
Seletiva para Maratona de Programação de 2015

Instituto de Matemática e Estatística
Universidade de São Paulo

Caderno de Problemas

Apoio e patrocínio:



Departamento de Ciência da Computação IME-USP

Sábado, 15 de agosto de 2015.

Instruções

- A competição tem duração de 5 horas;
- Faltando 1 hora para o término da competição, o placar será congelado, ou seja, o placar não será mais atualizado;
- Faltando 15 minutos para o término da competição, os times não receberão mais a resposta de suas submissões.
- A entrada de cada problema deve ser lida da entrada padrão (teclado);
- A saída de cada problema deve ser escrita na saída padrão (tela);
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que seu código será aceito;
- Na saída, toda linha deve terminar com o caracter ‘\n’;
- Na saída, o número zero em ponto flutuante deve ser impresso no formato 0.0, nunca com sinal negativo na frente. Se imprimir -0.0, será considerado errado!
- O nome do arquivo de códigos em Java deve ser **exatamente** como indicado abaixo do nome de cada problema. Para C/C++ é recomendado usar o nome indicado;
- Para códigos em Java, o nome da classe principal deve ser **igual** ao nome do arquivo.

Respostas das submissões		
Not answered yet	-	Paciência
YES	-	Código aceito. Parabéns!
NO	Compilation error	Erro de compilação
	Wrong answer	Errado. Pode tentar de novo.
	Time limit exceeded	Seu programa demora muito para dar a resposta (certa ou errada)
	Runtime error	Erro em tempo de execução (ex.: <i>segmentation fault</i>)
	Problem name mismatch	Leia as duas últimas instruções
	Presentation error	Não está imprimindo no formato exigido no enunciado
	If possible, contact staff	Não sei, você conseguiu fazer algo inesperado

Problema A: Bridge

Arquivo: *bridge.[c/cpp/java]*

Autor: *Stefano Tommasini*

Bridge é um grande jogo de cartas! O mundial juvenil de bridge foi realizado na Tailândia em 2006, mesmo país da final mundial da ICPC em 2016! Vamos falar sobre as regras do jogo. Ele é jogado com um baralho de 52 cartas, 13 de cada naipe (copas, espadas, paus e ouros). As cartas têm uma ordenação (A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3, 2), onde ás é a maior carta e 2 a menor. Existe também um trunfo pré definido (H = copas, S = espadas, C = paus, D = ouros, NT = não há trunfo). Um naipe definido como trunfo vale mais que os outros.

O jogo é jogado por 4 jogadores (N, E, S, W), como mostrado na figura, e consiste de várias rodadas, chamadas de vazas. Um jogador começa a rodada/vaza jogando uma carta. Em seguida, cada jogador, em sentido horário, joga uma carta até que todos os jogadores tenham jogado. Nesse momento, acaba a vaza. O jogador que começa a vaza pode jogar a carta que quiser; o naipe dessa carta é definido como o naipe da vaza. Todo jogador que não for o jogador que começou a vaza é obrigado a jogar uma carta com o mesmo naipe da vaza atual, a não ser que ele não tenha nenhuma tal carta. Nesse caso, pode jogar qualquer carta.

O vencedor de uma vaza é definido da seguinte forma: se uma carta de trunfo tiver sido jogada na vaza, ganha o jogador que jogou a maior carta de trunfo. Caso contrário, ganha a maior carta do naipe da vaza. O jogador que ganhou a vaza começa a seguinte.

O jogador N inicia a primeira vaza. Os jogadores N e S formam uma dupla, enquanto E e W formam outra. O objetivo do jogo é ganhar o maior número possível de vazas.

	N	
	♠Q873	
	♥82	
	♦J3	
	♣A9632	
W		E
♠AJ952		♠106
♥106		♥K94
♦KQ84	S	♦1062
♣K8	♠K4	♣QJ1074
	♥AQJ753	
	♦A975	
	♣5	

Figura 1: Exemplo de uma situação onde cada jogador tem 13 cartas.

Nesse problema, cada jogador tem R cartas e queremos saber quantas vazas a dupla NS consegue ganhar se todos jogarem de forma ótima.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância inicia com uma linha contendo uma string (dentre H, S, C, D, NT) indicando o trunfo da partida e um inteiro R indicando o número de cartas que os jogadores têm. Cada uma das próximas 4 linhas contém uma lista de R cartas separadas por espaços. Uma carta é representada como uma cadeia XY , onde X está dentre A, K, Q, J, T (representando 10), 9, 8, 7, 6, 5, 4, 3, 2, e Y representa o naipe (dentre H, S, C, D). As 4 linhas contêm as cartas dos jogadores N, E, S e W, nessa ordem. Note que não existe carta repetida!

