

III Maratona de Programação do IFB

Maratona

24 de novembro de 2018



(Este caderno contém 12 problemas)

Comissão Organizadora:

Prof. Daniel Saad Nogueira Nunes (IFB)
Prof. Edson Alves da Costa Júnior (UnB/FGA)
Prof. Felipe Duerno (UnB/FGA)
Prof. Matheus de Souza Faria (UnB/FGA)
Prof. Vinicius Borges (UnB)
Arthur Komatsu (UnB/FGA)
Jeremias Moreira Gomes (UnB)
Lucas Mattioli (UnB/FGA)

Apoio:

Anderson Nascimento (IFB)
Carlos Eduardo Santana (IFB)
Danyelle Angelo (IFB)
Flávia Dias Campos (IFB)
Gregory Pereira (IFB)



QNM 40, Área Especial nº 01, Taguatinga/DF, 72146-000 , Brasil
Telefone (61) 2103-2200
<http://www.ifb.edu.br/taguatinga>

Problema A

Raio de Visão

Limite de tempo: 1s

Autor: Daniel Saad

Joãozinho está programando o seu novo ARPG (Action Role Playing Game). No momento ele está concentrado em identificar quais jogadores conseguem enxergar um determinado item quando este cai de algum monstro.

Neste jogo, cada personagem tem um raio de visão específico, desta forma, alguns personagens podem enxergar mais longe do que outros. Este raio de visão engloba depende de vários fatores, como: a classe do jogador (paladino, feiticeira, bárbaro, *etc*); os itens que cada personagem carrega em seu inventário e o nível do personagem.

Se um personagem tem um raio de visão R , qualquer item que se encontre dentro do círculo com centro no personagem e raio R pode ser visto por este personagem.

Como Joãozinho ainda tem que trabalhar no módulo de geração de números aleatórios, ele pediu que você o ajudasse nesta tarefa.

Assim, dada as localizações dos vários jogadores, juntamente com os raios de visão de cada um e a localização de um item, você deverá escrever um programa que determine quais jogadores conseguem enxergar este item.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 10^3$) indicando o número de jogadores presentes no jogo.

As próximas N linhas contém, cada uma, a identificação de cada jogador, a qual consiste de três números reais X_i, Y_i ($-1000 \leq X_i, Y_i \leq 1000$) e R_i ($0 \leq R_i \leq 1000$), indicando respectivamente a coordenada (X_i, Y_i) do i -ésimo jogador e o seu raio de visão.

Por fim, a última linha contém dois números reais C e D ($-1000 \leq C, D \leq 1000$), indicando a coordenada (C, D) da localização do item.

Saída

Você deverá imprimir uma linha com as identificações dos jogadores que conseguem enxergar o item, separadas por espaço e em ordem crescente. A identificação do i -ésimo jogador deve ser escrita como “P< i >”, em que “< i >” corresponde ao valor de i . A base da identificação numérica dos jogadores começa de 1.

Caso nenhum jogador consiga ver o item, uma linha com as palavras “Ninguem viu!” deverá ser impressa.

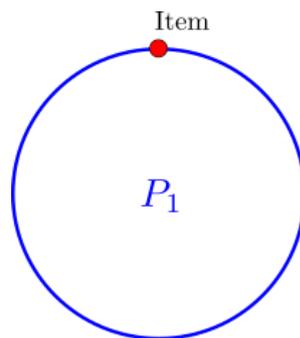
Exemplo

III Maratona de Programação do IFB

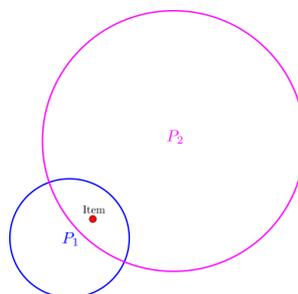
Entrada	Saída
1	P1
0.0 0.0 1.0	
0.0 1.0	
2	P1 P2
0.0 0.0 1.0	
1.5 1.5 2.0	
0.5 0.5	
3	Ninguem viu!
0.0 0.0 1.0	
0.0 1.5 2.0	
-2.0 -1.0 4.0	
2.0 0.5	

