
Problem A. Ordenando Palavras

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Escreva um programa que ordene uma sequência de palavras de acordo com o seu tamanho. Em caso de empate, o programa deverá preservar a ordem da entrada, isto é, no caso de palavras de mesmo tamanho, a que apareceu primeiro tem precedência sobre a que apareceu depois.

Input

A primeira linha da entrada possui um inteiro n ($1 \leq n \leq 10^5$). As próximas n linhas, possuem cada, uma palavra constituída apenas com letras minúsculas e com no máximo 10 caracteres.

Output

Imprima as palavras de acordo com o critério estabelecido.

Examples

standard input	standard output
5 abra cadabra pe de cabra	pe de abra cabra cadabra
5 e d c b aaaa	e d c b aaaa
9 o rato roeu a roupa do rei de roma	o a do de rei rato roeu roma roupa

Problem B. Fantástica Fábrica de Fibonacci

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

A Fantástica Fábrica de Fibonacci é responsável pela maior produção de chocolates do mundo. Para produzir os chocolates, eles utilizam uma fórmula mágica, em que os ingredientes são despejados em quantidades presentes na sequência de Fibonacci. Por exemplo, para produzir um chocolate de 15 gramas, podem ser utilizados ingredientes de peso 2, 8 e 5.

Uma linha especial de chocolates dessas fábricas possui um convite dourado cujo detentor poderá visitar a fábrica para conhecer as suas instalações. Estes chocolates possuem uma propriedade especial: além das quantidades de ingredientes utilizados pertencerem à sequência de Fibonacci, estas quantidades nunca são consecutivas, com respeito à sequência.

Buscando minimizar custos, o administrador da fábrica, William, quer saber o número mínimo de ingredientes para fabricar os chocolates que possuirão o convite dourado.

Input

A primeira linha da entrada possui um inteiro n ($1 \leq n \leq 10^5$), que representa o número de consultas a serem respondidas.

As próximas n linhas possuem, cada, um inteiro q_i ($1 \leq q_i \leq 10^{18}$), que indica o peso do chocolate especial que deve ser fabricado.

Output

Para cada consulta, imprima uma linha com o número mínimo de ingredientes utilizados para fabricação do chocolate especial.

Example

standard input	standard output
3	1
1	1
5	2
15	

Problem C. Números Distintos

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Dada uma sequência de n inteiros, verifique a quantidade de números distintos que ela possui.

Input

A primeira linha da entrada possui um inteiro n ($1 \leq n \leq 10^5$), que descreve o tamanho da sequência de inteiros. A próxima linha possui n inteiros, v_1, \dots, v_n , ($1 \leq v_i \leq 10^9$), separados por um espaço, descrevendo cada, um elemento da sequência.

Output

Imprima o número de inteiros distintos.

Examples

standard input	standard output
5 2 3 2 2 3	2
5 1 2 3 4 5	5
5 5 5 5 5 5	1

Problem D. Espetinho do Barbosinha

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes

O *Espetinho do Barbosinha* se tornou bastante popular na cidade do Gama devido ao baixo preço e alta qualidade das “jantinhas” que são oferecidas.

Em um certo momento, Barbosinha se deparou com um problema. A procura por suas “jantinhas” era tão grande que ele ficou preocupado em não conseguir atender todos os seus clientes de maneira adequada devido à falta de cadeiras. Neste esquema, cada cliente poderia, através de um aplicativo, inserir o horário estimado de chegada e saída. Dessa forma, Barbosinha saberia de antemão o número de cadeiras que ele precisaria deixar disponível e tomar as devidas providências.

Como o formulário preenchido pelos clientes obteve um grande número de respostas, Barbosinha não estava conseguindo mensurar corretamente o número mínimo de cadeiras que ele necessitaria e pediu ajuda a você para automatizar esta tarefa. Dado os horários estimados de chegada e saída dos clientes em um determinado dia, você deverá construir um programa que compute o número mínimo de cadeiras que Barbosinha precisará providenciar aos seus clientes. Considere que uma cadeira está reservada a cada cliente durante a sua estadia e portanto não pode ser usada por outro cliente neste período.

Input

A primeira linha da entrada possui um inteiro N ($1 \leq N \leq 10^5$) contendo o número de clientes que preencheram o formulário.

