

A. Kobus odeia sorteios

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

No meio do ano realizamos a Seletiva IME-USP, uma prova que seleciona quais times vão representar a USP Butantã nas competições oficiais. Nessa prova, fica claro que as pessoas do MaratonUSP são muito competitivas, afinal de contas, somos um grupo de programação competitiva.

Nesta edição, as provas estão sendo realizadas individualmente, e não mais em times de 3 pessoas. Após o fim da prova, com o placar geral já disponível, Kobus percebeu que empatou em primeiro lugar na seletiva com outros N participantes, que são da POLI. Como não aceitamos nada a menos que o melhor — ainda mais contra politécnicos — ela quer ficar em primeiro lugar. Mas, infelizmente, o critério de desempate é sorteio.

Para realizar o sorteio, Nathan vai pegar a string $A = abcdefghijklmnopqrstuvwxyz$, que representa o alfabeto, e trocar algumas letras de lugar, obtendo uma nova string A' . Em outras palavras, A' é uma permutação de A . Usando essa nova string como ordem alfabética, Nathan vai então ordenar os nomes, e o nome que ficar em primeiro lugar é o ganhador.

Kobus tem acesso ao laboratório secreto onde o Nathan estuda e, com isso, ela pode alterar A' . Ajude Kobus a ficar em primeiro lugar ou diga se isso é impossível.

Input

A primeira linha contém um inteiro N ($1 \leq N \leq 1000$) — a quantidade de participantes que empataram com a Kobus em primeiro lugar.

Cada uma das próximas N linhas contém uma string de caracteres minúsculos s_i ($1 \leq |s_i| \leq 100$) — o nome do i -ésimo participante empatado com Kobus. É garantido que todas as pessoas participando da competição tem nomes distintos.

Output

Imprima uma string A' ($|A'| = 26$) que é uma permutação de A que faz com que Kobus fique em primeiro lugar. Se varias respostas existirem, imprima qualquer uma.

Caso não seja possível que Kobus fique em primeiro lugar, imprima "sem chance" (sem aspas).

Examples

input
3 lamarca kob gabriel
output
sem chance

input
5 fabiana marcos roberto

julia maquiesperto
output
abcdefghijklmnopqrstuvwxy

Note

A ordenação de uma lista de palavras utiliza a ordem relativa entre duas palavras. Podemos chamar essa ordem relativa de ordem lexicográfica. Uma palavra x_1, x_2, \dots, x_a é lexicograficamente menor que a palavra y_1, y_2, \dots, y_b se uma das duas condições abaixo é válida:

- $a < b$ e $x_1 = y_1, x_2 = y_2, \dots, x_a = y_a$, isto é, a primeira palavra é um prefixo da segunda; ou se
- existe uma posição j ($1 \leq j \leq \min(a, b)$), tal que $x_1 = y_1, x_2 = y_2, \dots, x_{j-1} = y_{j-1}$ e $x_j < y_j$, isto é, na primeira posição em que as palavras diferem, a letra da primeira palavra naquela posição é menor do que a letra da segunda palavra naquela mesma posição (a letra da primeira palavra *vem antes no alfabeto* do que a letra da segunda palavra).

Por exemplo, seja a lista de palavras dada por $\{aaa, aa, ba\}$. Se o alfabeto for $abcd \dots z$, então a lista ordenada é $\{aa, aaa, ba\}$. Porém, se o alfabeto for $bacd \dots z$ então a lista ordenada vira $\{ba, aa, aaa\}$.

B. Guidi quer ficar monstro

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

O sedentarismo e a perda de massa corporal é um male que afeta os alunos universitários, inclusive nós, da maratona. Para ficar em forma e evitar problemas de saúde, Guidi comprou um kit de barras e anilhas junto com um banco de supino reclinável para ficar monstro.

Como bom maratonista, Guidi pediu para o grande mago Eterom criar uma dieta baseada no algoritmo simplex e uma rotina de exercícios otimizada com técnicas combinatórias. A rotina de exercícios é uma sequência de movimentos, onde cada movimento é representado por um caractere. Vale ressaltar que a ordem em que cada movimento é executada é essencial para a efetividade do exercício.