Saída

Para cada instância, imprima uma linha contendo um único inteiro, correspondente ao número máximo de vazas que a dupla NS pode ganhar.

Restrições

- $1 \leq R \leq 4$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
2	1
S 1	0
2H	
AH	
2S	
KH	
NT 1	
2H	
AH	
2S	
KH	

Problema B: Renzo e a decoração capicuânica

Arquivo: *capicua*.*[c/cpp/java]*

Autor: Marcio T. I. Oshiro / Carlinhos

Nas ruínas de Wat Phra Si Sanphet (วัดพระศรีสรรเพชญ์) estão inscrições famosas que apenas recentemente tiveram seu significado desvendado. Vários números decoram as ruínas, escritos usando os dígitos tailandeses.

Há dois anos, o famoso pesquisador peruano Renzo “el intrépido” Morales verificou que a maioria dos números encontrados nas ruínas são capicuas, isto é, representam o mesmo valor se lidos ao contrário. Por exemplo, 171 é capicua, mas 17 não é capicua.

Intrigado pela presença de números que não são capicuas na decoração das ruínas, Renzo descobriu que, apesar de esses números não serem capicuas quando representados em base 10 (utilizada na escrita tailandesa), eles são capicuas se representados em uma base diferente. A representação em uma base $b > 0$ de um número N dado na base 10 é dada pela sequência $a_m a_{m-1} \cdots a_1 a_0$, tal que $0 \leq a_i \leq b - 1$, para todo $0 \leq i \leq m$, $a_m \neq 0$ e

$$a_m b^m + a_{m-1} b^{m-1} + \cdots + a_1 b + a_0 = N.$$

No exemplo anterior, a representação em base 2 do número 17 é 10001, que é capicua.

Para comprovar sua descoberta, Renzo quer que você escreva um programa que recebe um número representado na base 10 e verifica em quais bases, de 2 a 16, sua representação é capicua.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância consiste de uma única linha contendo um número inteiro N escrito na base 10.

Saída

Para cada instância, imprima em uma única linha a sequência crescente das bases, de 2 a 16, para as quais a representação de N é capicua. Se a representação de N não for capicua para nenhuma base entre 2 e 16, imprima -1.

Restrições

- $0 \leq N < 2^{31}$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
2	2 4 16
17	4 16
2570	

Problema C: Competição de Robótica

Arquivo: *robotica.[c/cpp/java]*

Autor: *Antonio Roberto C. Jr.*

Malai participará da final tailandesa de robótica na próxima semana. Nessa competição, cada competidor dará ao seu respectivo robô uma instrução e o número de vezes que essa instrução deverá ser executada. No final da competição, o robô que se aproximar mais do alvo final será o campeão.

Cada robô pode ser representado como um ponto (x, y) no plano cartesiano e sua posição inicial será a origem do plano. Cada instrução é definida por $I(\alpha, \ell)$, onde α representa um ângulo em graus e ℓ uma distância em metros. Para executar a instrução $I(\alpha, \ell)$, o robô deve rotacionar α graus no sentido anti-horário em torno da origem e depois somar ℓ metros na coordenada x da sua posição.

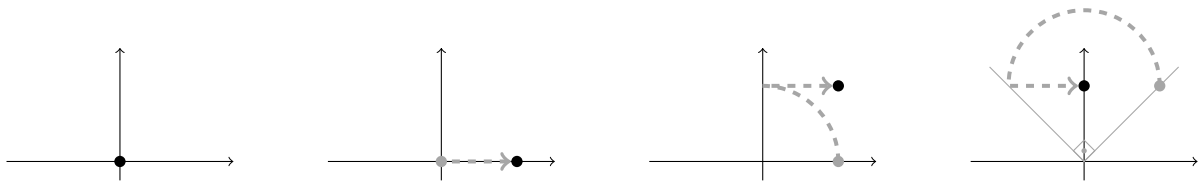


Figura 2: Execução da instrução $I(90, 10)$ três vezes.

Malai é iniciante nessa competição e precisa da sua ajudar para descobrir a posição final do robô.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância consiste de uma única linha contendo o ângulo α , a distância ℓ e o número n de vezes que a instrução $I(\alpha, \ell)$ deverá ser executada. Considere que α e n são números inteiros e ℓ é um número racional.

Saída

Para cada instância, imprima uma única linha contendo as coordenadas (x, y) , em metros, da posição final do robô, após executar n vezes a instrução $I(\alpha, \ell)$. As coordenadas x e y devem ser arredondadas e impressas com 2 casas decimais.