Notas

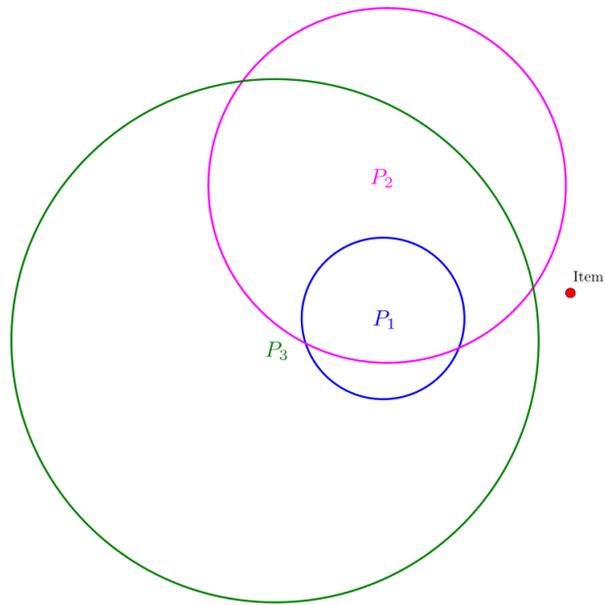
O primeiro caso de teste está representado pela figura a seguir. Nele podemos ver que só o P1 enxerga o item.



O segundo caso de teste, representado pela figura abaixo, ilustra a situação em que os jogadores P1 e P2 enxergam o item.



Finalmente, a próxima figura, que ilustra o terceiro caso de teste, mostra uma situação em que nenhum jogador consegue enxergar os itens.



Problema B

Corretor Automático

Limite de tempo: 1s

Autor: Daniel Saad

O professor Fagundes, cansado de passar intermináveis horas corrigindo as provas de seus alunos, resolveu mudar a sua metodologia de avaliação. Agora, o professor só elabora provas com questões de múltipla escolha.

Querendo poupar ainda mais esforço, Fagundes contratou você como estagiário para bolar um sistema que: leia as opções marcadas pelo aluno, compare-as com o gabarito, e dê a nota final.

Entrada

A primeira linha da entrada consiste de um inteiro N ($1 \leq N \leq 200$) que indica o número de questões da prova do prof. Fagundes.

Em seguida, há N linhas contendo dois caracteres maiúsculos separados por espaço e que podem assumir valores de 'A' a 'E'. O primeiro caractere representa a opção marcada pelo aluno enquanto o segundo representa a opção correta, fornecida pelo gabarito

Saída

Imprima, em uma linha, as palavras "Nota: <x>", em que "<x>" representa a nota do aluno de 0 a 10 com precisão de 2 casas decimais.

Para cada caso de teste, se sua resposta é y e a resposta do juiz é z , sua resposta será considerada correta se $\frac{|y-z|}{\max(1,z)} \leq 10^{-2}$.

Exemplo

Entrada	Saída
1 A A	Nota: 10.00
2 B E D D	Nota: 5.00
6 D D A B C C D A E E B B	Nota: 6.67

Notas

III Maratona de Programação do IFB

No primeiro caso de teste, o aluno acertou a única questão da prova, obtendo nota 10.00.

No segundo caso de teste, apenas metade das questões foram acertadas, obtendo-se a nota 5.00.

Por fim, o terceiro caso de teste apresenta um cenário em que $2/3$ da prova estão corretos, obtendo a nota de 6.67.

Problema C

Popstar

Limite de tempo: 1s

Autor: Daniel Saad

Um concurso está sendo realizado para determinar qual é o aluno *popstar* do IFB.

Para ser *popstar*, um aluno deve:

- Ser conhecido por todos os outros alunos.
- Conhecer apenas a si mesmo, afinal, um *popstar* não perde o seu tempo com outras pessoas.

Assim que identificado, o aluno *popstar* será premiado com as “sandálias da humildade”, para que este venha a melhorar suas relações interpessoais.

Como são muitos alunos, você foi escolhido para fazer um programa que dê como resposta o identificador do aluno *popstar*, caso haja algum. Como você não é um aluno *popstar*, aceitou este desafio.

Entrada

A entrada consiste de uma linha contendo um inteiro N ($1 \leq N \leq 1000$) indicando o número de alunos do IFB.

Em seguida há N linhas. A i -ésima linha possui N valores $V_{i,j}$ de modo que $V_{i,j} = 1$ se o i -ésimo aluno conhece o j -ésimo, e $V_{i,j} = 0$, caso contrário.

