regionals | latin america region
icpc international collegiate programming contest
icpc.foundation

Maratona de Programação da SBC 2025

Sub-Regional Brasil do ICPC

13 de Setembro de 2025

Sessão de Aquecimento

Informações Gerais

Este caderno contém 4 problemas; as páginas estão numeradas de 1 a 4, não contando esta página de rosto. Verifique se o caderno está completo.

Este conjunto de problemas também está sendo utilizado simultaneamente nas seguintes competições: The 2025 ICPC Gran Premio de Centroamerica, The 2025 ICPC Gran Premio de Mexico e Competencia Boliviana de Programación.

Leia atentamente as instruções a seguir. Informações adicionais, incluindo os limites de tempo dos problemas estarão em um documento disponível no sistema utilizado para a competição, o BOCA.

A) Sobre os nomes dos programas
1) Para soluções em C/C++ e Python, o nome do arquivo-fonte não é significativo, pode ser qualquer nome.
2) Se sua solução é em Java, ela deve ser chamada *codigo*`.java` onde *codigo* é a letra maiúscula que identifica o problema. Lembre que em Java o nome da classe principal deve ser igual ao nome do arquivo.
3) Se sua solução é em Kotlin, ela deve ser chamada *codigo*`.kt` onde *codigo* é a letra maiúscula que identifica o problema. Lembre que em Kotlin o nome da classe principal deve ser igual ao nome do arquivo.

B) Sobre a entrada
1) A entrada de seu programa deve ser lida da *entrada padrão*.
2) A entrada é composta de um único caso de teste, descrito em um número de linhas que depende do problema.
3) Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
4) Cada linha, incluindo a última, contém exatamente um caractere final-de-linha.
5) O final da entrada coincide com o final do arquivo.

C) Sobre a saída
1) A saída de seu programa deve ser escrita na *saída padrão*.
2) Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
3) Cada linha, incluindo a última, deve conter exatamente um caractere final-de-linha.

Promoção:

SBC Sociedade Brasileira de Computação

Problema A

Atenção à Reunião

Vinicius está em uma reunião da diretoria do Instituto de Consultoria de Palestras e Comentários (ICPC) pensando que seria muito bom se os membros da diretoria fossem mais concisos, e mantivessem suas falas dentro do tempo definido para cada diretor, para que a reunião terminasse antes do almoço. Infelizmente, talvez devido à natureza da instituição, todos gostam muito de falar.

Sabendo que

- há N diretores que irão falar na reunião;
- cada diretor irá falar pelo mesmo tempo;
- e entre duas falas consecutivas há um intervalo de 1 minuto,

determine a duração máxima de cada fala, em minutos, para que a reunião dure no máximo K minutos.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 100$), o número de diretores. A segunda linha contém um inteiro K ($1 \leq K \leq 1000$ e $K \geq N$), a duração máxima da reunião em minutos. Para todos os casos de entrada, a fala de cada diretor tem duração de pelo menos 1 minuto.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, indicando a duração da fala de cada membro da diretoria, em minutos.

Exemplo de entrada 1	**Exemplo de saída 1**
7 120	16

Explicação do exemplo 1:

Há 7 diretores e a duração máxima da reunião é 120 minutos. Se cada diretor falar por 16 minutos, temos $16 \times 7 = 112$ minutos. Como há seis intervalos entre as falas, e cada intervalo tem duração de um minuto, ao todos temos 118 minutos. Note que, nesse caso, dois minutos do tempo da reunião não são utilizados, e que, se as falas fossem superiores a 16 minutos, o tempo total superaria o limite de 120 minutos.

Exemplo de entrada 2	**Exemplo de saída 2**
1 10	10

Explicação do exemplo 2:

Há apenas um diretor e a duração da reunião é 10 minutos. Assim, a duração máxima da fala do diretor é 10 minutos.

Exemplo de entrada 3	**Exemplo de saída 3**
100 1000	9

Problema B

Interativo

Este problema é interativo.

Existe um número escondido N. Seu objetivo é descobrir N realizando no máximo 10 perguntas.

Em cada pergunta, você deve informar um número X. O sistema responderá se X é menor, igual ou maior que N.

Interação

Você pode fazer no máximo 10 perguntas para descobrir o número escondido N ($1 \leq N \leq 1000$). Cada pergunta deve ser feita no formato: "`?` X" (sem aspas), em que $1 \leq X \leq 1000$.

Após cada pergunta, estará disponível para leitura uma linha contendo:

- `<`, se $N < X$.
- `=`, se $N = X$.
- `>`, se $N > X$.

Quando tiver certeza do valor de N, imprima "`!` N" (sem aspas) e termine o programa.

O interator **não é adaptativo**, ou seja, o número escondido N é fixo durante toda a interação.

Após cada escrita na saída, não se esqueça de realizar o *flush* do *buffer*. Caso contrário, você poderá receber o veredito `TIME LIMIT EXCEEDED`. Para fazer isso, utilize:

- `fflush(stdout)` em `C`.
- `cout.flush()` em `C++`.
- `sys.stdout.flush()` em `Python`.
- `System.out.flush()` em `Java`.
- `System.out.flush()` em `Kotlin`.