Restrições

- $0 \leq n \leq 10^9$
- $0 \leq \alpha \leq 365$
- $0 \leq \ell \leq 10$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
4	10.00 0.00
90 10 1	10.00 10.00
90 10 2	0.00 10.00
90 10 3	1.50 0.00
30 1.5 121	

Problema D: Passeios aleatórios pela Tailândia

Arquivo: *passeios.[c/cpp/java]*

Autor: *Stefano Tommasini*

A Tailândia tem algumas centenas de ilhas. Todas as ilhas de tamanho razoável têm pequenas pistas de voo, onde aviões de pequeno porte operam. O sistema de transporte, no entanto, é bastante estranho às vistas de um estrangeiro...

Os barqueiros são muito confiáveis. Partindo, por exemplo, de Ko Khang Khao (เกาะค้างคาว), você consegue um barco para as ilhas vizinhas a preços razoáveis: Ko Sichang (เกาะสีชัง), Ko Kham Yai (เกาะขามใหญ่), Ko Kham Noi (เกาะขามน้อย), Ko Ram Dok Mai (เกาะรำดอไม้), Ko Prong (เกาะปรัง), ou Ko Yai Thao (เกาะใหญ่ท่า)¹.

Os pilotos de avião, por sua vez, são muito temperamentais. Você pega o avião e ele deixa você em uma ilha aleatória, que ele decide de acordo com sua própria vontade. Toda ilha tem a mesma probabilidade de ser escolhida pelo piloto, inclusive a ilha de origem. Apesar de o destino ser incerto, toda viagem de avião tem o mesmo preço fixo.

Assim, quando você quer ir de uma ilha a outra tem sempre duas opções. Pegar um barco para uma ilha vizinha, cujo preço varia de acordo com o trecho percorrido, ou arriscar e pegar um avião, pagando o preço fixo de K baht.

Considere que as ilhas são numeradas de 1 a N . Sua tarefa neste problema é determinar o valor mínimo esperado de uma viagem a ser feita da ilha 1 para a ilha N .

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância inicia com 3 inteiros, N , M e K , representando o número de ilhas, a quantidade de barqueiros e o custo da viagem de avião.

As próximas M linhas contêm 3 inteiros cada, A , B , C , indicando que existe um barqueiro cobrando C baht para ir da ilha A para a B , ou da ilha B para a A . Entre cada par de ilhas existe no máximo um barqueiro.

Saída

Para cada instância, imprima um número real arredondado para 3 casas decimais com o valor mínimo esperado. O número deve sempre ser impresso com as 3 casas decimais.

Restrições

- $1 \leq N \leq 10^5$
- $1 \leq M \leq 10^5$
- $1 \leq K \leq 10^3$
- $0 \leq C \leq 10^3$

¹Sim, Ko significa ilha em tailandês

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
2	3.000
3 3 1	15.000
0 1 10	
0 2 20	
1 2 5	
3 3 100	
0 1 10	
0 2 20	
1 2 5	

Problema E: Fuga de Ayutthaya

Arquivo: *fuga*.*[c/cpp/java]*

Autor: Renzo Gonzalo Gómez Díaz

Ayutthaya foi um dos primeiros reinos da Tailândia. Ele existiu entre os anos 1351 e 1767. A organização de Investigadores de Mistérios Extraordinários (IME) tem um grande interesse em revelar os segredos dessa antiga civilização. Um dos historiadores mais renomeados dessa instituição é Márcio “o imprescindível” Himura. Atualmente, ele está investigando as regras de conduta e os castigos que eram impostos na época do rei Ramathibodi I, fundador de Ayutthaya. Descobertas recentes mostram uma forma de punição dada aos habitantes de Ayutthaya que não se convertiam ao budismo theravada, religião que foi adotada pelo rei Ramathibodi I.