Saída

Imprima, em uma linha, as palavras “Popstar: <i>”, onde “<i>” representa o identificador do aluno *popstar*. Caso não haja nenhum aluno na condição de *popstar*, deve ser impressa uma linha com a mensagem “Nao ha popstar.”.

Exemplo

Entrada	Saída
1	Popstar: aluno #1
1	
2	Popstar: aluno #2
1 1	
0 1	
3	Nao ha popstar.
0 1 0	
1 1 1	
0 1 1	

Notas

III Maratona de Programação do IFB

No primeiro caso de teste, o aluno #1 é *popstar*, pois ele conhece a si mesmo e não há outros alunos.

No segundo caso de teste, o aluno #2 *popstar*, pois ele conhece somente a si mesmo e o aluno #1 o conhece.

Finalmente, não há aluno *popstar* no terceiro caso de teste, pois, para cada aluno, ele não se conhece ou não é conhecido por todos os outros.

Problema D

Middle Square

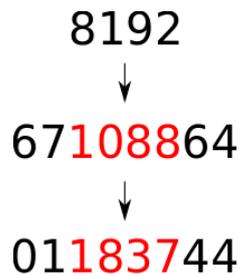
Limite de tempo: 1s

Autor: Daniel Saad

No desenvolvimento do seu jogo de RPG, Joãozinho estava em busca de métodos de geração de números pseudoaleatórios para implementar o cálculo de itens que cairiam dos monstros após o personagem do jogo matá-los. Foi então que Joãozinho se deparou com um método chamado **Middle Square**.

Neste método, dada uma semente inicial de 4 dígitos decimais, calcula-se o quadrado deste número. Caso o quadrado possua menos de 8 dígitos decimais, zeros à esquerda são adicionados até que o número fique com 8 dígitos. Por fim, os quatro dígitos centrais são selecionados como o próximo número pseudoaleatório. Cada número pseudoaleatório gerado é utilizado como semente para gerar o próximo número da sequência.

Por exemplo, caso o número 8192 seja utilizado como semente, serão gerados em sequência os números 1088 e 1837, conforme mostra a figura abaixo. Repare que, na geração do número 1837, um 0 (zero) à esquerda foi adicionado após elevar 1088 ao quadrado.



Joãozinho não tinha certeza se este era um bom método de geração de números pseudoaleatórios. Como ele estava ocupado (de novo, Joãozinho?) desenvolvendo o módulo de interface com o usuário de seu jogo, ele precisa de ajuda para implementar este método. Ele solicitou a você um programa que, dada uma semente inicial, liste todos os números pseudoaleatórios gerados pelo **Middle Square** até que ocorra uma repetição.

Entrada

A primeira linha da entrada contém o número N ($0 \leq N \leq 10000$) de casos testes que o seu programa deverá processar.

As N linhas seguintes contém, cada uma, o valor do inteiro V_i ($0 \leq V_i \leq 9999$), indicando a semente inicial do i -ésimo caso de teste.

Saída

Para cada caso de teste o seu programa deverá imprimir duas linhas. A primeira linha deverá conter as palavras “**Teste #<i>**”, onde “<i>” indica o número do teste. A próxima linha deverá listar os números pseudoaleatórios gerados pelo método **Middle**

Square a partir da semente data, separados por um espaço em branco, até que ocorra uma repetição.

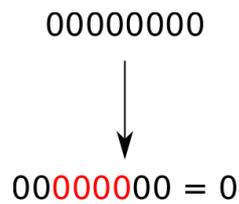
O valor repetido não deve ser impresso.