As N linhas seguintes contém, cada uma, duas strings, separadas por espaço, no formato **HH:MM:SS**, indicando respectivamente o tempo de chegada e saída de cada cliente. Considere que as horas seguem o padrão 24h e que um cliente não pode preencher o tempo de saída em uma dia diferente do dia de chegada.

Output

Seu programa deverá imprimir o número mínimo de cadeiras que deverão ser disponibilizadas para atender todos os clientes.

Examples

standard input	standard output
2 12:00:00 12:30:00 13:00:00 13:30:00	1
3 10:00:00 10:30:00 10:30:01 11:00:00 10:00:00 12:00:00	2
6 08:00:00 10:00:00 12:00:00 16:00:00 09:00:00 13:00:00 18:00:00 23:00:00 15:00:00 20:00:00 08:00:00 22:00:00	3

Note

No primeiro caso de teste, os dois clientes não estão no espetinho do Barbosinha no mesmo momento, portanto apenas 1 cadeira é necessária.

No segundo caso de teste, o primeiro e segundo clientes não se topam, mas o terceiro cliente estará no espetinho do Barbosinha em um momento em comum com o primeiro e segundo clientes.

No terceiro caso de teste, existirão momentos em que 3 clientes estarão no espetinho do Barbosinha simultaneamente: especificamente das 9h às 10h, ou das 12h às 13h e das 15h às 16h, ou das 18h às 20h. Qualquer outro horário possuirá uma quantidade menor de clientes.

Problem E. F1

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

No treino classificatório para a formação do grid de largada da Fórmula 1, os pilotos percorrem a extensão do circuito o mais rápido possível. O tempo da volta é registrado por sensores localizados em três pontos específicos da pista, as três parciais que somadas determinam o tempo da volta.

Larga na *pole position* o piloto que completou a volta em menor tempo. A segunda posição fica com o segundo mais rápido, e assim por diante. Em caso de empate no tempo, cuja precisão é de milésimo de segundo, o primeiro piloto a ter registrado a marca terá a primazia.

Dados os tempos das três parciais dos N pilotos que participaram da sessão classificatória, determine o grid de largada da corrida.

Fato curioso: no GP da Europa de 1997, que decidiria o campeão da temporada, Jacques Villeneuve e Michael Schumacher estavam separados na tabela por apenas um ponto. No treino classificatório, os dois marcaram exatamente o mesmo tempo: 1:21.072! Se não bastasse a coincidência, Heinz-Harald Frentzen, companheiro de Villeneuve na Williams, largou em terceiro, com o mesmo tempo de 1:21.072 que resultou na pole de Villeneuve!

Input

A primeira linha da entrada contém o número N ($1 \leq N \leq 20$) de pilotos que participaram do treino classificatório. Na sequência, são das as informações da melhor volta no treino classificatório para cada piloto, na ordem em que foram registradas.

As informações de cada piloto ocupam 2 linhas cada. A primeira delas contém o nome do piloto, formado por uma string de, no máximo, 30 caracteres alfabéticos maiúsculos ou minúsculos. A segunda linha contém o tempo de cada uma das parciais do piloto, separados por um espaço em branco. O tempo é dado no formato **ss:mmm**, com $0 \leq ss \leq 59$ e $0 \leq mmm \leq 999$.

Output

Imprima N linhas, onde a i -ésima linha contém a posição, o nome e o tempo de volta do i -ésimo piloto mais rápido. A posição de largada deve ser seguida de um ponto final e de um espaço; o nome deve ser seguido por um espaço, um traço e um espaço; e o tempo de volta deve seguir o formato **M:ss.mmm**, onde **M** é dado em minutos, **ss** em segundos (com dois dígitos) e **mmm** em milisegundos (com três dígitos).

Examples

standard input	standard output
5 Verstappen 20.252 25.654 21.602 Albon 20.380 25.816 21.739 Hamilton 20.310 25.725 21.664 Vettel 20.289 25.699 21.643 Bottas 20.362 25.792 21.720	1. Verstappen - 1:07.508 2. Vettel - 1:07.631 3. Hamilton - 1:07.699 4. Bottas - 1:07.874 5. Albon - 1:07.935
3 Villeneuve 26.754 26.753 27.565 Schumacher 25.943 25.944 29.185 Frentzen 28.375 26.753 25.944	1. Villeneuve - 1:21.072 2. Schumacher - 1:21.072 3. Frentzen - 1:21.072

Note

A saída do primeiro caso corresponde ao grid do GP do Brasil de 2019. Os tempos das parciais, porém, é fictício.