Esse castigo consistia em deixar a pessoa presa em uma sala que tinha uma única saída, e onde era aceso fogo em diferentes lugares. Se a pessoa conseguia chegar à saída antes de ser atingida pelo fogo, sua vida era perdoada. Márcio tem acesso a diferentes registros que mostram a estrutura de algumas salas onde eram feitos esses castigos. Porém, não existem documentos onde se registre se a pessoa punida foi perdoada ou não. Márcio gostaria de saber se para cada uma dessas pessoas existe alguma chance de ela ter sido perdoada. Para isso, Márcio representou cada sala como uma grade de N linhas e M colunas onde cada posição contém um símbolo com o seguinte significado

Símbolo	Significado
.	livre
#	parede
F	fogo
S	início
E	saída

onde “início” se refere à posição da pessoa dentro da sala quando o fogo foi aceso. Além disso, Márcio impôs as seguintes restrições para modelar cada situação:

- O fogo se propaga nas quatro direções cardinais (N, S, E, O) à velocidade de uma célula por minuto.
- A pessoa presa também se move em essas quatro direções com essa mesma velocidade.
- Nem o fogo e nem a pessoa presa podem ultrapassar uma parede.
- Se a pessoa e o fogo chegarem a ocupar a mesma posição no mesmo instante de tempo, então considera-se que essa pessoa morre instantaneamente.

Você é um membro do IME e Márcio quer saber se você merece tal distinção. Por isso, encarregou a você a tarefa de determinar se um preso teve a chance de ser perdoado.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância consiste de diversas linhas. A primeira linha contém dois inteiros, N e M . As seguintes N linhas contém exatamente M símbolos representando, como descrito no enunciado, uma sala onde uma pessoa foi presa.

Saída

Para cada instância, imprima uma linha contendo um único caractere. Imprima Y se a pessoa teve alguma chance de ter sido perdoada; caso contrário, imprima N.

Restrições

- $1 \leq N \leq 10^3$
- $1 \leq M \leq 10^3$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
3	Y
4 5	N
....S	N
.....	
.....	
F...E	
4 4	
...S	
....	
....	
F..E	
3 4	
###S	
####	
E..F	

Problema F: Lutando contra os Rajasi

Arquivo: *rajasi*.*[c/cpp/java]*

Autor: *Stefano Tommasini*

O Muay Thai é uma arte marcial originária da Tailândia. Muitos praticantes dessa arte são considerados lendas pelo povo tailandês. Dentre todos eles, Nai Khanom Tom é considerado o “pai” do muay thai. A seguinte história faz parte da lenda desse lutador.

O rei Mangra da Birmânia organizou uma luta entre um preso de guerra tailandês e um guerreiro birmano para decidir qual tinha o melhor estilo de luta. Nai Khanom Tom foi escolhido para lutar e derrotou com muita facilidade seu adversário. Porém, o árbitro da luta alegou que isso aconteceu pela dança (Ram Muay) feita por Nai Khanom Tom. Por isso, o rei mandou a que Nai lutasse contra dez guerreiros birmanos, um após o outro. Ainda assim, o resultado foi o mesmo, Nai Khanom Tom venceu todos. Após ver as habilidades de Nai, o rei Mangra concedeu-lhe a liberdade.

Essa história passou de geração a geração. Existem até pessoas que acreditam que Nai Khanom Tom podia vencer qualquer quantidade de adversários, incluindo seres da mitologia tailandesa.

Você é um grande fã do muay thai e deseja testar a veracidade dessa afirmação. Suponha que Nai Khanom Tom possui H pontos de vida e tem que lutar contra N Rajasis. Cada um deles possui x_i pontos de vida e y_i pontos de recuperação. Para vencer uma luta, os pontos de vida de Nai devem de ser maiores do que os pontos de vida do Rajasi. Após lutar, Nai perde x_i pontos de vida e recupera y_i pontos em seguida. Além disso, pelo seu treinamento Nai conhece K feitiços que podem ser usados para derrotar um Rajasi instantaneamente. Porém, nesse caso, ele não perde nem ganha os pontos de vida como o caso anterior.

Agora, chegou a sua oportunidade de provar essa hipótese. Dada a descrição de um conjunto de N Rajasis, você deve responder se Nai Khanom Tom pode vencer todos eles. Note que Nai Khanom Tom pode lutar com os Rajasis na ordem que ele quiser.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

A primeira linha de cada instância contém 3 inteiros, N , H e K , onde H representa o valor inicial dos pontos de vida. Cada uma das N linhas seguintes contém 2 inteiros, x_i e y_i .

Saída

Para cada instância, imprima uma linha contendo Y se for possível que Nai Khanom Tom vença todos os Rajasis; imprima N caso contrário.

Restrições

- $1 \leq N, K \leq 2 \times 10^3$
- $0 \leq H, x_i, y_i < 10^9$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
2	Y
2 10 2	Y
20 10	
100 1	
2 10 0	
9 10	
10 1	

Problema G: Loteria tailandesa

Arquivo: *loteria*.*[c/cpp/java]*

Autor: Arthur Nascimento

Borommachathirat IV (สมเด็จพระบรมราชาธิราชที่ 4) foi um monarca do reino de Ayutthaya no século XVI. Borommachathirat IV decidiu organizar uma loteria para sua população de súditos, utilizando uma certa quantidade de dados. São dados tradicionais tailandeses, que podem ter várias faces, cada face podendo ocorrer com a mesma probabilidade.