Exemplo

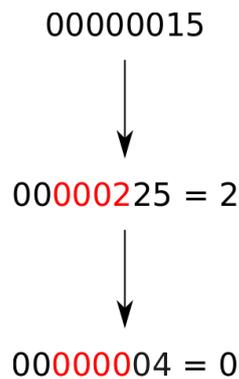
Entrada	Saída
3	Teste #1
0	0
15	Teste #2
1100	15 2 0
	Teste #3
	1100 2100 4100 8100 6100

Notas

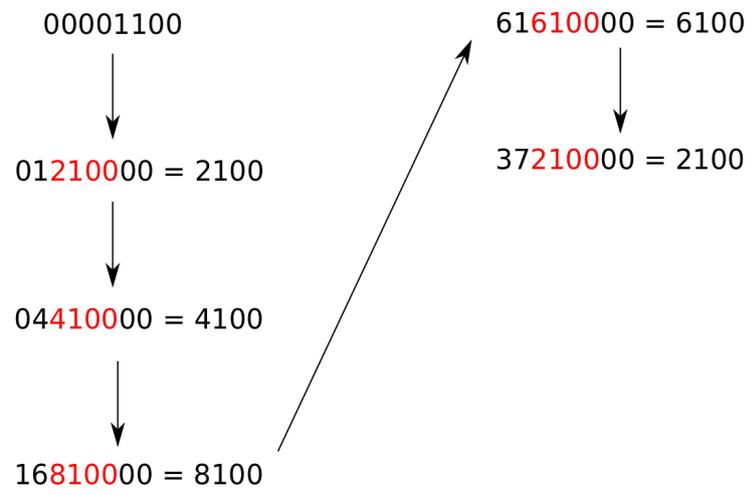
O primeiro caso de teste gera a sequência com o número 0 apenas, conforme figura abaixo:



Já no segundo caso de teste, a sequência 15, 2, 0 é gerada, conforme a próxima figura:



No terceiro caso de teste, a sequência 1100, 2100, 4100, 8100, 6100 é gerada, conforme a figura a seguir:



Problema E

Pisca-Pisca

Limite de tempo: 1s

Autor: Matheus Faria

Revirando as decorações do Natal passado, Maria encontrou um antigo pisca-pisca de N cores. Estes piscas-piscas são famosos por conter exatamente N cores de lâmpadas e por seguir um padrão de cores. Por exemplo, o pisca-pisca de 3 cores segue o padrão Vermelho - Verde - Azul, e repete esse padrão por toda a extensão do pisca-pisca.

Como este pisca-pisca era muito antigo, algumas lâmpadas queimaram. Ajude Maria a saber quantas lâmpadas de cada tipo ela deve comprar para reparar o seu lindo pisca-pisca.

Entrada

A primeira linha da entrada contém o número N ($1 \leq N \leq 10^3$) de cores de lâmpadas distintas e o número Q ($1 \leq Q < 10^6$) de lâmpadas queimadas, separados por um espaço em branco.

A linha seguinte contém Q valores q_i ($1 \leq q_i \leq 10^6$), indicando as posições das lâmpadas queimadas.

Saída

Imprima, em uma linha, o número de lâmpadas necessárias para cada cor distinta, seguindo a ordem das lâmpadas que serão utilizadas no pisca-pisca, separados por um espaço em branco.

Assuma que as cores distintas são numeradas consecutivamente de 1 a N e que a sequência começa na lâmpada 1 e segue sequencialmente até a lâmpada N , antes de reiniciar o seu ciclo.

Exemplo

Entrada	Saída
3 6	2 1 3
1 3 6 7 8 9	
5 10	2 3 3 1 1
7 8 13 1 2 90 84 21 23 12	
7 6	1 2 1 2 0 0 0
3 1 23 4 2 25	

Notas

No primeiro caso de teste, temos 3 cores distintas e 6 luzes danificadas. Como as luzes sempre vão estar na sequência 1 – 2 – 3, teremos a seguinte situação:

III Maratona de Programação do IFB

1 2 3 1 2 3 1 2 3
1 x 3 x x 6 7 8 9

Logo, serão necessárias 2 lâmpadas da cor 1, 1 da cor 2, e 3 da cor 3.

Problema F

Reservatório

Limite de tempo: 1s

Autor: Vinicius Borges

Seu Alfredo possui em sua fazenda um reservatório que armazena água captada da chuva. Diariamente, Seu Alfredo verifica se o nível do reservatório aumenta ou diminui para utilizar a água de maneira racionada nos dias que se sucedem. Futuramente, Seu Alfredo pretende ativar uma piscina que irá receber água desse reservatório. No entanto, a ideia é que essa piscina seja ativada somente nos dias consecutivos em que o nível de água do reservatório apresentar aumento.

