

VIII Maratona de Programação do IFB

Caderno de Problemas

18 de novembro de 2023



(Este caderno contém 12 problemas)

Comissão Organizadora:

Daniel Saad Nogueira Nunes (IFB)
Edson Alves da Costa Júnior (UnB/FGA)
Guilherme Novaes Ramos (UnB)
Vinicius Ruela Pereira Borges (UnB)
Alberto Tavares Duarte Neto
Bruno Vargas de Souza
Enzo Yoshio Niho
Henrique de Oliveira Ramos
Jeremias Moreira Gomes

Apoio:



Lembretes

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

C/C++

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

Java

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

Python

- Tenha cuidado ao selecionar a versão correta na submissão.

Problema A

Adição e Subtração

Limite de tempo: 3s
Limite de memória: 256MB

Autor: Daniel Saad Nogueira Nunes

Jeremias é um rapaz que gosta muito de resolver desafios. A única coisa que ele gosta mais do que isso é elaborar desafios para outras pessoas resolverem. Certo dia ele propôs o seguinte desafio: dada uma sequência de números, é possível formar uma expressão aritmética, apenas envolvendo sinais de soma e subtração, sem alterar a ordem da sequência, de modo a obter um número específico? Formalmente: dada a sequência de n números, v_1, \dots, v_n , é possível construir uma expressão $s_1v_1s_2v_2 \dots s_nv_n$, em que s_i é um sinal de $+$ ou de $-$, de modo a que o resultado seja igual a um número m ?

Deixe Jeremias muito contente e resolva o seu desafio.

Entrada

A primeira linha da entrada possui dois inteiros, n e m , separados por um espaço. No caso, n representa o tamanho da sequência de inteiros e m o inteiro alvo. A próxima linha possui n inteiros, v_1, v_2, \dots, v_n , separados por um espaço.

Restrições:

- $1 \leq n \leq 20$
- $-10^9 \leq m \leq 10^9$
- $1 \leq v_i \leq 10^5, 1 \leq i \leq n$.

Saída

Imprima uma linha com a mensagem **SIM**, caso seja possível compor uma expressão aritmética com a sequência de números envolvendo apenas sinais de soma ou subtração cujo resultado seja igual a m . Imprima uma linha com **NAO**, caso contrário.

Exemplo

Entrada	Saída
5 5	SIM
1 2 3 4 5	
4 5	NAO
1 3 6 8	
3 -14	SIM
5 1 10	

Notas

No primeiro exemplo, é possível compor a expressão $1 + 2 + 3 + 4 - 5$ e obter 5.

No segundo exemplo, não é possível construir uma expressão que satisfaça o objetivo.

No último exemplo, podemos formar a expressão $-5 + 1 - 10$ e obter -14 .

Problema B

Maximize a permutação

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Henrique Ramos

Alberto e Henrique estão em busca de patrocinadores para o cenário de Maratonas de Programação no DF. Recentemente, os dois ficaram muito felizes, pois uma empresa interessada no apadrinhamento comunicou que entraria na empreitada, desde que o grupo fosse capaz de mostrar seu valor.

Para testar a turma de maratonistas, os dirigentes da empresa enviaram dois números: N e K , e pediram para que fosse montada a sequência de N elementos $A = [1, 2, 3, \dots, k, (k-1), (k-2), \dots, (n-k)]$. Sabendo que a quantidade de inversões em uma sequência é a quantidade de pares de índices (i, j) , $1 \leq i < j \leq n$, tais que $A[i] > A[j]$, os dirigentes desafiaram os estudantes a encontrar a permutação lexicograficamente máxima de A cuja quantidade de inversões não é maior que a quantidade de inversões de A .

Nota: uma sequência s é lexicograficamente maior que uma sequência t se t é um prefixo de s ou se para o primeiro i tal que $s_i \neq t_i$, $s_i > t_i$,

Entrada

A primeira e única linha contém N e K , ($K \leq N < 2K$), $1 \leq K \leq 2 \cdot 10^5$.

