A saída é uma linha contendo o tamanho da maior sequência de aminoácidos que pode ser formada.

Exemplos de entradas	Exemplos de saídas
AUGUGA	1
AUGAUGACGACCUAG	4
AUGUGAAUGAUGACGACCUAG	4

I Caminhada em Barcelona

Limite de Tempo: 1s

De férias em Barcelona, Joãozinho resolveu voltar a se exercitar. Mas para não repetir o mesmo trajeto sempre, Joãozinho decidiu calcular quantas possibilidades diferentes de se chegar em um determinado local ele teria, para então variar a sua rota todos os dias.

A cidade de Barcelona, vista do céu, parece um *grid*, onde os quarteirões são quadrados contornados por acessos paralelos aos seus lados, e o ponto (X, Y) é o ponto de interseção entre os acessos dos quarteirões que ficam na coluna X e dos acessos dos quarteirões localizados na linha Y .

No entanto, alguns acessos estão bloqueados devido a uma série de protestos que estão ocorrendo, e Joãozinho não quer passar pelos bloqueios, pois não é possível entrar neles e nem sair por conta da muvuca.

Além disso, Joãozinho sempre parte de um ponto específico, localizado no ponto $(1, 1)$ deste *grid*. Durante a sua caminhada, Joãozinho não pode passar pelo meio dos quarteirões (isto é, deve ser movimentar apenas pelos acessos) e ele pretende usar apenas as rotas mais curtas para chegar ao seu destino, afinal ele voltou a se exercitar agora. Então, mesmo que exista uma rota que o leve ao seu destino, se ela for mais longa do que seria o caminho caso não existisse nenhum bloqueio, ele não seguirá por ela e não a contabilizará como caminho válido.

Ajude Joãozinho a se exercitar e verifique através de um programa o número de rotas diferentes que ele pode traçar até determinado local.

Entrada

A primeira linha da entrada contém um inteiro T ($1 \leq T \leq 200$), que indica o número de casos de teste. Cada caso de teste é representado por uma linha com quatro inteiros N ($1 \leq N \leq 100$), M ($1 \leq M \leq 100$), K ($0 \leq K \leq N \cdot M$) e L ($1 \leq L \leq N \cdot M$) indicando respectivamente o número de linhas de quarteirão, colunas de quarteirão, acessos bloqueados e consultas por parte de Joãozinho.

A seguir, existem K linhas, cada uma com uma posição (X, Y) de acessos bloqueado, com $(1 \leq X \leq N)$ e $(1 \leq Y \leq M)$. Finalmente, seguem L linhas indicando a posição (X', Y') de um destino almejado por Joãozinho, com $1 \leq X' \leq N$ e $1 \leq Y' \leq M$.

Saída

Em cada caso de teste, para cada destino almejado por Joãozinho, o seu programa deverá imprimir, em uma linha, o número de possíveis caminhos para chegar ao destino módulo $2^{31} - 1$.

Exemplos de entradas	Exemplos de saídas
3	0
1 1 1 1	1
1 1	1
1 1	2
2 2 0 3	1
1 1	1
2 1	0
2 2	0
3 3 3 4	
1 2	
2 2	
3 1	
1 1	
2 1	
3 2	
3 3	

J Tempo de Estudo

Limite de Tempo: 1s

Juninho precisa estudar para a prova de Análise de Algoritmos mas, como todo bom aluno, ele quer estudar progressivamente de modo a cobrir todo o conteúdo.

É necessário um total de T minutos de estudo para ir bem nesta prova mas, a cada dia, Juninho gosta de reservar um tempo para revisão que é equivalente a $I\%$ da quantidade de minutos restantes de estudos para completar o total em relação ao início do dia.

Ao mesmo tempo que Juninho quer cobrir todas as horas necessárias para concluir o estudo, ele quer usar todos os dias que ele tem até a prova para estudar para ela, de modo a deixar o conteúdo na cabeça o mais fresco possível. Além disso, Juninho também reserva espaço para a sua saúde mental, e ele priorizou que a quantidade de minutos diários de estudo não pode ultrapassar M minutos.

Faça um programa que, calcule a quantidade mínima de minutos por dia necessária para concluir o estudo para a prova de Análise de Algoritmos, considerando também o tempo para a revisão de conteúdo, dado que Juninho dispõe apenas de, no máximo, M minutos por dia para estudar.