Por seu forte senso de justiça, o monarca exige que o sorteio seja perfeitamente justo, ou seja, que cada um de seus súditos tenha a mesma chance de ser sorteado. O sorteio consiste de um número finito de lançamentos e, após cada lançamento, é decidido se houve algum vencedor ou se será necessário realizar um novo lançamento. Os lançamentos devem seguir as seguintes regras:

- vários dados podem ser jogados simultaneamente no mesmo lançamento;
- resultados anteriores podem influenciar na escolha dos dados para os lançamentos seguintes;
- um mesmo dado pode ser escolhido para vários lançamentos.

O importante é garantir que, como resultado do sorteio, cada um dos habitantes tenha iguais chances de ser sorteado. Veja que nem sempre isso é possível. Por exemplo, se tivermos 5 pessoas e apenas um dado de 6 faces, não há como realizar o sorteio. Já com este dado é possível realizar o sorteio se a população for de 3, 6, 18, ou 36 pessoas, por exemplo.

Sua tarefa neste problema é fazer um programa para ajudar o monarca a decidir se é possível realizar o sorteio com os dados disponíveis.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias. Cada instância é composta por duas linhas. A primeira linha contém 2 inteiros, N e K , que representam o número de pessoas e o número de dados, respectivamente. A segunda linha contém K inteiros. O i -ésimo inteiro dessa linha, digamos f_i , representa o número de faces do i -ésimo dado.

Saída

Para cada instância, imprima uma linha contendo um único caractere. Imprima Y se é possível fazer o sorteio; caso contrário, imprima N.

Restrições

- $1 \leq N \leq 10^{18}$
- $0 \leq K \leq 10^5$
- $1 \leq f_i \leq 10^{18}$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
3	Y
18 1	N
6	Y
10 2	
4 6	
1 0	

Problema H: Resguardando os Templos

Arquivo: *templos.[c/cpp/java]*

Autor: *Renzo Gonzalo Gómez Díaz*

Existem milhares de templos budistas na Tailândia. Em geral, os templos budistas são chamados de “wat”. Além disso, existem alguns templos que recebem uma distinção pela sua importância e são chamados de “templos reais”. Um exemplo desse último tipo é o templo “Wat Phra Kaew”, localizado no Grande Palácio de Bangkok. Esse templo acolhe a imagem do Buda Esmeralda, que é a mais venerada na Tailândia. Em 2016, a final mundial do ACM ICPC será realizada em Phuket, Tailândia. Por isso, espera-se que o turismo se incremente nessa cidade. Isso levou às autoridades de Phuket a pensar em melhorar a segurança dos templos reais da cidade.

Por esta razão a Unidade de Segurança de Phuket (USP) contratou a pesquisadora Lua “a engenhosa” Kuratowski. A USP está interessada em resolver o seguinte problema, dados N templos reais e M ruas que ligam esses templos, posicionar guardas nessas ruas de forma que todo templo real seja vigiado. Consideramos que um templo é vigiado se pelo menos uma das ruas que tem como extremo esse templo é vigiada por algum guarda. Por outro lado, as ruas foram construídas de forma que sempre existe uma sequência de ruas que liga dois templos quaisquer. Além disso, por costumes ancestrais de Tailândia, se fazemos uma trilha pelos templos reais da cidade, sempre andando por ruas distintas e visitando templos distintos até retornar ao templo de partida, sempre temos a certeza de ter passado por um número par de ruas, já que existe certa superstição com os números ímpares.

Como Lua é muito esperta, basta-lhe apenas olhar a descrição dos templos de Phuket, que ela já sabe qual é a resposta ótima (aquela que usa o número mínimo de guardas). Ela escreveu a resposta em um papel, enquanto assobiava “Elephant Gun”, e o entregou à USP.

Ela sabe que você deseja participar da final mundial do próximo ano, e considera que este é um bom problema para testar suas habilidades. Por isso, desafia a você a resolvê-lo.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância consiste de diversas linhas. A primeira linha contém dois inteiros, N e M que representam o número de templos reais de Phuket e o número de ruas entre templos reais, respectivamente. Cada templo é identificado por um inteiro entre 1 e N . As M linhas seguintes descrevem as ruas da cidade. Cada rua é descrita por dois inteiros que representam os templos que ela liga.