Guidi, um competidor astuto e arдил, não quer deixar seus concorrentes descobrirem o treino do grande mago. Para esconder seu treino, ele abriu um guia de musculação do Centro de praticas esportivas da USP (CEPE-USP) e teve uma grande ideia: camuflar seu treino otimizado dentro de alguma sequência de movimentos do CEPE!

Por exemplo, seja $soads$ a sequência de movimentos do treino do grande mago e $tsasdas$ a sequência de movimentos do treino do CEPE. A melhor sequência de exercícios que Guidi consegue fazer de forma a esconder seu treino dentro de uma sequência de movimentos do treino do CEPE é $sads$ (supino, agachamento, deadlift e supino).

Dadas as sequências de movimento do grande mago Eterom e do CEPE, ajude Guidi a descobrir o tamanho da maior sequência de movimentos que ele pode fazer de forma a esconder seu treino sagrado.

Input

Duas linhas com uma string cada, A e B ($1 \leq |A|, |B| \leq 1000$) — a sequência de movimentos sugerida pelo grande mago Eterom e a sequência de movimentos do CEPE, respectivamente. As strings são compostas por letras minúsculas.

Output

Um inteiro N — o tamanho da maior sequência de movimentos que Guidi pode fazer de forma a esconder seu treino otimizado.

Examples

input
soads tsasdas
output
4

input
babanad lalaxadnad
output
5

C. Harada e os números da sorte

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

As pessoas que participam do mundo de programação competitiva compartilham de uma mesma paixão, números. Porém, alguns indivíduos tem preferências, como números naturais, números inteiros, números transcendentais, entre outros.

Harada não é diferente, possuindo certa preferência por alguns números, que ele considera números da sorte. Em particular, Harada tem uma lista de Q números da sorte.

No dia de seu aniversário, Harada ganhou um lindo grafinho e um baralho com N cartas, onde cada uma dessas cartas contém um número inteiro entre 0 e 9. Agora que a festa de aniversário acabou, Harada ficou curioso para saber qual o maior subconjunto da sua lista de números da sorte ele conseguiria formar utilizando as cartas de seu novo baralho.

Por exemplo, suponha que a lista de números da sorte seja $[23, 111, 112]$ e que o baralho possua 6 cartas, quatro cartas com o número 1, uma carta com o número 2 e uma carta com o número 3. Então, ele conseguiria formar os seguintes subconjuntos: $\{23\}$, $\{111\}$, $\{112\}$, $\{23, 111\}$, com $\{23, 111\}$ sendo o maior deles.

Porém, querendo evitar a má sorte — afinal de contas, não se brinca com os próprios números da sorte! — Harada pediu sua ajuda para realizar esta tarefa.

Input

A primeira linha contém dois inteiros Q e N ($1 \leq Q \leq 7$), ($1 \leq N \leq 100$) — o número de números da sorte na lista de Harada e o número de cartas no baralho que ele ganhou, respectivamente.

A segunda linha contém Q números distintos — onde q_i ($0 \leq q_i \leq 10^9$) é o i -ésimo número da sorte na lista de Harada. É garantido que nenhum número da lista possui zeros á esquerda.

A terceira e ultima linha contém N números — onde n_i ($0 \leq n_i \leq 9$) é o número da i -ésima carta no baralho que Harada ganhou.

Output

Na primeira linha, imprima um inteiro R — o tamanho do maior subconjunto dos números da sorte que Harada consegue formar.

Na segunda linha, imprima R números distintos — onde r_i é o i -ésimo número da sorte do maior subconjunto.

A ordem em que os números são impressos na segunda linha não importa, e se existir mais de um subconjunto, imprima qualquer um.