Saída

Imprima uma permutação de A , com N números, onde a quantidade de inversões não supera a quantidade de inversões de A e seu valor lexicográfico é o maior possível.

Exemplo

Entrada	Saída
4 3	2 1 2 3
6 4	3 2 1 2 3 4

Notas

Para o primeiro caso de teste: $N = 4, K = 3$ O vetor $A = [1, 2, 3, 2]$ possui uma inversão: $(1, 4)$ e o vetor $B = [2, 1, 2, 3]$ também possui apenas a inversão $(2, 4)$.

No segundo caso de teste $N = 6, K = 4$ O vetor $A = [1, 2, 3, 4, 3, 2]$ possui 4 inversões $(3, 6), (4, 5), (4, 6), (5, 6)$ e o o vetor $B = [3, 2, 1, 2, 3, 4]$ possui 4 inversões $(1, 2), (1, 3), (1, 4), (2, 3)$. Em ambos casos, pode ser provado que as respostas são lexicograficamente máximas

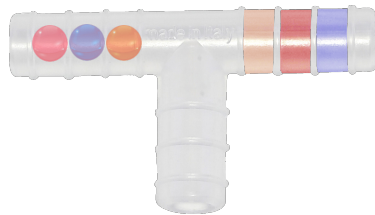
Problema C

Conectado

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Jeremias Moreira Gomes

A empresa Inesquecível Fábrica de Brinquedos adora criar novos quebra-cabeças para mentes de diversas idades. Dessa vez, eles desenvolveram um jogo chamado Conectado, que é muito intrigante e divertido.



O jogo é composto de três tubos vedados de um lado conectados pelo outro lado (que é aberto) para formar um “T”, como na figura. O jogo funciona da seguinte forma: inicialmente todas as bolas coloridas se encontram na posição horizontal a esquerda (parte inicial) e o objetivo é mover todas as bolas para o lado direito (parte final), sendo uma bola por vez, casando as cores. Além disso, pode-se utilizar o espaço vertical central para reservar bolas durante as movimentações. Os movimentos permitidos no jogo são:

- *M*: mover uma bola da parte inicial para a parte final;
- *G*: guardar uma bola da parte inicial no espaço reservado; e
- *R*: recuperar uma bola do espaço reservado para a parte final.

Apesar da fabricação em massa, os engenheiros responsáveis pela produção estão desconfiados que algumas versões do jogo não podem ser resolvidas. Você foi contratado para verificar e gerar soluções para o jogo Conectado.

Entrada

A entrada contém 3 linhas. A primeira linha contém um inteiro N ($1 \leq N \leq 10^5$) que indica o número de cores do jogo. A segunda linha da entrada contém N strings S ($1 \leq |S| \leq 30$), separadas por espaço, que lista as posições iniciais de cada bola. A terceira linha contém S strings, também separadas por espaços, que indicam a posição final delas para vencer o jogo.

Saída

A saída deve conter uma linha com a lista de movimentações necessárias para vencer o jogo ou a frase “Defeito de fabrica!” se isto não for possível.

Exemplo

Entrada	Saída
3 vermelho azul laranja laranja vermelho azul	GMMR
4 rosa vermelho verde azul rosa verde vermelho azul	MGMRM
5 verde-oliva rosa ocre cinza vermelho vermelho ocre cinza verde-oliva rosa	Defeito de fabrica!

Problema D

Daileon

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Daniel Saad Nogueira Nunes

“Daileon!”, bradou o incrível Jaspion para invocar o seu gigante guerreiro para combater as forças do mal com seu magnífico raio cósmico. Apesar disso, Daileon não respondeu, e Jaspion ficou no “prego”.