Saída

Para cada instância, imprima uma única linha contendo o número mínimo de guardas necessários para vigiar todos os templos reais da cidade.

Restrições

- $1 \leq N \leq 10^3$
- $1 \leq M \leq 5 \times 10^3$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
2	3
5 5	3
1 2	
1 4	
2 3	
4 3	
3 5	
4 3	
1 2	
1 3	
1 4	

Problema I: As vias férreas Kunming-Cingapura

Arquivo: *vias*. [c/cpp/java]

Autor: Arthur Nascimento

A via férrea Kunming-Cingapura é um conjunto de vias (já construídas e em plano de construção) que têm como objetivo ligar diferentes cidades da Ásia. Esse projeto começou em 1900 com a proposta de construir uma via férrea que ligasse a cidade de Kunming (China) a Cingapura, por parte do império britânico. Depois, em 1918, essa via férrea foi conectada ao conjunto de vias de Tailândia mediante uma via férrea que ligava Bangkok e Cingapura. No ano 2000, a ASEAN (Associação de Nações do Sudeste Asiático) propôs completar esse sistema de vias férreas.

O término desse projeto está previsto para o ano 2020. Devido à importância desse sistema para a integração do sudeste asiático, as empreiteiras responsáveis têm contactado você para calcular o custo mínimo para manter este sistema ao longo do tempo. Cada via (já construída e por construir) tem um custo de manutenção. O que se deseja é, dadas as N cidades que o sistema Kumming-Cingapura liga, as M vias iniciais do sistema e as Q vias que serão adicionadas ao longo do tempo, calcular o custo mínimo para manter o sistema conectado após construir cada uma dessas Q vias. Consideramos que o sistema está conectado se, para cada par de cidades existe um conjunto de vias que as ligam.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância é composta por diversas linhas. A primeira linha de cada instância contém três inteiros, N , M e Q (descritos no enunciado) separados por um espaço. As seguintes M linhas descrevem as vias iniciais do sistema. Cada via é representada por três inteiros a , b e c , onde a e b representam as cidades que são ligadas por essa via, e c é o custo de manutenção. As Q linhas seguintes representam as vias adicionadas ao sistema. A i -ésima linha desse grupo de Q linhas descreve a i -ésima via adicionada ao sistema através de três inteiros, como no caso anterior.

Saída

Para cada instância, imprima Q linhas. A i -ésima linha desse grupo de Q linhas deve conter um único inteiro que representa o custo mínimo de manutenção do sistema após a adição da i -ésima via.

Restrições

- $1 \leq N, M, Q \leq 3 \times 10^4$
- $1 \leq a, b \leq N$
- $1 \leq c \leq 3 \times 10^4$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
1	18
4 3 5	17
1 2 5	15
2 3 6	14
3 4 7	14
1 4 8	
1 2 4	
2 4 5	
3 4 5	
1 4 6	

Problema J: Os chedis de Kamphaeng Phet

Arquivo: *chedis*.*[c/cpp/java]*

Autor: Stefano Tommasini

Um chedi (conhecido também como estupa, pagode ou dágaba) é um monumento geralmente em formato de torre cônica construído sobre os restos mortais de uma pessoa importante da religião budista. Alguns sítios históricos tailandeses contêm dezenas desses monumentos, muitos deles dedicados a monges ou líderes religiosos antigos (bhikkhu - ภิกขุ, como são chamados na Tailândia). Vários desses monges dedicaram suas vidas à contemplação e oração a diferentes entidades. Para citar um exemplo, no sítio de Kamphaeng Phet encontram-se vários chedi em que as inscrições fazem referência a Garuda (ครุฑ)². O mesmo se repete no sítio de Si Satchanalai e de Sukhothai.

A língua tai tem diferenças muito sutis entre as diferentes letras, que torna a análise dos especialistas extremamente difícil. Por exemplo, quando qualquer símbolo da palavra correspondente ao deus Ramakien (รามเกียรติ์) é mudado, seu significado é completamente alterado. Lembrando que algumas dessas ruínas têm mais de 700 anos, estes escritos já sofrem a ação do tempo e de vândalos há séculos. Assim, o trabalho desses pesquisadores é muito complicado.

Um outro exemplo é o par de inscrições abaixo encontrada em chedis diferentes.

จัดตั้งขึ้นโดยพระภิกษุพนัญญานุ
จัดตั้งขึ้นโดยพระบรมพุทธานุญาต

Os especialistas acreditam que se refiram à mesma entidade. Para chegar a essa conclusão, eles desenvolveram um método que chamam de análise da diferença probabilística mínima.