Entrada

A primeira linha da entrada contém o número K ($1 \leq K \leq 1000$), o qual indica a quantidade de casos de teste.

Cada caso de teste é representado por uma única linha com um inteiro N ($1 \leq N \leq 60$), correspondendo ao numero de dias até a prova, e três reais T ($1 \leq T \leq 1440N$), M ($1 \leq M \leq 1440$) e I ($0 \leq I < 100$), os quais correspondem respectivamente ao total de minutos necessários para o estudo da prova, à quantidade de minutos máximos que Juninho está disposto a gastar com o estudo por dia e à porcentagem dedicada à revisão de conteúdo. Os quatro valores de cada caso de teste estão separados por um espaço em branco.

Saída

Para cada caso de teste imprima, em uma linha, a quantidade de minutos diária que Juninho deverá utilizar para estudar para a prova, ou “impossivel”, caso não seja possível determinar tal quantidade sobre a restrição da quantidade mínima de minutos máximos que Juninho reservou para estudar a disciplina. Se sua resposta é x e a resposta do juiz é y , sua resposta será considerada correta se $\frac{|x-y|}{\max(1,y)} \leq 10^{-3}$.

Exemplos de entradas	Exemplos de saídas
3	4.31089877
10 30.0 10.0 7.2	impossivel
20 300.0 10.0 5.1	330.00000000
1 300.0 330.0 10.0	

K Kiko, Tesouro

Limite de Tempo: 1s

Hoje é um dia especial para nosso querido Tesouro, é o aniversário de Kiko. Sabendo disso, Professor Girafales decidiu realizar uma das maiores promessas já feitas, comprar-lhe uma bola quadrada. Porém ele já está atrasado para a festa, e precisa fazer duas coisas: a primeira é comprar a bola quadrada, e a segunda é buscar Kiko, que está brincando com o Chavinho. Como o Professor Girafales gosta muito de contar o seus passos, quando chegar a casa de Dona Florinda, ele quer saber quantos passos deu.

Entrada

A entrada consiste em vários casos de teste, cada caso tem o número N de linhas e o número M de colunas que o mapa da vila possui, com $1 \leq N, M \leq 100$. As N linhas seguintes contém, cada uma, M caracteres, que representam o mapa da vila, onde o símbolo de '#' representa uma área onde o professor Girafales não pode andar, e o símbolo '.' uma área livre. No mapa, o Professor Girafales marcou a Dona Florinda como D, o Kiko como K e a Bola Quadrada como B. A entrada termina quando N ou M forem iguais a 0, e estes valores não devem ser processados.

O Professor sempre irá sair do início do mapa, fica localizado no canto superior esquerdo, e é garantido que sempre será possível alcançar os três objetivos.

Saída

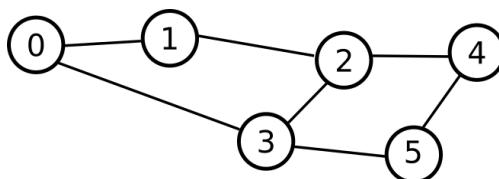
Para cada caso de teste imprima o número de passos que Professor Girafales deve dar para chegar na Bola Quadrada, no Kiko, e na casa da Dona Florinda, nesta ordem.

Exemplos de entradas	Exemplos de saídas
10 10	10 11 8
.....##	25 21 10
.#.#.#..B	
.#.####.#.	
.#K...#.#.	
...##.#.#.	
.#.#.#...##	
.#....#.#.	
.##.....#	
.###...#.#.	
....D#####	
15 15	
.###....#...###.	
..##....##..B#.	
..##.##.##...##.	
..##.##.##...##.	
#....##.....	
#.#####.###.##.	
D....#.#.#.#.	
#.###.#.###.###	
.....##.###	
.####.....##	
.#####	
.....##.#####	
.#K##.##.#####	
#####.....####	
#####...####	
0 0	

L Nunca tire o lápis

Limite de Tempo: 2s

Uma brincadeira recorrente em revistinhas de pensamento lógico é uma na qual, dada uma malha de pontos conectados, você deve colocar o lápis no ponto inicial 0 e percorrer as conexões. Você vence a brincadeira caso consiga chegar ao ponto 0 passando por todos os demais pontos, sem passar pela mesma conexão duas vezes.