Exemplo de interação 1

Leitura	Escrita
	`? 2`
`>`	
	`? 5`
`=`	
	`! 5`

Explicação do exemplo 1:

Neste caso, o número escondido era 5.

Problema C

Bacon Number

Carlinhos adora filmes, e recentemente tem estado fascinado com o número de Bacon, mais conhecido como *Bacon Number*, que é definido da seguinte forma.

- O número de Bacon do ator Kevin Bacon é igual a 0;
- Se o menor número de Bacon de um ator com quem X tenha aparecido em um mesmo filme for b, o número de bacon do ator X é $b + 1$.

Ou seja, o número de Bacon mede o menor caminho entre qualquer ator e o ator Kevin Bacon, em que dois atores são conectados se eles apareceram juntos em um mesmo filme.

Carlinhos está interessado em um problema mais geral: dados dois atores, como conectá-los através de filmes e atores intermediários? São dados N filmes, e, para cada filme, quais dos M atores existentes atuaram nele. Carlinhos quer responder Q consultas: na i-ésima delas, queremos computar alguma forma de conectar o ator x_i com o ator y_i. Devemos achar alguma sequência $x_i = a_1, f_1, a_2, f_2, \ldots, f_{k-1}, a_k = y_i$, em que $1 \leq a_j \leq N$ são atores e $1 \leq f_j \leq M$ são filmes, e o ator a_j atuou nos filmes f_{j-1} e f_j, ou indicar que não existe tal sequência.

Entrada

Na primeira linha da entrada, são fornecidos dois inteiros N ($1 \leq N \leq 100$) e M ($1 \leq M \leq 10^6$), o número de filmes e o número de atores. Seguem N linhas. Na i-ésima delas, o primeiro inteiro n_i ($1 \leq n_i \leq M$) denota o número de atores no filme i. Seguem n_i números em ordem crescente separados por espaço: os índices, de 1 a M, dos atores que atuaram no filme i. Na próxima linha, leia um número Q ($1 \leq Q \leq 10^4$): o número de consultas. As próximas Q linhas descrevem as consultas. Na i-ésima delas, leia dois números x_i, y_i ($1 \leq x_i \neq y_i \leq M$), os atores que queremos conectar. É garantido que o número total de atores nos filmes é no máximo 10^6. Isto é, $\sum_i n_i \leq 10^6$.

Saída

Para cada uma das consultas, se não existe sequência, imprima uma linha com -1. Caso contrário, imprima duas linhas. Na primeira, o número de atores k_i ($2 \leq k_i \leq 10^6$) em alguma maneira de conectar x_i e y_i. Na segunda, imprima a sequência como descrita, com k_i atores e $k_i - 1$ filmes, de maneira alternada. Se houver mais de uma maneira de conectar os atores, imprima qualquer uma delas.

Exemplo de entrada 1	**Exemplo de saída 1**
4 6 3 1 2 5 3 1 3 5 2 2 4 1 6 4 1 5 1 4 3 4 1 6	2 1 1 5 3 1 1 2 3 4 4 3 2 1 1 2 3 4 -1

Problema D

Altura Mínima

Carlitos é um entusiasta de aventuras com um amor insaciável por parques de diversões. Apesar da sua paixão vibrante, Carlitos enfrenta um desafio único: a sua estatura limitada. Enquanto planeja ansiosamente sua aventura de fim de semana, ele percebe que suas limitações verticais podem atrapalhar sua experiência no parque de diversões. Não se trata apenas de escolher um parque; trata-se de encontrar um onde ele possa aproveitar a emoção dos brinquedos.

Imagine o caleidoscópio de cores, as risadas jubilosas e a adrenalina dos passeios. Carlitos sempre foi atraído pela energia dos parques de diversões. Com o fim de semana se aproximando, ele se debruça sobre os folhetos do parque, estudando os requisitos de altura de cada passeio. O objetivo dele é maximizar sua diversão, e é aí que você entra.

Sua tarefa é ajudar Carlitos a determinar o número de passeios que ele pode desfrutar em um parque específico. Considerando sua altura e os requisitos mínimos de altura de cada passeio, oriente-o a aproveitar ao máximo sua aventura no parque de diversões.

Entrada

A primeira linha contém dois números inteiros, N e H ($1 \leq N \leq 6$ e $90 \leq H \leq 200$), que representam a quantidade de brinquedos em um parque e a altura de Carlitos em centímetros, respectivamente.

Na segunda linha da entrada, serão fornecidas as alturas mínimas $A_1, \ldots, A_N$ ($90 \leq A_i \leq 200$) de cada um dos brinquedos do parque.

Saída

Seu programa deve imprimir uma única linha contendo a quantidade de brinquedos nos quais Carlitos pode ir, ou seja, a quantidade de brinquedos para os quais a altura de Carlitos é pelo menos tão grande quanto a altura mínima necessária.

Exemplo de entrada 1	**Exemplo de saída 1**
1 100 100	1

Exemplo de entrada 2	**Exemplo de saída 2**
6 120 200 90 100 123 120 169	3