Examples

input
3 6 23 111 112 1 2 3 1 1 1
output
2 23 111

input
6 10 400 289 22 0 20 24 9 5 0 0 1 2 4 2 0 8
output
3 400 289 0

D. Mashup do Corona

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Como todo grupo muito organizado, o MaratonUSP gosta de se preparar com bastante antecedência. Assim, Lamarca ficou encarregado de organizar um Mashup nos próximos 10^{18} dias. Por ser um contest em time, e nenhum time querer ser prejudicado por ter poucos maratonheiros, a quantidade de pessoas que vai participar no contest deve ser múltipla de 3.

Lamarca fez uma enquete no Telegram com os 10^{18} dias, mas ele esqueceu de permitir aos participantes que votassem em múltiplas opções. Contudo, ao conversar entre si, as pessoas perceberam que todos que podiam participar em um dia x também podiam participar em todos os dias depois de x .

Ao escolher uma data, TODAS as pessoas que podem participar naquele dia vão aparecer, mas, caso o número de pessoas não seja múltiplo de 3, um time ficará triste e haverá caos total. Claramente, Lamarca não quer o caos e também gostaria de escolher o melhor dia para o Mashup.

Ajude o Lamarca a descobrir este dia! O melhor dia é o dia em que vai a maior quantidade de pessoas e o contest ainda acontece, isto é, o dia em que vai a maior quantidade de pessoas e que nenhum time fica triste. Ou diga se não existe uma data com essas características.

Input

A primeira linha contém um único inteiro N ($3 \leq N \leq 10^6$) — representando o número de participantes do contest.

A segunda contém N inteiros a_i ($1 \leq a_i \leq 10^{18}$) — a_i representa a data a partir da qual o i -ésimo membro do MaratonUSP pode participar do mashup.

Output

Imprima um único inteiro D , o melhor dia para o contest. Se não tiver data válida, imprima -1 .

Examples

input
8 1 9 4 7 8 4 9 9
output
4

input
4 1 2 3 3
output
-1

E. Aprendendo novas linguagens

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Baledoc ingressou na faculdade recentemente e está preparado para se tornar o melhor programador do mundo. Em um dos eventos da faculdade, ele fez um site utilizando #React, #GraphQL, #Typescript, #Jest, #Golang, #Postgres, #Kafka, #Blockchain, #Machine #Learning, #Docker e #Kubernetes. No final do projeto, ele não sabia muito bem o que o site fazia, mas ficou maravilhado com todas as coisas novas que aprendeu e precisa aprender #mais.

Ele notou que o caminho para ser o melhor programador do mundo começa em aprender todas as linguagens, por isso decidiu ir até a livraria e comprar todos os dicionários que encontrasse.

Chegando em casa, ele notou que dicionários não são a melhor forma de aprender uma linguagem nova, mas ele não vai #desistir. Convenientemente, dada a sua perseverança, ele já entende algumas palavras de cada uma das linguagens dos dicionários que ele comprou.

Toda palavra no dicionário é definida em termos de outras palavras. Baledoc consegue entender uma palavra nova quando ela pode ser definida em termos de palavras que ele já entende, para isso, ele precisa entender todas as palavras que a definem. Ajude Baledoc a descobrir quantas palavras ele var conseguir entender no total.

#Submit2Learn #Learn1000000007Submit

Input

A primeira linha da entrada contém um inteiro n ($1 \leq n \leq 1000$) — a quantidade de palavras que Baledoc já conhece. A segunda linha contém as n palavras k_i ($1 \leq |k_i| \leq 10$) conhecidas por Baledoc, separadas por espaço.

A terceira linha da entrada contém um inteiro m ($0 \leq m \leq 100$) — a quantidade de palavras no dicionário. As próximas $2m$ linhas contém informações sobre o dicionário, 2 linhas para cada palavra.

Para cada palavra, a primeira linha contém uma string s ($1 \leq |s| \leq 10$) e um inteiro k ($1 \leq k \leq 10$) — a palavra que é definida no dicionário e a quantidade de palavras usadas na sua definição, respectivamente. A segunda linha contém as k palavras s_i ($1 \leq |s_i| \leq 10$) usadas para definir s , separadas por espaço.