Durante N dias, Seu Alfredo planeja verificar diariamente as medições v_i que expressam se o nível do reservatório aumentou ou diminuiu em *Unidades de Água* (UA). Para auxiliar o Seu Alfredo, sua tarefa é construir um programa que calcule o maior valor total referente às altas consecutivas do nível do reservatório.

Por exemplo, se em 6 dias o nível do reservatório variou conforme $\{2, -4, 5, 3, 2, -1\}$, então Seu Alfredo ativará a piscina entre os dias 3 e 5 somente, pois o nível do reservatório subiu 10 UA (foi o maior variação positiva).

Entrada

A entrada consiste de um único caso de teste. A primeira linha contém dois valores inteiros $2 \leq N \leq 10^4$, que descreve uma quantidade de dias. A segunda linha contém N inteiros v_i , que expressam a variação do nível do reservatório em UA, em que $-10^4 \leq v_i \leq 10^4$ e $1 \leq i \leq N$.

Saída

Imprima um único valor inteiro que represente o maior acumulado de água que o reservatório de Seu Alfredo consiga armazenar. O valor 0 deve ser impresso caso não seja possível ativar a piscina em nenhum dia.

Exemplo

Entrada	Saída
6	10
2 -4 5 3 2 -1	
7	7
-3 2 5 -1 0 -1 -3	
5	0
-4 -3 -1 -5 -2	

Notas

III Maratona de Programação do IFB

No primeiro caso de teste, Alfredo observou o total de 10UA de aumento do reservatório entre os dias 3 e 5.

No segundo caso de teste, Alfredo observou o total de 7UA de aumento do reservatório entre os dias 2 e 3 ou entre os dias 2 e 5.

No terceiro caso de teste, Alfredo não observa aumento dos níveis de reservatório em qualquer período observado.

Problema G

Financiamento

Limite de tempo: 1s

Autor: Edson Alves

Carlos deseja adquirir um bem cujo valor à vista é de V , mas dispõe de, no máximo, M reais mensais para pagar uma prestação. À prazo, o mesmo bem pode ser adquirido por N prestações mensais P tais que

$$V = \frac{P}{(1+i)} + \frac{P}{(1+i)^2} + \dots + \frac{P}{(1+i)^N},$$

onde i é a taxa de juros mensal.

Determine o prazo mínimo, em meses, do financiamento que Carlos deve contrair para adquirir o bem pagando, no máximo, M reais por mês de prestação.

Entrada

A entrada consiste em uma única linha, com os inteiros V, j, M ($1 \leq V \leq 10^6, 1 \leq j \leq 99, 1 \leq M \leq 10^4$), separados por um espaço em branco, onde V é o valor do bem à vista, em reais; j é a taxa de juros, em porcentagem; e M é o valor máximo, em reais, que Carlos dispõe para pagar a prestação mensal.

Observe que $i = (j/100.0)$.

Saída

Imprima, em uma linha, o número mínimo de meses do financiamento que Carlos deve contrair. Se o tempo necessário for maior do que 10^6 meses ou for impossível obter prestações menores ou iguais a M , imprima o valor -1 .

Exemplo

Entrada	Saída
1000 2 100	12
1000 3 1	-1
2500 5 350	10

Notas

No primeiro caso, um financiamento de 12 meses resultaria numa prestação de R\$ 94,56 reais, dentro das possibilidades de Carlos. Veja que 13 meses resultaria numa prestação de R\$ 88,12 reais, mas Carlos quer pagar sua dívida o mais rápido possível; em 11 meses a prestação seria de R\$ 102,18, acima de seu limite mensal.

No segundo caso, o financiamento de R\$ 1.000,00 a um juros mensal de 1% não pode ser pago com parcelas de um real (o juros mensal seria maior do que este valor, e a dívida nunca seria paga).

No terceiro caso, seriam 10 parcelas mensais de R\$ 323,76 reais.

Problema H

Matemática Prefixada

Limite de tempo: 1s

Autor: Matheus Faria

Joana está aprendendo novos conceitos na matemática. Ela aprendeu que expressões prefixas são aquelas que o operador vem na frente dos operandos. Ou seja, a operação $A + B$ torna-se $+AB$. Outro exemplo é a operação $A + B * C$, que seria escrita na forma $+A * BC$.

Joana sabe que expressões prefixas só podem ser avaliadas de uma única maneira. Porém ela está com muita dificuldade para resolvê-las. Ajude ela a encontrar os resultados das expressões prefixas.