Problem F. Índice-h

Input file: **standard input**
 Output file: **standard output**
 Time limit: 1 second
 Memory limit: 256 megabytes

O *índice-h* é utilizado para mensurar o impacto da produção científica de autores levando em consideração não só o número de publicações, mas também o número de citações de cada publicação.

Simplificadamente, índice-h de um determinado autor tem valor h caso ele possua h publicações com pelo menos h citações cada. Formalmente, o índice-h é o maior valor de h que satisfaz estas condições.

Dada uma lista de autores associada a suas publicações e citações, faça um programa que liste estes autores em ordem decrescente de índice-h. Em caso de empate, a ordem lexicográfica dos nomes dos autores deve ser considerada.

Input

A primeira linha da entrada possui um inteiro N ($1 \leq N \leq 10^3$), que indica a quantidade de autores.

As próximas $2N$ linhas estão divididas em conjuntos de duas linhas e descrevem as informações relativas a cada autor.

A primeira linha de cada conjunto contém uma palavra, com até 30 caracteres maiúsculos, que corresponde ao sobrenome do autor, e um inteiro P ($1 \leq P \leq 10^3$), que descreve o número de publicações deste autor, separados por um espaço.

A segunda linha de cada conjunto contém P inteiros $C_1, C_2, \dots, C_i, \dots, C_P$ ($0 \leq C_i \leq 10^5$), separados por um espaço, de modo que C_i indica o número de citações associada a i -ésima publicação do autor correspondente.

Output

Imprima a lista de autores juntamente com o índice-h de cada um deles, separados por um espaço, e em ordem decrescente de índice-h, considerando as observações do enunciado.

Examples

standard input	standard output
2 ADLEMAN 3 1 0 2 RIVEST 2 3 2	RIVEST 2 ADLEMAN 1
3 SILVA 2 0 0 FLOYD 4 2 5 19 17 NAVARRO 4 1000 2345 954 8	NAVARRO 4 FLOYD 3 SILVA 0
2 KNUTH 3 1 2 3 KARP 3 3 2 1	KARP 2 KNUTH 2

Note

No primeiro exemplo, Adleman possui uma publicação com pelo menos uma citação, mas não possui duas publicações com pelo menos duas citações. Já Rivest, possui duas publicações com pelo menos duas citações, logo, Rivest tem índice-h 2 enquanto Adleman tem índice-h 1.

O segundo exemplo é análogo.

O terceiro exemplo possui dois autores com índice-h 2, por conta disso, Karp deve ser apresentado antes de Knuth na lista de autores de acordo com a ordem lexicográfica.

Problem G. Observe o equilíbrio

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Sejam a e b dois números inteiros. Dizemos que b divide a se existe ao menos um inteiro c tal que $a = bc$.

Uma sequência de números inteiros x_1, x_2, \dots, x_N é dita equilibrada se, para quaisquer dois elementos x_i, x_j da sequência, ou x_i divide x_j ou x_j divide x_i .

Dada uma sequência de inteiros, determine se ela é ou não equilibrada.

Input

A primeira linha da entrada contém o valor do inteiro N ($2 \leq N \leq 2 \times 10^5$).

A segunda linha da entrada contém N inteiros x_i ($-10^9 \leq x_i \leq 10^9, 1 \leq i \leq N$), separados por um espaço em branco.

Output

Imprima, em uma linha, a mensagem “Sim” se a sequência x_1, x_2, \dots, x_N é equilibrada, ou a mensagem “Nao”, caso contrário.

Examples

standard input	standard output
3 0 1 2	Sim
3 12 4 6	Nao
2 -7 -7	Sim

Note

No primeiro caso, 1 divide qualquer inteiro, e qualquer inteiro divide zero. Deste modo, a sequência é equilibrada.

No segundo caso, 4 divide 12, 6 divide 12 mas 4 não divide 6 e nem 6 divide 4. Portanto, a sequência não é equilibrada.

No terceiro caso, qualquer inteiro divide a si mesmo.