Então, Jaspion ligou para o seu seguro intergalático para saber por que a sua nave mãe, e incrível guerreiro, não atendeu o seu chamado. O seguro falou que Daileon estava fora da área de cobertura da rede intergalática. O atendente explicou que entre Jaspion e Daileon, existem vários repetidores de sinal, os quais propagam o chamado para a nave, e que, quando não é possível a comunicação, Daileon não responde.

Os repetidores de sinal, o transmissor de Jaspion, e o receptor de Daileon, possuem um raio de cobertura e conseguem se comunicar se a área de cobertura, induzida por este raio, possui alguma interseção com outra área de cobertura, mesmo que essas áreas apenas se toquem.

Ajude o nosso incrível Jaspion a verificar se ele conseguirá se comunicar com Daileon ou não.

Entrada

A primeira linha possui um inteiro n e um real r , que indicam, respectivamente, o número de repetidores e o raio de cobertura dos repetidores e dos comunicadores de Jaspion e Daileon.

A próxima linha possui coordenadas reais (a, b) , que indicam a posição do guerreiro Jaspion.

Em seguida, as posições reais (c, d) do guerreiro Daileon são informadas.

Por fim, seguem n linhas, cada qual com uma dupla de reais (x_i, y_i) , informando a posição de cada repetidor.

Restrições:

- $0 \leq n \leq 1000$
- $0 < r \leq 100$
- $-10^3 \leq a, b, c, d, x_i, y_i \leq 10^3, 1 \leq i \leq n$.

Saída

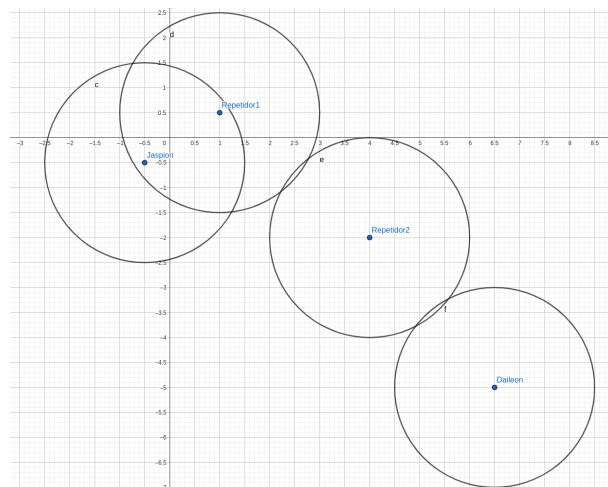
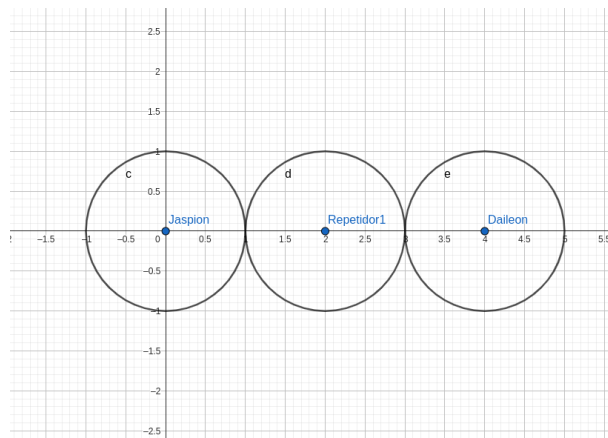
Imprima uma linha com “SIM” se Jaspion consegue se comunicar com Daileon e “NAO”, caso contrário.