Todas as palavras da entrada possuem apenas letras minúsculas de a até z .

Output

Um único inteiro, a quantidade de palavras que Baledoc consegue entender no total.

Examples

input
9 you one more of a or cat have than 3 two 4 one more than one cats 6 two or more of a cat i 1 you
output
12

input
3 a b d 3 abacaba 3 aba c aba abadaba 3 aba d aba aba 3 a b a
output
5

Note

Para o primeiro caso, Baledoc inicialmente entende 9 palavras, ("you", "one", "more", "of", "a", "or", "cat", "have", "than") e consegue entender todas as palavras no dicionário. Ele entende "two" e "i" pois entende todas as palavras necessárias, e agora que ele entende "two", consegue entender "cats", que usa "two" em sua definição.

F. Confundindo o Morete

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Morete é um aluno de Ciências Moleculares que tem se dedicado muito para a Maratona, participando de muitos contests do Codeforces. Estes contests são provas com duração aproximada de 2 horas em que o participante tem que resolver uma determinada quantidade de problemas, ganhando ou perdendo *rating* (pontuação na plataforma) dependendo de seu desempenho.

Ao final de cada contest, Morete escreve em um cartão quanto ganhou ou perdeu de *rating*, e em seguida, coloca este cartão em cima de um monte, formando uma pilha com seus cartões. Morete faz isso para controlar o seu desempenho.

Thiago, o seu arqui-inimigo, com inveja da dedicação de Morete, decide atrapalhá-lo e para isso, insere um único cartão com um número qualquer, em algum lugar da pilha de cartões de *rating*. No final do mês, quando Morete decide analisar o seu desempenho, nota que, em algum momento, ficou com saldo negativo de *rating*, o que o deixou confuso, pois pelo que se recorda, sempre ficou com um saldo não-negativo de *rating*.

Sua tarefa é ajudar o Morete a descobrir quais são as cartas suspeitas de terem sido inseridas no monte por Thiago, ou dizer se Morete chapou e errou as contas.

Por exemplo, seja $[500, -100, -400, 100, -500, 200]$ os números escritos nos cartões da pilha com os *ratings* do Morete após Thiago ter inserido seu cartão, onde o número no começo da lista está na base da pilha, e o do fim, no topo. Então, os cartões suspeitos seriam os com valores -400 e -500 (índices 3 e 5, respectivamente), pois, qualquer um deles, se ignorado, faz com que Morete sempre tenha um *rating* não-negativo.

Input

A primeira linha contém um inteiro N ($2 \leq N \leq 10^5$) — o número de cartões na pilha, após Thiago ter inserido o seu.

A segunda linha contém N inteiros — onde r_i ($-10^9 \leq r_i \leq 10^9$) é o valor do i -ésimo cartão da pilha.

Output

Caso em nenhum momento Morete tenha ficado com saldo negativo, imprima "morete chapou: errou conta" (sem aspas).

Caso não exista pelo menos um cartão, que se ignorado, Morete sempre fique com saldo não-negativo, imprima "morete chapou: ficou com saldo negativo" (sem aspas).

Caso contrário, na primeira linha, imprima k — a quantidade de cartões suspeitos. Na segunda linha, imprima k inteiros — as posições desses cartões na pilha, indexados a partir de 1 e em qualquer ordem.

Examples

input
6 500 -100 -400 100 -500 200
output
2 3 5

input
4 200 -180 20 80
output
morete chapou: errou conta!

input
7 1100 -700 60 -700 200 -700 2000
output
morete chapou: ficou com saldo negativo!

G. Jogo dos pontinhos azuis

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Os alunos do IME-USP são fanáticos por jogos, um jogo muito popular no instituto é o jogo dos pontinhos azuis. Nesse jogo, n alunos se reúnem em uma roda, de forma que cada aluno pode ver os demais $n - 1$ alunos, e cada um tem pintado em sua testa um pontinho que pode ser azul ou marrom. O objetivo do jogo é descobrir quais alunos tem um ponto azul na testa, sendo que, inicialmente, ninguém sabe a cor do próprio ponto. O jogo funciona em rodadas, quando um aluno descobre todos os alunos que tem um ponto azul ele se retira do círculo no final da rodada. O jogo termina quando todos os alunos se retiraram do círculo.

