Centro de Educação Superior a Distância do
Estado do Rio de Janeiro – CEDERJ

Curso de Tecnologia em Sistemas de Computação – TSC

EAD-05.009 Fundamentos de Programação

Caderno de Exercícios
Aula 8

(Estruturas de Dados – Listas, Lista de Listas)

Professores

Dante Corbucci Filho
Leandro A. F. Fernandes

Instruções

- Utilize Python 3 e a IDE PyCharm na elaboração de soluções para os problemas propostos;
- A entrada de cada problema deve ser lida da entrada padrão (teclado);
- A saída de cada problema deve ser escrita na saída padrão (tela);
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que a saída emitida será considerada correta;
- Na saída, toda linha deve terminar com o caractere `'\n';`
- Utilize o URI Online Judge (http://www.urionlinejudge.com.br) e submeta sua solução para correção automática.

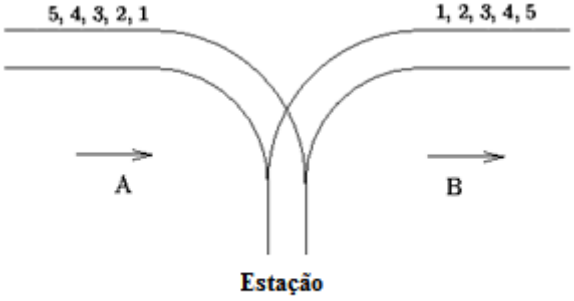
Referências Autorais

Os exercícios apresentados nesta lista foram extraídos do URI Online Judge (http://www.urionlinejudge.com.br). Acesse a URL apresentada abaixo do título de cada problema para proceder com a correção automática de sua solução e, também, para consultar a autoria do enunciado.

Problema A: Trilhos

https://www.urionlinejudge.com.br/judge/pt/problems/view/1062

Há uma famosa estação de trem na cidade PopPush. Esta cidade fica em um país incrivelmente acidentado e a estação foi criada no último século. Infelizmente os fundos eram extremamente limitados naquela época. Foi possível construir somente uma pista. Além disso, devido a problemas de espaço, foi feita uma pista apenas até a estação (veja figura abaixo).



A tradição local é que todos os comboios que chegam vindos da direção A continuam na direção B com os vagões reorganizados, de alguma forma. Suponha que o trem que está chegando da direção A tem *N* <= 1000 vagões numerados **sempre** em ordem crescente **1, 2, ..., *N***. O primeiro que chega é o **1** e o último que chega é o ***N***. Existe um chefe de reorganizações de trens que quer saber se é possível reorganizar os vagões para que os mesmos saiam na direção B na ordem $\mathbf{a_1, a_2, a_n}$..

O chefe pode utilizar qualquer estratégia para obter a saída desejada. No caso do desenho ilustrado acima, por exemplo, basta o chefe deixar todos os vagões entrarem na estação (do 1 ao 5) e depois retirar um a um: retira o 5, retira o 4, retira o 3, retira o 2 e por último retira o 1. Desta forma, se o chefe quer saber se a saída 5,4,3,2,1 é possível em **B,** a resposta seria **Yes**. Vagão que entra na estação **só pode sair para a direção B** e é possível incluir quantos forem necessários para retirar o primeiro vagão desejado.

Entrada

O arquivo de entrada consiste de um bloco de linhas, cada bloco, com exceção do último, descreve um trem e possivelmente mais do que uma requisição de reorganização. Na primeira linha de cada bloco há um inteiro *N* que é a quantidade de vagões. Em cada uma das próximas linhas de entrada haverá uma permutação dos valores **1, 2, …, *N***. A última linha de cada bloco contém apenas 0. Um bloco iniciando com zero (0) indica o final da entrada.