Entrada

A entrada consiste em um número N ($1 \leq N \leq 26$) de valores. Em seguida há N linhas, cada uma com letra maiúscula e um número M ($1 \leq M \leq 100$). Na última linha da entrada há uma expressão prefixa válida, que será composta pelas operações $+ - */$ e por letras maiúsculas.

Todas as letras que fazem parte da expressão estão listadas em uma das N linhas que a precedem.

Saída

Imprima o resultado da expressão. Todas as expressões retornam valores válidos, sem entrar em qualquer indeterminação, ou qualquer resultado que não pertença ao conjunto dos naturais.

Exemplo

III Maratona de Programação do IFB

Entrada	Saída
2	3
A 1	
B 2	
+ A B	
3	94
A 10	
B 12	
C 7	
+ A * B C	
4	341
A 8	
B 23	
C 10	
D 1	
* + A B + C D	
4	711
A 10	
B 20	
C 33	
D 41	
+ + A * B C D	

Notas

No primeiro caso de teste, como $A = 1$ e $B = 1$, a expressão prefixa $+AB$ avalia em 3.

No segundo caso de teste, tomando $A = 10$, $B = 12$ e $C = 7$ a expressão $+A * BC$ tem como resultado 94.

O terceiro caso de teste, com $A = 8$, $B = 23$, $C = 10$ e $D = 1$, tem a expressão $* + AB + CD$ avaliada em 341.

Problema I

Quem Eliminar?

Limite de tempo: 1s

Autor: Lucas Mattioli

Marlin é um ávido jogador de MMORPG. Ultimamente, ele tem se aventurado bastante em WoW (World of Warcraft), de forma a ficar acordado até as 6 da manhã e perder aulas importantíssimas devido à falta de sono.

No WoW, tanto o personagem do jogador quanto seus inimigos podem ser vistos como pontos em um plano $2D$. Cada inimigo tem uma posição inicial de *spawn* representada por um ponto $S_i = (X_i, Y_i)$, um alcance de visão L_i , um padrão de movimentação M_i (podendo ser horizontal ou vertical) e uma distância máxima de deslocamento em relação ao ponto de *spawn* representada por D_i . Todos os inimigos se movem pelo mapa com velocidade de 1 unidade de espaço por 1 unidade de tempo.

Em qualquer posição (A_i, B_i) que um inimigo i se encontrar, ele pode eliminar qualquer jogador que esteja dentro ou nas bordas do quadrado de lado L_i que tem como centro o ponto (A_i, B_i) .

Além disso, a movimentação de um inimigo i funciona da seguinte maneira:

- Se seu padrão de movimentação for vertical, ele se movimenta, em linha reta, de S_i até o ponto $(X_i, Y_i + D_i)$. Logo depois, se movimenta do ponto $(X_i, Y_i + D_i)$ até o ponto $(X_i, Y_i - D_i)$. Por fim, retorna, também em linha reta, ao ponto S_i e o processo é repetido infinitamente.
- Se seu padrão de movimentação for horizontal, ele se movimenta, em linha reta, de S_i até o ponto $(X_i + D_i, Y_i)$. Logo depois, se movimenta do ponto $(X_i + D_i, Y_i)$ até o ponto $(X_i - D_i, Y_i)$. Por fim, retorna, também em linha reta, ao ponto S_i e o processo é repetido infinitamente.

Durante suas movimentações, os inimigos não colidem entre si e podem ocupar a mesma posição em qualquer momento.

Finalmente, Marlin, querendo ir ao banheiro desesperadamente, pretende deixar seu personagem parado numa posição (X_S, Y_S) no meio de uma *quest*, onde N inimigos estão se movimentando de um lado para o outro conforme seus respectivos padrões de movimentação. Para não morrer, Marlin pretende eliminar todo e qualquer inimigo que representa uma ameaça; isto é: todo inimigo que, eventualmente, terá Marlin no seu campo de visão e o eliminará impiedosamente. Marlin é habilidoso o suficiente para eliminar todos os inimigos de uma só vez, independente da distância que os mesmos se encontram. Além disso, Marlin é rápido o suficiente para eliminar qualquer inimigo antes que o mesmo tente fazer o mesmo com Marlin. E, claro, para poupar tempo, Marlin não eliminará os inimigos que não representam uma ameaça.