No exemplo acima, é possível vencer o jogo percorrendo as seguintes ligações: 0-1, 1-2, 2-4, 4-5, 5-3, 3-0. Dado um jogo, diga se é possível vencer ou não.

Entrada

A entrada consiste em, no máximo, 10 casos de teste. Cada caso tem início em uma linha com o número N de pontos e o número M de conexões, separados por um espaço em branco, onde $3 \leq N \leq 16$, $1 \leq M \leq N(N-1)/2$. As M linhas seguintes contêm conexões a e b , sendo a o identificador do primeiro ponto e b o identificador do segundo ponto, separados por um espaço, onde $0 \leq a, b < N$ e $a \neq b$.

A entrada termina com os valores $N = M = 0$, os quais não devem ser processados.

Saída

Para cada caso diga se é possível ou não vencer o jogo, de acordo com as regras apresentadas. Caso seja possível imprima, em uma linha, a mensagem "Fechou o ciclo". Caso contrário, imprima a mensagem "Num deu =/".

Exemplos de entradas	Exemplos de saídas
6 7	Fechou o ciclo
0 1	Num deu =/
1 2	
2 3	
3 0	
2 4	
4 5	
5 3	
9 12	
0 1	
0 4	
0 3	
0 7	
1 4	
2 4	
2 6	
2 5	
3 6	
5 6	
6 8	
7 8	
0 0	

M Reconhecimento de Texto

Limite de Tempo: 2s

José Grafoldo acaba de ser contratado pela Boogle e seu primeiro trabalho é desenvolver um algoritmo para reconhecimento de texto em imagem. As imagens que José está analisando contêm apenas os caracteres A , B e C , conforme mostrado na figura abaixo:



Figura 1: formato dos caracteres válidos nas imagens.

Ajude José, desenvolvendo um algoritmo para reconhecer esses três caracteres.

Entrada

A entrada contém um único caso de teste, descrevendo a imagem a ser analisada. A primeira linha da entrada contém dois inteiros L e C ($0 < L, C \leq 200$), as próximas L linhas contêm C pixels cada. Um pixel pode ter o caractere '.' (pixel branco) ou '#' (pixel preto). A imagem de entrada respeita as seguinte regras:

- Contém apenas os caracteres mostrados na Figura 1;
- Contém ao menos um caractere válido;
- Todo pixel preto faz parte de algum caractere válido;
- Cada caractere válido consiste de um conjunto conectado de pixels pretos, ou seja, para cada pixel preto existe pelo menos um outro pixel preto acima, abaixo, à sua direita ou à sua esquerda;
- Os caracteres não se encostam e não há caracteres dentro de outros caracteres;
- Os caracteres podem estar distorcidos ou terem sofrido rotação, mas cada caractere é topologicamente equivalente a um dos caracteres da Figura 1 (duas formas são topologicamente equivalentes se uma pode ser transformada na outra apenas por deformações, sem rasgá-la).

A Figura 2 representa a imagem do exemplo de entrada.

Saída

A saída deve conter uma única linha com os caracteres encontrados em maiúsculo em ordem lexicográfica.

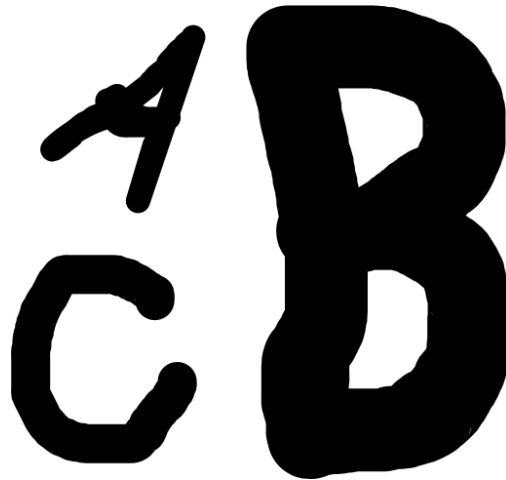


Figura 2: imagem do exemplo de entrada.

Exemplos de entradas	Exemplos de saídas
<pre> 15 30###.....#####.....#####.....###.....###.....##.##.....###.....###.....#####.....###.....###.....##.....##.....###.....###..... ..##.....##.....#####.....#####.....###.....###.....#####.....###.....###.....##.###.....###.....###.....##.....###.....###.....##.....##.....###.....###.....#####.....#####..... </pre>	<p>ABC</p>