Saída

O arquivo de saída contém a quantidade de linhas correspondente às linhas com permutações no arquivo de entrada. Cada linha de saída deve ser **Yes** se for possível organizar os vagões da forma solicitada e **No**, caso contrário. Há também uma linha em branco após cada bloco de entrada. No exemplo abaixo, o primeiro caso de teste tem 3

permutações para 5 vagões. O último zero dos testes de entrada não deve ser processado.

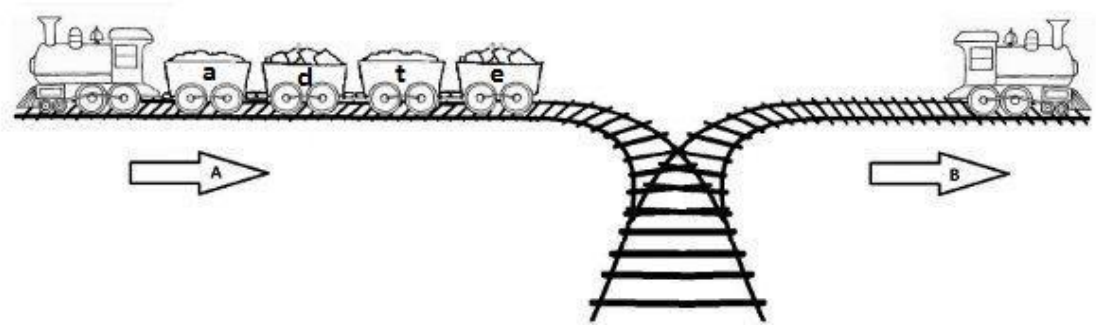
Exemplo

Entrada	Saída
5 5 4 3 2 1 1 2 3 4 5 5 4 1 2 3 0 6 1 3 2 5 4 6 0 0	Yes Yes No Yes

Problema B: Trilhos Novamente... Traçando Movimentos

https://www.urionlinejudge.com.br/judge/pt/problems/view/1063

Você lembra aquela estação de trem da cidade PopPush? Apenas para relembrar, existe uma estação de trem em um país incrivelmente acidentado. Além disso, a estação foi construída no século passado e infelizmente os fundos eram muito limitados. Em um determinado trecho foi possível construir apenas uma pista e, a solução encontrada para transportar as cargas nos dois sentidos foi construir uma estação que permitisse desconectar os vagões de uma locomotiva e conectar em outra, que iria a outro sentido.



Cada trem que chega à direção A é manobrado e seus vagões continuam na direção B, reorganizados conforme o chefe da estação deseja. Ao chegar pelo lado A, cada vagão é desconectado e vai até a estação e depois segue para a direção B, para ser conectado na segunda locomotiva. Você pode desconectar quantos trens deseja na estação, mas o vagão que entra na estação só pode sair pelo lado B e uma vez que ele sai, não pode mais entrar novamente.

Todos os vagões são identificados pelas letras minúsculas (**a** até **z**). Isto significa 26 vagões no máximo. O chefe da organização dos vagões precisa agora que você ajude a resolver para ele, através de um programa, qual a sequência de movimentos é necessária para obter a saída desejada após a entrada na estação, seguindo para a direção B. O movimento de entrada e saída da estação é descrito respectivamente pelas letras **I** e **R** (Insere e Remove). Utilizando a figura dada como exemplo, a entrada ***e***, ***t***, ***d***, ***a*** para uma saída desejada ***d***, ***a***, ***t***, ***e***, resulta nos movimentos I, I, I, R, I, R, R, R.

Entrada

A entrada consiste em vários casos de teste, onde cada caso de teste é composto por 3 linhas. A primeira das 3 linhas contém um número inteiro ***N*** que representa o número total de vagões. A segunda linha contém a sequência dos vagões que vêm do lado A e a Terceira linha contém a sequência que o chefe de organização deseja como saída para o lado B. A última linha de entrada contém apenas 0, indicando o fim da entrada.