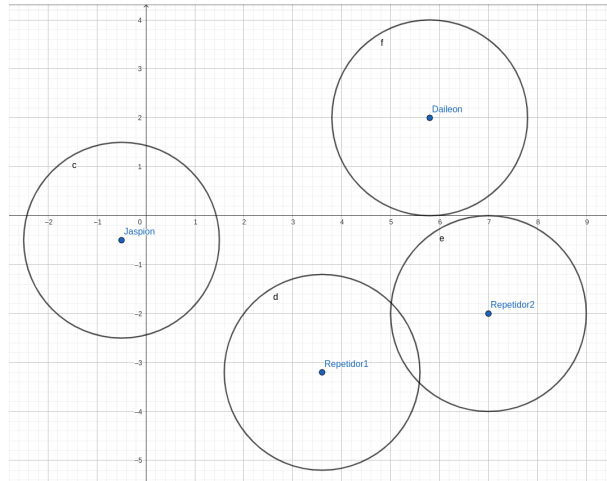
Exemplo

Entrada	Saída
1 1.0 0.0 0.0 4.0 0.0 2.0 0.0	SIM
2 2.0 -0.5 -0.5 6.5 -5.0 1.0 0.5 4.0 -2.0	SIM
2 2.0 -0.5 -0.5 5.8 2.0 3.6 -3.2 7.0 -2.0	NAO

Notas

Os três exemplos de entrada podem ser visualizados abaixo.





É considerado que um círculo possui interseção com outro se a distância entre dois pontos quaisquer deles for menor ou igual a 10^{-3} . Também considera-se que Jaspion pode se comunicar diretamente com Daileon, sem o uso de repetidores, desde que eles estejam próximos o suficiente.

Problema E

Escolhendo a equipe do IFB

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Edson Alves

O professor Saad, técnico das equipes que representam o IFB – campus Taguatinga – na Maratona de Programação, deseja montar uma nova equipe para o ano de 2024. Há N estudantes interessados em fazer parte do nova equipe. A cada um deles foram atribuídos três índices inteiros, no intervalo de zero a 100, que representam três habilidades fundamentais: algoritmos ('A'), codificação ('C') e trabalho em equipe ('E'). Saad irá montar a equipe com os três estudantes que apresentarem os melhores índices em uma destas habilidades, a ser determinada pelo professor. Caso dois ou mais estudantes tenham índices iguais, o critério de desempate será o nome do estudante, em ordem alfabética.

Auxilie Saad escrevendo um programa que, dados os valores de N , a habilidade escolhida pelo professor e os nomes e índices dos estudantes, determine a composição da equipe.

Entrada

A primeira linha da entrada contém os valores do inteiro N ($3 \leq N \leq 100$) e do caractere H ($H \in \{A, C, E\}$), separados por um espaço em branco, representando o número de estudantes e a habilidade escolhida pelo professor, respectivamente.

Cada uma das N linhas seguintes contém quatro informações, separadas por um espaço em branco: o nome do estudante (s_i) e os índices do estudante em algoritmos (a_i), codificação (c_i) e trabalho em equipe (e_i), com $0 \leq a_i, c_i, e_i \leq 100$. Assuma que s_i é composto de, no mínimo, 2 e de, no máximo, 20 caracteres alfabéticos, sendo o primeiro deles maiúsculo e os demais minúsculos. É garantido que todos os estudantes têm nomes distintos.

Saída

Imprima os nomes dos três estudantes que farão parte da equipe, um em cada linha, do estudante com maior índice na habilidade escolhida pelo professor Saad para o menor.

Exemplo

Entrada	Saída
5 A	Carlos
Ana 70 85 50	Beto
Beto 80 70 65	Davi
Carlos 85 70 70	
Davi 75 50 80	
Erick 50 80 80	
5 C	Ana
Ana 70 85 50	Erick
Beto 80 70 65	Beto
Carlos 85 70 70	
Davi 75 50 80	
Erick 50 80 80	
5 E	Davi
Ana 70 85 50	Erick
Beto 80 70 65	Carlos
Carlos 85 70 70	
Davi 75 50 80	
Erick 50 80 80	

Notas

No primeiro caso, a habilidade escolhida pelo professor Saad foi algoritmos. Dentre os 5 estudantes, o com maior índice nesta habilidade é Carlos (85), seguido por Beto (80) e Davi (75).