Você pode ajudá-lo a computar quantos inimigos devem ser eliminados impiedosamente?

Entrada

A primeira linha da entrada contém dois inteiros X_S e Y_S ($-10^8 \leq X_S, Y_S \leq 10^8$), representando a posição de Marlin.

A segunda linha da entrada contém um inteiro N ($1 \leq N \leq 10^5$), representando a quantidade de inimigos na *quest*.

Por fim, cada uma das próximas N linhas representa um inimigo. Cada linha contém quatro inteiros X_i, Y_i, L_i, D_i ($-10^8 \leq X_i, Y_i \leq 10^8, 2 \leq L_i \leq 10^8, 1 \leq D_i \leq 10^8$) e um carácter M_i ($M_i \in \{H, V\}$), representando, respectivamente: a coordenada X do ponto S_i , a coordenada Y do ponto S_i , o alcance de visão do i -ésimo inimigo, a distância máxima de deslocamento do i -ésimo inimigo e o padrão de movimentação do i -ésimo inimigo.

Observações:

1. L_i é par.
2. Se $M_i = H$, então o padrão de movimentação do i -ésimo inimigo é horizontal. Se $M_i = V$, então o padrão de movimentação do i -ésimo inimigo é vertical.

Saída

Imprima uma única linha contendo um inteiro que representa a quantidade mínima de inimigos que Marlin tem que eliminar de forma a não correr riscos de morte enquanto estiver no banheiro.

Exemplo

Entrada	Saída
5 3	1
4	
2 5 2 1 H	
2 2 2 2 H	
3 2 2 4 V	
0 -10 4 2 H	
0 0	2
2	
3 3 8 1 H	
0 0 2 10 V	
5 3	2
4	
2 5 10 1 H	
2 2 2 2 H	
3 2 2 4 V	
0 -10 4 2 H	

Notas

No primeiro exemplo, Marlin precisa eliminar apenas o segundo inimigo.

No segundo exemplo, a posição de *spawn* do segundo inimigo é a mesma de Marlin: isso não é problema para Marlin, ele consegue eliminá-lo de qualquer jeito.

Problema J

Conquista de Territórios

Limite de tempo: 1s

Autor: Felipe Duerno

Joana acaba de inventar um novo jogo de tabuleiro. Neste jogo, o mapa é retangular de tamanho L por C e existem dois tipos de território: água e terra.

O mapa fica então dividido em regiões, onde uma região é definida por todos os territórios do tipo terra que compartilham adjacências na horizontal ou na vertical, mas não na diagonal. Nesse sentido, os territórios do tipo água são os delimitadores das regiões e não podem ser conquistados.

A missão de cada jogador é conquistar o maior número de territórios, com exatamente N soldados. Um soldado, ao ser colocado em uma posição do mapa, conquista diretamente aquele território e, indiretamente, todos os territórios daquela região. Quando todos os territórios do tipo terra são conquistados, é dito que o planeta foi conquistado e o jogo acaba.

Dado o mapa do jogo e as posições dos soldados, verifique se o planeta foi conquistado ou quantos territórios ainda precisam ser conquistados.

Entrada

A primeira linha da entrada contém dois inteiros positivos L e C ($1 \leq L, C \leq 1000$), indicando o número de linhas e colunas do mapa, respectivamente. Cada uma das próximas L linhas contém C caracteres cada. Cada caractere indica o terreno de uma coordenada do mapa: “.” para água e “#” para terra.

Na linha seguinte, inteiro positivo N ($1 \leq N \leq 1000$), indicando a quantidade de territórios conquistados diretamente, deve ser lido. Cada uma das próximas N linhas contém dois inteiros X ($1 \leq X \leq L$) e Y ($1 \leq Y \leq C$), indicando as coordenadas (linha e coluna) de uma território conquistado diretamente.

Não é possível conquistar os rios (territórios do tipo água), desta forma, as coordenadas X e Y nunca apontarão para um território do tipo água. A contagem das linhas vai de cima para baixo (linha mais acima tem valor 1 e a mais abaixo tem valor L) e a contagem das colunas vai da esquerda para a direita (coluna mais a esquerda tem valor 1 e a mais a direita tem valor C).