Saída

O arquivo de saída contém a quantidade de linhas correspondente ao número de casos de teste de entrada. Cada linha de saída contém uma sequência de **I** e **R** conforme

o exemplo. Se não for possível mostrar a saída, as operações devem ser interrompidas e a mensagem "**Impossible**" deve ser impressa, com um espaço após a sequência.

Exemplo

Entrada	Saída
4 e t d a d a t e 5 o s t a p p a t o s 0	IIIRIRRR IIIIIRRR Impossible

Problema C: BRINDE FACE 2015

https://www.urionlinejudge.com.br/judge/pt/problems/view/1944

A FACE em 2015 está apoiando a terceira edição da Maratona de Programação, mas desta vez a organização solicitou sua ajuda para criar um sistema de sorteio utilizando as letras da palavra FACE. Como a feira utiliza uma proposta diferenciada e alegre, cada participante que entra na feira ganha 4 letras, uma de cada cor e em formato de bloco de madeira, conforme Figura 1, e deve inseri-las num painel. Se, no momento da inserção, as 4 letras formarem o contrário das 4 últimas letras, o visitante ganhará um brinde.

Figura 1: Entrada de FACE no painel seguido de ACEF.

Por exemplo: suponha que já tiveram 3 participantes que entraram na feira e o painel ficou da seguinte forma: F A C E E C F A A C F E A C E F. Note que sempre que o painel fica vazio, assim como no início do evento, as letras F A C E são inseridas pela organização do evento. Agora, na entrada do quarto participante, ele inseriu as letras F E C A e, com isso, receberá um brinde por fechar o contrário de A C E F. Após essa situação, o painel deve ficar F A C E E C F A A C F E.

Escreva um algoritmo que dadas as letras recebidas e inseridas pelos participantes, diga quantos participantes ganharam brindes. Lembre-se que sempre que o painel fica vazio as letras F A C E são inseridas pela organização do evento.

Entrada

A primeira linha de cada caso de teste contém um inteiro N ($1 \leq N \leq 100$), representando o número de visitantes que vão receber as letras. Em cada uma das N linhas seguintes deve ser informada a combinação das 4 letras que o visitante deseja inserir no painel, separadas por espaço.

Saída

Para cada grupo de visitantes, devem ser informados quantos destes receberão brindes.

Exemplo

Entrada	Saída
4 E C F A A C E F F E C A A F C E	2

Entrada	Saída

3 E A C F A F C E E F C A	0

Entrada	Saída
6 E C A F E C A F E C A F E C A F E C A F E C A F	6

Problema D: Fila do Supermercado

https://www.urionlinejudge.com.br/judge/pt/problems/view/2065

Hoje é a inauguração de um grande supermercado em sua cidade, e todos estão muito excitados com os baixos preços prometidos.

Este supermercado tem ***N*** funcionários que trabalham no caixa, identificados por números de 1 a ***N***, onde cada funcionário leva um determinado tempo $\mathbf{v_i}$ para processar um item de um cliente. Ou seja, se um cliente tem $\mathbf{c_j}$ itens em sua cesta, um determinado funcionário levará $\mathbf{v_i}*\mathbf{c_j}$ segundos para processar todos os itens deste cliente.

Quando um cliente entra na fila para ser atendido ele espera até que um funcionário esteja livre para atendê-lo. Se mais de um funcionário estiverem livres ao mesmo tempo, o cliente será atendido pelo funcionário de menor número de identificação. Tal funcionário só estará livre novamente após processar todos os itens deste cliente.

Há ***M*** clientes na fila para serem atendidos, cada um com um determinado número de itens na sua cesta. Dadas as informações sobre os funcionários nos caixas e os clientes, o gerente pediu sua ajuda para descobrir quanto tempo levará para que todos os clientes sejam atendidos.

Entrada

A primeira linha conterá dois inteiros N e M, indicando o número de funcionários no caixa e o número de clientes, respectivamente ($1 \leq N \leq M \leq 10^4$).