No segundo caso, a habilidade é codificação: Ana tem o maior índice (85), seguida por Erick (80). Para a terceira e última vaga na equipe, há um empate entre dois estudantes, ambos com índices iguais a 70: Beto e Carlos. Usando o critério de desempate (ordem alfabética dos nomes), Beto é o escolhido.

Problema F

Feijão, Feijão

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Guilherme Ramos

Pra fazer economia, você foi morar numa pensão onde a bóia (todo dia!) era só, feijão, feijão. Durante a semana inteira, você com o coração na mão quando via as compras lá da feira: uma porção de cestas só de feijão. Reclamar com a Teresa, não adiantava não pois até na sobremesa vinha doce de feijão! Domingos, e feriados variava a refeição, vinham ovos estrelados, carne seca (no feijão)...

Dado o prato do dia, indique se a leguminosa está presente ou não.

Entrada

A entrada consiste de um string com não mais de 100 caracteres descrevendo a refeição do dia. Não há caracteres especiais.

Saída

Se houver feijão na refeição, apresente a mensagem “Era so feijao feijao...”. Caso contrário, apresente “Variou a refeicao!”

Exemplo

Entrada	Saída
Feijao preto com quiabo	Era so feijao feijao...
sopa de nabo	Variou a refeicao!
FAROFA COM TUTU DE FEIJAO	Era so feijao feijao...

Problema G

Geoplano

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Edson Alves

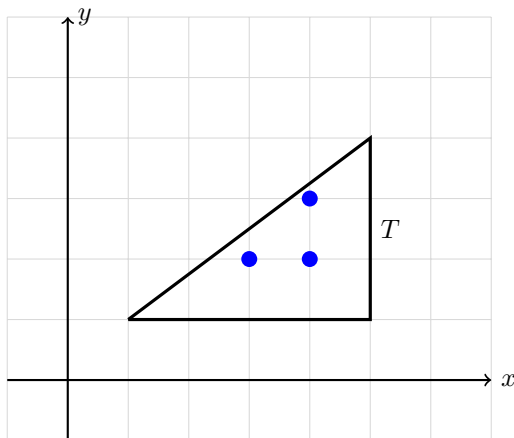
Uma ferramenta simples, prática e útil no ensino de Geometria é o geoplano, que pode ser construído artesanalmente por qualquer estudante ou professor. Basta ter uma placa de madeira, pregos, lápis, régua e um martelo, e seguir os passos abaixo:

1. com a régua e o lápis, marque $N \times M$ pontos igualmente espaçados, de modo a formar uma malha com N linhas e M colunas; e,
2. em cada um destes pontos, use o martelo para fixar um prego. O ideal é que, após fixados, todos os pregos tenham a mesma altura em relação a placa.

Uma vez finalizado o geoplano, é possível usar ligas de elástico para representar diferentes figuras geométricas: segmentos de reta, triângulos, quadrados e diferentes polígonos.

Para familiarizar os estudantes com o geoplano e com as características de um triângulo, o professor Saad passou a seguinte tarefa: dadas as coordenadas de três pregos distintos, não colineares, o estudante deve determinar quantos pregos ficaram dentro do triângulo delimitado pelo elástico que envolve os três pregos indicados pelo professor. Atenção: não devem ser contabilizados os pregos que encostam no elástico!

Por exemplo, considere os pregos localizados nas coordenadas $(1, 1)$, $(5, 1)$, $(5, 4)$, conforme mostra a figura abaixo:



Os três pontos azuis representam os pregos que ficaram dentro do triângulo. Observe que os pregos localizados em $(1, 1)$ e $(5, 3)$, por exemplo, tocam no elástico (na figura representado pela linha que delimita o triângulo), e portanto não devem ser contabilizados.

Como são muitos estudantes e muitos exercícios para corrigir, auxilie o professor Saad escrevendo um programa que receba as coordenadas dos três pregos e que retorne a contagem dos pontos interiores.