Essa análise funciona da seguinte maneira. Sejam $a = a_1a_2 \dots a_N$ e $b = b_1b_2 \dots b_M$ duas inscrições com N e M caracteres, respectivamente. O valor do parâmetro chamado diferença é inicializado com zero. A cada passo, analisa-se um par de caracteres (a_i, b_j) , com $1 \leq i \leq N + 1$ e $1 \leq j \leq M + 1$, começando por (a_1, b_1) . Note que quando $i = N + 1$, a_i é o caracter vazio e o mesmo vale para b_j quando $j = M + 1$. Se $a_i = b_j$, pode-se considerar que existe uma correspondência entre tais caracteres nas inscrições e trocar o par atual por (a_{i+1}, b_{j+1}) . Tal correspondência nunca existe se $a_i \neq b_j$ e um caracter de uma inscrição nunca pode corresponder a dois caracteres da outra inscrição. Se for considerado que não existe correspondência entre os caracteres do par, então o valor da diferença deve ser aumentada por uma das seguintes formas:

- adicionando 1 à diferença e trocando o par atual por (a_{i+1}, b_j) ou por (a_i, b_{j+1}) ;
- adicionando K à diferença e trocando o par atual por (a_{i+x}, b_{j+y}) , onde x e y são escolhidos aleatoriamente com distribuição uniforme nos intervalos $[1, N - i + 1]$ e $[1, M - j + 1]$, respectivamente. Se $i > N$, apenas y varia aleatoriamente. Se $j > M$, apenas x varia aleatoriamente.

A análise termina quando os caracteres a_i e b_j são ambos o caracter vazio. Note que o valor da diferença pode variar de acordo com as escolhas tomadas.

Os especialistas consideram que quanto menor for a diferença obtida, maior será a correspondência entre as inscrições. Assim, eles pedem que você escreva um programa para encontrar a diferença mínima esperada entre duas inscrições.

²Não confunda com o personagem homônimo do Street Fighter.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância inicia com 3 inteiros, N , M , e K , onde N é o tamanho da primeira inscrição e M é o tamanho da segunda inscrição. As próximas duas linhas contém, respectivamente, a primeira e a segunda inscrição. Uma inscrição consiste de uma cadeia de caracteres de **a** a **z**.

Saída

Para cada instância, imprima um número real arredondado para 3 casas decimais com a diferença mínima esperada entre as duas inscrições. Imprima a resposta com exatamente 3 casas decimais.

Restrições

- $1 \leq N, M \leq 3 \times 10^3$
- $0 \leq K \leq 10^5$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
2	2.000
3 3 2	0.000
aab	
aba	
4 3 0	
abcc	
eee	

Problema K: Treinando com as larvas de Phuket

Arquivo: *larvas*.*[c/cpp/java]*

Autor: Arthur Nascimento

A culinária tailandesa é conhecida mundialmente por tentar combinar especiarias de forma a sempre ter presente num prato os elementos doce (açúcar, frutas, pimentões), apimentado, azedo (vinagre, tamarindo, lima) e salgado (shoyu, molho de peixe). O prato mais exótico, que sempre é oferecido aos turistas, é o que contém larvas de inseto fritas. O paladar ocidental costuma rejeitar a ideia de comer larvas, mas elas são muito apreciadas na Tailândia e disputadas em festas e encontros. As crianças tailandesas costumam brincar com a comida e, em particular, adoram construir triângulos usando as larvas fritas como arestas.

Marcos “o resolvedor” de Rubik ajuda no treinamento de alunos de sua universidade para uma famosa competição de programação de computadores. No ano que vem, a final mundial dessa competição será em Phuket, na Tailândia.

De conhecimento desse costume das crianças tailandesas, Marcos teve uma ideia para um treino especial. A ideia é preparar uma grande quantidade de larvas fritas de diversos comprimentos. Cada um de seus amigos, em sequência, deve selecionar 3 delas para formar um triângulo e precisará comer uma quantidade de larvas fritas proporcional à área do triângulo formado.

Marcos espera que, para comer menos larvas fritas possível, você escreva um programa para escolher as larvas que formam um triângulo de área mínima. Dessa forma, além de treinar suas habilidades de programação, você também estará treinando para enfrentar a culinária tailandesa. Caso goste dessa iguaria, você pode usar esse programa para ajudar seus amigos, fazendo com que mais larvas fritas sobrem na sua vez.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância consiste de um inteiro N , representando o número de larvas, seguido de N números reais, representando seus comprimentos. O comprimento de cada larva é um número real entre 1 e 500.