Em seguida haverá ***N*** inteiros $\mathbf{v_i}$, indicando quanto tempo leva para o **i**-ésimo funcionário processar um item ($1 \leq \mathbf{v_i} \leq 100$, para todo $1 \leq \boldsymbol{i} \leq \boldsymbol{N}$).

Em seguida haverá ***M*** inteiros $\mathbf{c_j}$, indicando quantos itens o ***j***-ésimo cliente tem em sua cesta ($1 \leq \mathbf{c_j} \leq 100$, para todo $1 \leq \boldsymbol{j} \leq \boldsymbol{M}$).

Saída

Imprima uma linha contendo um inteiro, indicando quanto tempo levará para que todos os clientes sejam atendidos.

Exemplo

Entrada	Saída
1 1 3 6	18

Entrada	Saída
1 2 1 5 3	8

Entrada	Saída
2 3 1 2 10 5 3	13

Problema E: Estacionamento Linear

https://www.urionlinejudge.com.br/judge/pt/problems/view/1523

Após muito tempo juntando dinheiro, Rafael finalmente conseguiu comprar seu carro (parcelado, é claro). Chega de pegar ônibus, agora sua vida será mais fácil. Pelo menos isso é o que ele pensava, até ouvir falar do estacionamento perto da faculdade onde ele decidiu estacionar o carro todos os dias.

O estacionamento tem apenas um corredor, com largura o suficiente para acomodar um carro, e profundidade suficiente para acomodar K carros, um atrás do outro. Como este estacionamento só tem um portão, só é possível entrar e sair por ele.

Quando o primeiro carro entra no estacionamento, o mesmo ocupa a posição próxima à parede, ao fundo do estacionamento. Todos os próximos carros estacionam logo atrás dele, formando uma fila. Obviamente, não é possível que um carro passe por cima de outro, portanto só é possível que um carro saia do estacionamento se ele for o último da fila.

Dados o horário de chegada e saída prevista de N motoristas, incluindo Rafael, diga se é possível que todos consigam estacionar e remover seus carros no estacionamento citado.

Entrada

Haverá diversos casos de teste. Cada caso de teste inicia com dois inteiros N e K ($3 \leq N \leq 10^4$, $1 \leq K \leq 10^3$), representando o número de motoristas que farão uso do estacionamento, e o número de carros que o estacionamento consegue comportar, respectivamente.

Em seguida haverá N linhas, cada uma contendo dois inteiros C_i e S_i ($1 \leq C_i, S_i \leq 10^5$), representando, respectivamente, o horário de chegada e saída do motorista i ($1 \leq i \leq N$). Os valores de C_i são dados de forma crescente, ou seja, $C_i < C_{i+1}$ para todo $1 \leq i < N$.

Não haverá mais de um motorista que chegam ao mesmo tempo, e nem mais de um motorista que saiam ao mesmo tempo. É possível que um motorista consiga estacionar no mesmo momento em que outro motorista deseja sair.

O último caso de teste é indicado quando $N = K = 0$, o qual não deverá ser processado.

Saída

Para cada caso de teste imprima uma linha, contendo a palavra “Sim”, caso seja possível que todos os N motoristas façam uso do estacionamento, ou “Nao” caso contrário.

Exemplo

Entrada	Saída
3 2 1 10 2 5 6 9 3 2 1 10 2 5 6 12 0 0	Sim Nao

Problema F: A Lista da Morte de Arya

https://www.urionlinejudge.com.br/judge/pt/problems/view/1856

Arya: *"Cersei. Walder Frey. Montanha. Meryn Trant."*

Para se manter motivada, Arya sempre lembra a lista de inimigos que ela mais odeia. O principal objetivo de sua jornada é matar todos na sua lista!

Entretanto, às vezes algum inimigo dela pode ser morto por outra pessoa. Quando ela descobre que tal inimigo morreu, ela o remove da sua lista. Além disso, Arya também pode fazer novos inimigos durante sua jornada. Quando ela faz um novo inimigo, tal inimigo é incluído na sua lista.