Na variante atual é garantido que ao menos um aluno tem um ponto de cor azul.

Dessa vez, os alunos decidiram iniciar uma partida logo antes da aula! Os professores não querem atrapalhar o jogo pois é uma tradição importante, mas querem saber quantos turnos o jogo vai durar. Lembre-se que os alunos do IME-USP são bem inteligentes e sempre terminarão o jogo no menor tempo possível.

Input

A primeira linha contém um inteiro n ($1 \leq n \leq 2 * 10^5$) — o número de alunos no jogo.

A segunda linha contém n inteiros a_1, \dots, a_n ($a_i \in \{0, 1\}$) — se $a_i = 0$ então o aluno tem um ponto marrom; se $a_i = 1$ então o aluno tem um ponto azul. É garantido que ao menos um aluno possui um ponto azul.

Output

Um único inteiro, o número mínimo de turnos que o jogo vai durar.

Examples

input
1 1
output
1

input
2 0 1
output
2

H. O comediante Nathan

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A Cidade Universitária é um lugar incrível, com muitas árvores, uma raia olímpica, e, o mais importante, capivaras. Além disso, o campus possui uma extensa área, com mais de $3700000m^2$. Para facilitar a locomoção dos estudantes, a USP dispõe de uma frota de ônibus circulares, que é responsável por levar os alunos através do campus.

Hoje, Nathan acordou muito feliz e decidiu levar alegria para os seus colegas universitários. Naturalmente, ele decidiu contar piadas no circular. Nathan entende que, para que a pessoa saia alegre do circular, ela não pode ouvir uma piada mais de uma vez. Além disso, é necessário que haja um intervalo de 5 minutos entre uma piada e outra.

Como Nathan é muito amigo do motorista, ele conseguiu uma lista com N pares (e_i, s_i) , o horário de entrada e saída do i -ésimo passageiro. Note que, múltiplos passageiros podem entrar ou sair em um determinado momento. Ademais, como os alunos estão quase sempre atrasados, se Nathan contar uma piada no mesmo instante que um passageiro esta saindo, este passageiro não vai ouvir a piada, porém, se Nathan contar uma piada no mesmo momento que alguém esta entrando, esta pessoa irá ouvir a piada.

Devido ao caráter inerentemente fofoqueiro dos universitários, as pessoas que estão no circular em um mesmo momento podem compartilhar as piadas que ouviram, e Nathan também quer levar isso em conta. Analogamente, um passageiro que esta saindo não consegue compartilhar uma piada com um passageiro que esta entrando no circular no mesmo momento.

Para simplificar as coisas, vamos assumir que Nathan contou a sua primeira piada no instante $t = 0$, a segunda em $t = 5$, a terceira em $t = 10$, e assim por diante. Além disso, como a universidade não faz muita questão de respeitar as leis de trânsito, um número ilimitado de passageiros podem estar no circular em um dado momento. Como o circular gasta exatamente 100 minutos para completar o seu trajeto, é garantido que nenhum passageiro fica nele por mais de 100 minutos, exceto Nathan, pois ele está determinado em sua tarefa.

Com a lista dos horários de entrada e saída dos passageiros em mãos, Nathan quer preparar seu repertório de piadas. Porém, preparar as piadas leva tempo, e ele não quer gastar mais tempo que o necessário, e, por isso, pediu sua ajuda para encontrar o número mínimo de piadas que ele precisa preparar para que seus objetivos sejam atingidos.

Input

As primeira linha contém um inteiro N ($1 \leq N \leq 10^5$) — o número de passageiros do circular.