Entrada

A entrada é composta por três linhas. Cada uma delas traz as coordenadas de um dos três pregos x_i e y_i ($1 \leq x_i, y_i \leq 10^9, 1 \leq i \leq 3$), separados por um espaço em branco. É garantido que os três pontos não são colineares, isto é, eles definem um triângulo não-degenerado.

Saída

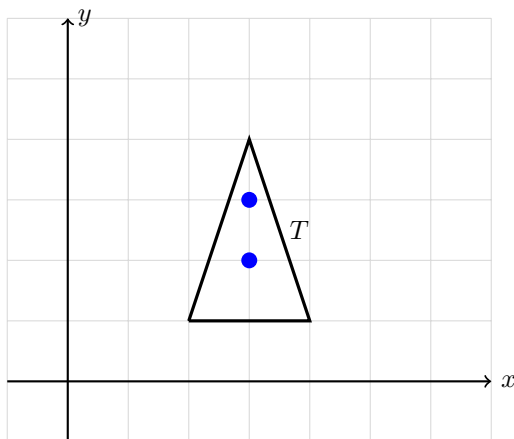
Imprima, em uma linha, o número de pregos que ficaram no interior do triângulo delimitado pelos pregos dados na entrada.

Exemplo

Entrada	Saída
1 1 5 1 5 4	3
2 1 4 1 3 4	2
4 12 8 3 16 5	39

Notas

O primeiro caso foi ilustrado na figura presente no texto do problema. No segundo caso, temos o seguinte cenário:



Problema H

Horta

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Vinicius Borges

Um agricultor preparou uma lavoura em uma fazenda em N pedaços de terra para plantação de alfaces. Cada pedaço possui área para receber exatamente um pé de alface. Momentos depois, o agricultor ficou sabendo que rúculas e coentros seriam plantados no mesmo local.

Isso demandou uma alteração no planejamento do agricultor. Nas suas observações, ele verificou que uma rúcula ocupa metade da área da terra normalmente ocupada pelo alface e que o coentro ocupa metade da área que é ocupada pela rúcula.

Para garantir uma produtividade diversificada, o agricultor deve plantar, pelo menos, K mudas de cada tipo de hortaliça (alface, rúcula e coentro). Determine a quantidade máxima de hortaliças que poderão ser plantadas na horta.

Entrada

A entrada consiste de uma única linha contendo dois números inteiros separados por espaço em branco N e K ($3 \leq N \leq 10^9, 1 \leq K \leq \frac{N}{3}$), indicando a quantidade de pedaços de terra na hora e a quantidade mínima de hortaliças que devem, obrigatoriamente, ser plantadas para cada tipo.

Saída

Imprima um único número inteiro contendo a resposta para o problema - a quantidade máxima de hortaliças que poderão ser plantadas na horta.

Exemplo

Entrada	Saída
3 1	8
4 2	8
6 3	12

Notas

No primeiro exemplo de teste, é melhor plantar 1 alface, 1 rúcula e 6 coentros para se chegar à resposta do problema e respeitando as restrições dos pedaços de terra.

Problema I

Igreja Invisível

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Guilherme Ramos

João estava perdido quando passou pela porta da Igreja Invisível e sentiu “o chamado”. O sucesso de sua existência agora está ligado ao exercício da fé, e seus esforços nesta nova vida logo foram recompensados com habilidades incríveis: o Pastor João consegue transformar as coisas!

Ele ainda não domina completamente seus poderes, visto que a transformação inverte a caixa de cada um dos elementos envolvidos... Além disso, as transformações mais recentes têm prioridade, e em caso de transformar um elemento mais curto em um elemento mais longo, a caixa segue a última letra do primeiro.

Apesar das dificuldades, ele fervorosamente tenta colaborar com sua comunidade. Ajude-o a atender aos pedidos de todas as almas sem lar que o procuram.