Arya quer matar seus inimigos um por um, na mesma ordem em que aparecem na sua lista. A qualquer momento, ela pode se perguntar quanto tempo irá levar para matar todos que estão entre dados dois inimigos. Para tal, dados dois inimigos ***a*** e ***b***, ela deve determinar quantos inimigos estão na lista entre ***a*** e ***b***, excluindo ambos. Ajude Arya respondendo tais perguntas.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 5 \times 10^4$), o número de inimigos inicialmente em sua lista.

Considere que todas as pessoas são numeradas de 1 a 10^9, inclusive. A próxima linha contém N inteiros, descrevendo a lista inicial de Arya. A próxima linha contém um inteiro Q ($1 \leq Q \leq 5 \times 10^4$), o número de operações. As próximas Q linhas descrevem as operações. Cada operação pode estar em um dos seguintes formatos:

- ***I p e*** ($1 \leq e, p \leq 10^9$): Insira a pessoa ***p*** *depois* do inimigo ***e*** na lista. É garantido que ***e*** está na lista, e ***p*** não está na lista;
- ***R e*** ($1 \leq e \leq 10^9$): Remova o inimigo ***e*** da lista. É garantido que ***e*** está na lista;
- ***Q a b*** ($1 \leq a, b \leq 10^9$): Determine quantos inimigos estão na lista entre ***a*** e ***b***, excluindo ambos. É garantido que ***a*** e ***b*** estão na lista.

Saída

Imprima uma linha para cada operação do tipo ***Q*** com sua resposta.

Exemplo

Entrada	Saída
3 3 8 2 6 Q 3 2 I 9 8 Q 3 2 R 8 I 1 2 Q 9 1	1 2 1

Problema G: A Fila de Desempregados

https://www.urionlinejudge.com.br/judge/pt/problems/view/1119

Em uma séria tentativa de reduzir a fila de desempregados, o novo Partido Nacional Trabalhista dos Rinocerontes Verdes decidiu uma estratégia pública. Todos os dias, todos os candidatos desempregados serão colocados em um grande círculo, voltados para dentro. Alguém é escolhido arbitrariamente como número 1, e os outros são numerados no sentido horário até ***N*** (os quais estarão à esquerda do 1°). Partindo do 1° e movendo-se no sentido horário, um contador oficial do laboratório conta ***k*** posições e retira um candidato, enquanto outro oficial começa a partir de ***N*** e se move no sentido anti-horário, contando ***m*** posições e retirando outro candidato. Os dois que são escolhidos são então enviados como estagiários para a reciclagem e se ambos os funcionários escolherem a mesma pessoa, ela (ele) é enviado para se tornar um político. Cada funcionário, em seguida, começa a contar novamente com a pessoa próxima disponível e o processo continua até que não reste ninguém. Note-se que as duas vítimas (desculpe, estagiários) deixam o anel ao mesmo tempo, por isso é possível que um funcionário conte a pessoa já selecionada pelo outro funcionário.

Entrada

Escreva um programa que leia sucessivamente três números (***N***, ***k*** e ***m***; ***k***, ***m*** > 0, 0 < ***N*** < 20) e determina a ordem no qual os candidatos são retirados para treinamento. Cada conjunto de três números estará em uma linha distinta e o final da entrada de dados é sinalizado por três zeros (0 0 0).

Saída

Para cada conjunto de três números de entrada, imprima uma linha de números especificando a ordem na qual as pessoas são escolhidas. Cada número pode ter até 3 dígitos. Liste o par escolhido partindo da pessoa escolhida pelo contador do sentido horário. Os pares sucessivos são separados por vírgula (mas não deverá haver vírgula após o último escolhido).

Exemplo

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

OBS: Cada * na saída representa um espaço em branco.