Seguem N linhas com dois inteiros, e_i e s_i ($0 \leq e_i < s_i, \leq 10^9$), ($1 \leq s_i - e_i \leq 100$) — os horários de entrada e saída do i -ésimo passageiro.

Output

Imprima um único inteiro — a menor quantidade de piadas que Nathan precisa preparar para o seu repertório.

Example

input
4
2 7
8 13
6 9
21 25

output
2

I. Mario Kart Competitivo

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Depois de muita insistência por parte dos seus amigos, Raphael "Raphael com 'ph'" Ribeiro finalmente comprou a versão para Mintendo Troca-Troca do icônico Mario Kart, jogo de corrida pouco convencional. Nele os competidores passam por pistas em diferentes cenários do mundo Mintendo acumulando pontos dependendo da sua colocação. O primeiro colocado receberia 15 pontos, o segundo 12, o terceiro 10, o quarto 9, e assim por diante até o décimo segundo lugar, que recebe 1 ponto.

Depois de muito treinar, Raphael percebeu que ele não só conseguia ganhar todas as corridas, mas também era capaz de ficar em qualquer colocação que quisesse. Ao descobrir das habilidades de Raphael, Noronha, um expert de Mario Kart, logo se interessou em lançar um desafio para ele. Dada uma pontuação N , se Raphael conseguisse atingi-lá no menor número possível de corridas no Mario Kart, então Noronha premiaria N dinheiros para o Raphael. Caso isso não acontecesse, então seria Raphael quem ficaria numa dívida de N dinheiros com Noronha.

Mesmo estando confiante de suas habilidades, Raphael não quer ter chance de perder seu dinheiro, por isso ele pediu sua ajuda! Escreva um programa que descubra uma sequência de colocações que ele deve atingir no Mario Kart para conseguir exatamente N pontos de forma que ela seja a menor possível!

Input

Um único inteiro N ($1 \leq N \leq 10^6$) — a soma de pontos que Raphael quer atingir.

Output

A primeira linha deve conter um único inteiro S — o tamanho da menor sequência de colocações para que Raphael consiga exatamente N pontos.

A segunda deve conter S inteiros p_i ($1 \leq p_i \leq 12$) — a colocação que Raphael deve atingir na i -ésima corrida, em qualquer ordem.

Examples

input
28
output
3 1 2 12

input
11
output
2 3 12

input
100
output
7 1 1 1 1 1 1 3

J. Raphael cantor

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Além de participar da Maratona e jogar Softball, Raphael também tem um outro hobby: cantar. Como ele é muito eclético e está sempre variando seu estilo, lança novos álbuns quase todos os anos, com gêneros variando entre Indie, Rock, Eletrônica, Rap, e outros.

Porém, Raphael não é muito criativo para dar nomes aos seus álbuns. Como ele faz aniversário no dia 1º de Janeiro, decidiu que todo álbum lançado teria como título a sua idade naquele ano.

Alguns fãs, inconformados que Raphael fosse tão jovem na capa de um álbum, decidiram investigar se Raphael tinha mentido sua idade em algum dos títulos. Em outras palavras, dado o ano de lançamento de cada álbum e seu respectivo título, os fãs decidiram investigar se a idade era consistente em todos eles.

Como esta tarefa é muito difícil, os fãs decidiram pedir a sua ajuda. Por favor, ajude-os nesta tarefa essencial.

Input

A primeira linha contém um inteiro N ($1 \leq N \leq 100$) — o número de álbuns lançados por Raphael.

Cada uma das próximas N linhas contém um par de inteiros a_i e t_i ($1900 \leq a_i \leq 2021$), ($0 \leq t_i \leq 200$) — o ano e título do i -ésimo álbum lançado.

Output

Caso Raphael tenha mentido sua idade em algum álbum imprima "mentiu a idade" (sem aspas). Caso contrario, imprima "idades corretas" (sem aspas).

Examples

input
5 2020 28 2000 8 2003 11 2001 9 2015 23
output
idades corretas

input

3
2000 15
1998 11
2010 25

output

mentiu a idade