Saída

Caso todos os territórios tenham sido conquistados, imprima o texto “O planeta foi conquistado.”, caso contrário, imprima o texto “Territorios conquistados: K/T ”, onde K deve ser substituído pelo número de territórios conquistados e T pelo número total de territórios passíveis de conquista.

Exemplo

III Maratona de Programação do IFB

Entrada	Saída
5 10###..... .###...###### 2 3 3 2 2	Territorios conquistados: 6/12
5 16 #.####..#.#...# #.#...#####.### .###.###.#...## ###..###...##.# .#..#####...### 4 5 16 2 1 3 4 2 9	0 planeta foi conquistado.
5 16 #.####..#.#...# #.#...#####.### .###.###.#...## ###..###...##.# .#..#####...### 3 2 1 3 4 2 9	Territorios conquistados: 33/45

Notas

No primeiro caso de teste, existem 12 territórios passíveis de conquista, divididos em duas regiões de 6 territórios cada. Os dois soldados foram colocados em apenas uma das regiões, deixando a outra por conquistar, desta forma, foram conquistados 6 de 12 territórios.

No segundo caso de teste, o posicionamento dos soldados garante que todas as regiões são conquistadas.

No terceiro caso de teste, existem 45 territórios passíveis de conquista. Os três soldados foram dispostos de forma que apenas 33 dos territórios foram conquistados.

Problema K

Noob Authenticator

Limite de tempo: 1s

Autor: Jeremias Gomes

Após o vazamento de dados do Instituto Fenomenal, a equipe de segurança decidiu aperfeiçoar a autenticação de usuários utilizando um aplicativo validador baseado na própria senha do usuário.

O aplicativo funciona da seguinte forma: toda vez que o usuário vai conectar ao sistema, o aplicativo validador utiliza como PIN a própria senha do usuário embaralhada, mas de forma que nenhum caractere da senha embaralhada esteja na mesma posição em relação a senha correta.

As regras para criação de senhas no instituto são:

1. As senhas devem conter no mínimo 3 caracteres;
2. As senhas podem conter letras minúsculas, maiúsculas ou números;
3. As senhas não podem conter caracteres repetidos; e
4. As senhas não podem conter espaços.

Dadas as condições mínimas para criação de senhas e as regras para o funcionamento do aplicativo, o Instituto Fenomenal gostaria da sua ajuda para saber quantos PINs diferentes o aplicativo poderá gerar, dada uma senha do usuário.

Entrada

A entrada contém um único caso de teste, onde é passada uma string S em que $3 \leq |S| \leq 50$.

Saída

Para o caso de teste passado, imprima uma linha contendo a quantidade de PINs diferentes que o aplicativo poderá gerar módulo 1000000007.

Exemplo

Entrada	Saída
ABC	2
12345	44

Notas

No primeiro caso, temos apenas duas possibilidades: BCA e CAB.

No segundo caso temos um total de 44 possibilidades. Algumas delas são: 23451, 21543 e 53421.

Problema L

Presente de Natal Palindrômico

Limite de tempo: 1s

Autor: Arthur Komatsu

Victória é fascinada por números palíndromos (números inteiros não-negativos que não mudam quando lidos de trás pra frente). Exemplo de números palíndromos são 0, 3, 11, 101, 1221 e 12321. Victória ganhou um número inteiro n no Natal. Ela ficou imaginando: será que existem dois números palíndromos cuja soma dá n ?

Entrada

A única linha da entrada consiste de um número inteiro n ($0 \leq n \leq 10^{12}$).

Saída

Imprima, em uma linha, quaisquer dois números inteiros não-negativos palíndromos a e b cuja soma seja igual a n . Estes números devem estar separados por espaço.

Caso não existam a e b que satisfaçam essa condição, uma linha com o valor “-1” deverá ser impressa.

Exemplo

Entrada	Saída
0	0 0
10	1 9
21	-1
1234567	65956 1168611

Notas

No primeiro caso de teste, os números palíndromos 0 e 0 somam 0.

No segundo caso de teste, os números palíndromos 1 e 9 somam 10.

No terceiro caso de teste, não existe nenhum par de números palíndromos cuja soma resulta em 21.

No quarto caso de teste, os números palíndromos 65956 e 1168611 somam 1234567.