Problema H: Fila do Bandejão

https://www.urionlinejudge.com.br/judge/pt/problems/view/1524

Um fenômeno muito comum na fila do bandejão (também conhecido como restaurante universitário) é ver uma pessoa recém chegada entrar no interior na fila em vez de no final. Isso ocorre sempre que tal pessoa encontra alguém de seu grupo já na fila.

Interessado em estudar esse fenômeno, um amigo pediu para você escrever um programa para estudar os grupos presentes na fila. Podemos supor que existem K grupos diferentes e toda pessoa pertence a exatamente um desses grupos. O tamanho de um grupo é definido pela distância entre as duas pessoas mais distantes dentro do grupo. Se o grupo consiste de apenas uma pessoa, seu tamanho é zero. Considerando que os grupos se organizam de forma que a soma dos tamanhos dos K grupos seja mínima, seu programa deve determinar qual é o valor dessa soma.

Entrada

A entrada é composta por diversas instâncias e termina com o final de arquivo (EOF). A primeira linha de cada instância contém os inteiros N, indicando o número de pessoas na fila, e K, indicando o número de grupos $(1 \leq K < N \leq 1.000)$. Na linha seguinte são apresentados $N - 1$ inteiros, $a_2, \ldots, a_N$, $(0 \leq a_2 \leq \cdots \leq a_N \leq 1.000.000)$ indicando as posições de cada pessoa em relação à primeira pessoa da fila. A posição da primeira pessoa é omitida, pois é sempre zero.

Saída

Para cada instância, imprima uma única linha contendo o valor mínimo que a soma dos tamanhos dos K grupos pode ter.

Exemplo

Entrada	Saída
5 2 1 2 5 6 4 3 0 1 2	3 0

Problema I: Acampamento de Férias

https://www.urionlinejudge.com.br/judge/pt/problems/view/1167

Nas férias de Julho, várias escolas de uma mesma região resolveram se organizar e levaram uma parte de seus alunos para um acampamento de férias por uma semana. Nestes acampamentos os alunos são divididos em chalés coletivos por gênero e idade, sempre com um supervisor ou supervisora que, além de dormirem com o grupo no chalé, também são responsáveis por criar e executar várias atividades interessantes e animadas, para todas as idades. Dentre as diversas atividades podem-se citar jogos, excursões, Gincana Musical, Gincanas Noturnas, etc. No primeiro dia de acampamento, devido à forte chuva, as atividades recreativas ficaram limitadas e as crianças foram levadas para o ginásio de esportes. Foi realizada uma gincana e uma das atividades da mesma consistiu em agrupar as crianças em um círculo (organizado no sentido anti-horário) do qual seriam retiradas uma a uma até que sobrasse apenas uma criança, que seria a vencedora.

No momento em que entra no círculo, cada criança recebe uma pequena ficha que contém um valor de 1 a 500. Depois que o círculo é formado, conta-se, iniciando na criança que está ao lado da primeira que entrou no círculo, o número correspondente à ficha que a primeira detém. A criança onde o número contado cair, deve ser retirada do grupo, e a contagem inicia novamente segundo a ficha da criança que acabou de ser eliminada. Para ficar mais interessante, quando o valor que consta na ficha é par, a contagem é feita no sentido horário e quando o valor que consta na ficha é ímpar, a contagem é feita no sentido anti-horário.

A brincadeira fez muito sucesso e o administrador do acampamento pediu para que sua equipe desenvolva um programa para que no próximo evento ele saiba previamente qual criança irá ser a vencedora de cada grupo, com base nas informações fornecidas.

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro **N** ($1 \leq \mathbf{N} \leq 100$), indicando a quantidade de crianças que farão parte de cada círculo e participarão da brincadeira. Em seguida, as **N** linhas de cada caso de teste conterão duas informações, o **Nome** e o **Valor** ($1 \leq \mathbf{Valor} \leq 500$) que consta na ficha de cada criança, separados por um espaço, na ordem de entrada na formação do círculo inicial.