Entrada

A entrada consiste de uma série de casos de teste, e termina com EOF. Primeiramente são listadas uma ou mais transformações possíveis para ele no formato ‘*X em Y*’ - cada uma significa que o Pastor João transforma *X* em *y*, onde ambas as incógnitas são palavras únicas e sem acentos ou espaços e com não mais de 100 caracteres cada. As linhas seguintes contêm, cada uma, o pedido de um fiel, na forma de uma única palavra (sem acentos e com até 100 caracteres). Sempre há pelo menos um fiel na porta da Igreja Invisível a espera do Pastor João.

Saída

Para cada pedido, apresente o resultado da transformação em uma linha. Caso Pastor João não consiga realizar a proeza, apresente a mensagem ‘`Ainda existe “impossivel”!`’.

Exemplo

Entrada	Saída
agua em vinho	VINHO
chao em ceu	CEU
pau em pedra	MEL
cuspe em mel	PEDRA
agua	
chao	
cuspe	
pau	
sal em pimenta	sAL
pimenta em sal	ACucar
SAL em acucar	
Pimenta	
saL	
x em y	Ainda existe "impossivel"!
a	

Notas

O primeiro exemplo apresenta 4 transformações possíveis e outros 4 pedidos. O segundo exemplo tem 3 transformações e 2 pedidos e o último um de cada. Note que nos dois primeiros, a caixa não é considerada para reconhecer a transformação mas é para realizá-la!

Problema J

Jogo da Velha

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Daniel Saad Nogueira Nunes

O jogo da velha é jogado por dois jogadores em um tabuleiro com dimensões 3 por 3. Cada jogador detém um símbolo, **O** ou **X**, e eles colocam esses símbolos em posições vazias do tabuleiro alternadamente. Ganha o jogador que completar uma linha, coluna, ou diagonal, com seus três símbolos. Caso o tabuleiro fique cheio e não haja um vencedor, é declarado um empate. Faça um programa que leia um tabuleiro e determine o resultado do jogo.

Entrada

A primeira linha da entrada possui um inteiro n , contendo o número de casos de testes. Cada caso de teste possui três linhas, cada uma com três símbolos, que podem assumir os valores de **O**, **X** ou **B**, este último representando uma casa do tabuleiro que não foi preenchida ainda. Existe uma linha em branco entre dois casos de teste.

É garantido que cada caso de teste sempre representa uma configuração válida ao final do jogo da velha. Também é garantido que, no caso de uma configuração de empate, todas as casas do tabuleiro estarão preenchidas.

Restrições:

- $1 \leq n \leq 10^3$.

Saída

Imprima uma linha com:

- **O**, caso o jogador detentor do símbolo **O** tenha ganhado o jogo.
- **X**, caso o vitorioso seja o jogador detentor do símbolo **X**.
- **velha**, caso o jogo tenha terminado em empate.

Exemplo

VIII Maratona de Programação do IFB

Entrada	Saída
4	0
OXX	0
X00	X
X00	velha
X00	
X0X	
OX0	
0XB	
BX0	
BXB	
X00	
OXX	
X00	

Problema K

Kátia e o Mahjong Numérico

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Edson Alves

Kátia é aficcionada no jogo Mahjong ao ponto de propôr, como tema do seu Trabalho de Conclusão de Curso, uma variante unidimensional e numérica do jogo: o Mahjong Numérico.

Nesta variante, os números de 1 a N são dispostos em uma sequência de tamanho $2N$, onde cada número aparece exatamente duas vezes. A cada rodada, o jogador pode eliminar da sequência um número que esteja em posição adjacente à sua cópia, isto é, os elementos a_i e a_{i+1} podem ser excluídos da sequência a se, e somente se, $a_i = a_{i+1}$, com $i = 1, 2, \dots, 2N - 1$. Uma vez que os elementos são removidos, a sequência

$$a = \{ a_1, a_2, a_3, \dots, a_{i-1}, a_i, a_{i+1}, a_{i+2}, \dots, a_{2N} \}$$

se torna a sequência

$$b = \{ b_1 = a_1, b_2 = a_2, b_3 = a_3, \dots, b_{i-1} = a_{i-1}, b_i = a_{i+2}, \dots, b_{2N-2} = a_{2N} \},$$

de tamanho $2N - 2$. O jogador vence a partida se, após N rodadas, ele obtiver uma sequência vazia. Se, em uma dada rodada, o jogador não puder eliminar nenhum número, então ele perde a partida.