Saída

Para cada instância, imprima uma única linha contendo a área mínima para os dados da instância, arredondado para 2 casas decimais; se não for possível formar um triângulo com as larvas, imprima -1.

Restrições

- $1 \leq N \leq 2 \times 10^3$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
3	5.33
4	-1
3 4 5 6	4.36
3	
1 2 4	
5	
3.4 2.8 7.1 5.2 10	

Problema L: Emplacando os *tuk-tuks*

Arquivo: *tuktuk*.*[c/cpp/java]*

Autor: Marcio T. I. Oshiro / Carlinhos

Na Tailândia, um tipo popular de transporte público é o chamado *tuk-tuk* (ตุ๊กตุ๊ก), também conhecido como auto-riquixá. O governo de Phuket decidiu criar um novo sistema de placas para os *tuk-tuks*, com a finalidade de diferenciá-los dos outros tipos de veículos. Devido ao turismo, que é uma das principais atividades econômicas da província, a frota de *tuk-tuks* vem crescendo rapidamente. Espera-se que com o novo sistema de placas seja possível criar uma quantidade suficiente de placas distintas para atender à demanda pelos próximos 42 anos.

Um sistema de placas é definido por dois números, C e D . Uma placa nesse sistema é uma cadeia com C consoantes seguidas por D dígitos. Uma placa não pode ser vazia (sem consoantes e sem dígitos).

No alfabeto tailandês existem 44 consoantes e 10 dígitos. No entanto, como os símbolos de algumas consoantes são parecidos com os de outras, o governo decidiu que serão utilizadas somente 26 consoantes, cujos símbolos foram considerados suficientemente diferentes.

Para garantir que existirão *tuk-tuks* suficientes para os competidores da Final Mundial da Maratona de Programação em 2016, o governo de Phuket quer saber qual o número de placas distintas é possível gerar com um determinado sistema de placas.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância consiste em uma linha contendo os números inteiros C e D representando as quantidades de consoantes e dígitos, respectivamente, em um sistema de placas.

Saída

Para cada instância, imprima uma linha com a quantidade de placas distintas que podem ser geradas pelo sistema correspondente. É garantido que a resposta sempre será menor que 2^{31} .

Restrições

- $0 \leq C \leq 6$
- $0 \leq D \leq 9$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
3	1000000
0 6	6760000
2 4	0
0 0	

Problema M: Removendo moedas no Kem Kradãñ

Arquivo: *moedas*.*[c/cpp/java]*

Autor: Marcos Kawakami

Andréh e Andréas são dois amigos multiperitos que gostam muito de jogos. Sabendo que vários de seus amigos anseiam por uma viagem a Phuket, na Tailândia, Andréh e Andréas querem desafiar seus amigos em um tradicional jogo tailandês, o Kem Kradãñ.

Kem Kradãñ (เกมกระดาน) é um jogo de tabuleiro tailandês existente desde o século II a.C. As peças do jogo são N moedas em que uma das faces é dourada e a outra é branca. Inicialmente as moedas são dispostas em linha sobre o tabuleiro e numeradas de 1 a N , da esquerda para a direita. Quando uma moeda de número i está com a face dourada exposta, ela pode ser retirada da mesa. Ao fazê-lo, as moedas de números $i - 1$ e $i + 1$, caso existam e ainda estejam no tabuleiro, são revertidas. Isto é, se estavam com a face dourada exposta, passam a ficar com a face branca exposta e vice-versa. O jogo consiste em tentar remover todas as moedas da mesa.

Antes de desafiar seus amigos, Andréh e Andréas querem ter certeza que as configurações escolhidas para o desafio têm solução. Para ajudá-los, dada uma configuração inicial, você deve determinar se é possível remover todas as moedas e, se possível, mostrar como fazer isso.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

Cada instância é dada por um inteiro N , representando o número de moedas, seguido por uma cadeia de comprimento N , formada pelos caracteres B (face branca exposta) e D (face dourada exposta), representando uma configuração inicial.

Saída

Para cada instância, imprima em uma linha Y se for possível remover todas as moedas, ou N caso contrário. Se for possível remover todas as moedas, a linha seguinte deve conter uma lista de N inteiros separados por espaços, cada qual representando o número de uma moeda, indicando a sequência em que elas devem ser removidas. Caso haja mais de uma sequência possível, você pode imprimir qualquer uma delas.

Restrições

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

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
4	Y
3	2 1 3
BDB	Y
5	1 4 2 3 5
DBDDB	N
5	Y
DDBDD	1 2 3 4 5 6
6	
DBBBBB	