Obs: O **Nome** de cada criança não deverá ultrapassar 30 caracteres e contém apenas letras maiúsculas e minúsculas, sem acentos, e o caractere "_". O final da entrada é indicado pelo número zero.

Saída

Para cada caso de teste, deve-se apresentar a mensagem Vencedor(a): xxxxxx, com um espaço após o sinal ":" indicando qual é a criança do grupo que venceu a brincadeira.

Exemplo

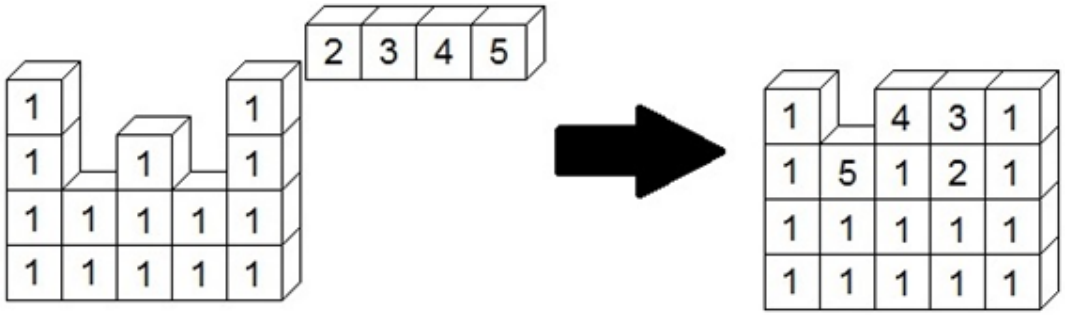
Entrada	Saída
3 Fernanda 7 Fernando 9 Gustavo 11 5 Maria 7 Pedro 9 Joao_Vitor 5 Isabel 12 Laura 8 3 Maria 4 Pedro 3 Joao 2 0	Vencedor(a): Fernanda Vencedor(a): Pedro Vencedor(a): Pedro

Problema J: Empurrando Blocos

https://www.urionlinejudge.com.br/judge/pt/problems/view/1874

A Empresa Blocos Regulares Inventando Serventia de Algo, mais conhecida como BRISA, constrói blocos, sempre do mesmo tamanho. Um detalhe que chama a atenção está na forma em que os blocos são armazenados em estoque, depois de fabricados. Os mesmos são formados por uma fileira de pilhas. A retirada de uma caixa do estoque é um tanto quando desordenado, pois se escolhe uma pilha aleatoriamente e retira-se algum bloco do topo dela.

Porém, a forma de armazenamento é um tanto quanto interessante: uma esteira, localizada na reta do topo da pilha mais à direita do estoque, é utilizada. Com isto, forma-se uma fila com os novos blocos. A esteira roda da direita para a esquerda. Assim que houver um espaço vago em uma das pilhas seguintes, o bloco será inserido na mesma, caso não haja, ele vai avançando até as pilhas seguintes. Segue abaixo um exemplo de inserção de blocos.



Entrada

Haverá diversos casos de teste. Cada caso de teste terá 3 números inteiros, H, P e F, indicando a altura da pilha mais a direita , a quantidade de pilhas de blocos e o tamanho da fila de blocos a ser inserida. Após isto, serão lidos H linhas com P valores, com valores 1, representando onde tem bloco, e 0, representando onde não tem bloco. A seguir, será lida uma linha com F valores, representando a fila com os blocos novos. O último caso de teste é representado por três zeros, e não deverá ser processado.

Saída

Para cada caso de teste, imprima as pilhas após a inserção dos novos blocos. Em alguns casos, a fila de novos blocos será mais que suficiente para que todas as pilhas fiquem do mesmo tamanho. Neste caso, desconsidere os blocos que sobrarem na fila.

Exemplo

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