Kátia está com dificuldades em implementar o protótipo do jogo. Ajude-a, escrevendo um programa que recebe os valores de N e dos elementos da sequência inicial a e retorne “Sim”, caso o jogador possa vencer a partida, ou “Nao”, caso não exista nenhuma sequência de jogadas válidas que leve o jogador à vitória.

Entrada

A primeira linha da entrada contém o valor do inteiro N ($1 \leq N \leq 10^5$).

A segunda linha contém $2N$ inteiros a_i ($1 \leq a_i \leq N, 1 \leq i \leq 2N$), separados por um espaço em branco, sendo que, para cada valor $v \in [1, N]$, há exatamente dois índices distintos j e k tais que $a_j = a_k = v$.

Saída

Imprima, em uma linha, a mensagem “Sim”, caso existe ao menos uma sequência de escolhas de remoções de números que levem o jogador à vitória, ou a mensagem “Nao” (sem acento), caso contrário.

Exemplo

Entrada	Saída
3	Sim
2 3 3 2 1 1	
2	Nao
1 2 1 2	

Notas

No primeiro caso, o jogador pode, na primeira rodada, eliminar o número 3 (que ocupa as posições 2 e 3, respectivamente), obtendo a sequência

2 2 1 1

Na segunda rodada ele pode eliminar o número 1, obtendo a sequência 2 2 e, por fim, na última rodada, eliminar o 2, vencendo a partida.

No segundo caso, já na primeira rodada não há nenhum número que possa ser eliminado pelo jogador, de modo que ele perde a partida.

Problema L

Ligações

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Vinicius Borges

Uma empresa de serviço de atendimento ao consumidor (SAC) está verificando a jornada de trabalho de seus operadores com a finalidade de evitar problemas. Para esse propósito, ela selecionou um operador qualquer. Esse operador recebe N ligações de clientes durante a sua jornada de trabalho, sendo que os clientes são enumerados de 1 a N . Sua função é sempre atender os clientes em ordem de chegada, desde que não esteja em atendimento.

Em sua jornada de trabalho, o operador de SAC precisa cumprir exatamente M minutos em ligações com os clientes, não considerando, portanto, o tempo em que está ocioso aguardando o próximo cliente. Dados os momentos em que os N clientes ligam para o SAC e as respectivas durações das ligações, determine de quantas maneiras distintas o operador pode cumprir sua jornada de trabalho.

Entrada

A primeira linha da entrada contém dois inteiros N e M ($1 \leq N \leq 100, 1 \leq M \leq 50$), indicando a quantidade de clientes e a jornada de trabalho, respectivamente.

As próximas N linhas descrevem os clientes. A i -ésima linha apresenta dois inteiros t_i e d_i ($1 \leq t_i, d_i \leq 25$) indicando o momento (minuto) da ligação e a duração da chamada do i -ésimo cliente, respectivamente.

Saída

Imprima um único inteiro indicando de quantas maneiras distintas o operador pode cumprir sua jornada de trabalho em exatamente m minutos.

Exemplo

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

Notas

No primeiro exemplo de teste, o operador consegue cumprir a jornada de trabalho de três maneiras distintas: quando atende apenas os clientes 1 e 2, ou os clientes 2 e 3 ou apenas o cliente 4.

No segundo exemplo de teste, o operador pode atender os clientes 1 e 4, ou ainda 2 e 4.

No terceiro exemplo de teste, não há como o operador cumprir a jornada de trabalho.