Questões

TecnoJr – Associação Empresa Júnior de Computação da UESC - CNPJ 058.250.46/0001-63
Campus Soane Nazaré, km 16, S/N, Rodovia Ilhéus - Itabuna, Salobrinho, CEP-45662-000
Fale conosco: tecnojr@uesc.com.br, Visite nosso site: www.tecnojr.com.br
Telefone: (73) 3680 5389

Questões

1) Neste problema, pede-se para armazenar, gerenciar e buscar por indivíduos definidos por um identificador único (inteiro) e pelas seguintes informações: Primeiro e último nome, data de nascimento e telefone.

Entrada

A entrada será feita por 4 comandos: add, del, info e query. A execução é encerrada com EOF.
O comando ''add'' recebe e armazena todos dados do individuo, e retorna erro se já existir individuo com mesmo identificador.

add <id> <first_name> <last_name> <birtday> <phone_number>

O comando ''del'' remove todos dados relacionados a um determinado identificador, e retorna erro se não existir individuo com o identificador fornecido.

del <id>

O comando ''info'' imprime todos dados de um determinado identificador, e retorna erro se não existir individuo com o identificador fornecido.

info <id>

O comando ''query'' realiza uma busca nos indivíduos cadastrados. Conforme as seguintes tags de busca:

- fn: Primeiro nome
- ln: Ultimo nome
- bd: Data de nascimento
- pn: Telefone

query (<tag>:<valor>)+

Saída

O comando ''add'' somente imprime na saída quando ocorre erro na inserção de um individuo, ocorrida na inserção de individuo com identificador duplicado.
O comando ''del'' somente imprime na saída quando o identificador solicitado não existe.
O comando ''info'' imprime todos dados de um determinado identificador, ou imprime erro se não existir individuo com o identificador fornecido.

O comando ''query'' retorna os identificadores que respeitem os critérios da busca na ordem crescente separados por espaços. Em caso de não existir nenhum individuo que respeite a busca, uma linha vazia deve ser impressa.

Exemplos

Entradas:

add 123 Roberto Nascimento 01/01/1960 +55-21-0190-0190
add 123 Joao Souza 11/10/2000 103-99
add 09 Andre Matias 01/01/1970 +55-21-0190-0190
add 222 Diogo Fraga 01/06/1967 +55-21-0190-0190
add 99 Seu Barbosa 01/01/1960 +55-21-0190-0190
add 100 Seu Beirada 01/01/1960 +55-21-9999-9999
add 155 Andre Fortunato 02/01/1962 +55-21-0190-0190
query bd:01/01/1960
query bd:01/01/1960 fn:Seu
query bd:01/01/1960 fn:Seu pn:+55-21-9999-9999
info 100
del 99
query bd:01/01/1960 fn:Seu
info 99
del 99

Saídas:

ID 123 ja cadastrado.
99 100 123
99 100
100
Seu Beirada 01/01/1960 +55-21-9999-9999

ID 99 nao existente.
ID 99 nao existente.

---

2) João é um fanático por miojos; ele os adora, e, como era de se esperar, ele levou vários pacotes quando foi acampar com seus colegas. Como João só gosta de miojos feitos com o tempo exato, ele se deseperou ao perceber que havia esquecido seu relógio em casa.

Por sorte, ele conseguiu, no caminho, comprar duas ampulhetas de durações diferentes. Por exemplo, se o miojo precisa de 3 minutos para ficar pronto, e João tiver uma ampulheta de 5 minutos e outra de 7, uma possível forma de cozinhar o miojo é:

1. João começa virando as duas ampulhetas ao mesmo tempo.
2. Quando a areia da ampulheta de 5 minutos se esgotar, João torna a virá-la.
3. João começa a preparar o miojo quando a areia da ampulheta de 7 minutos acabar.

4. João tira o miojo do fogo quando a ampulheta de 5 minutos acabar novamente.

Dessa forma, o miojo ficará 3 minutos no fogo (do minuto 7 ao minuto 10). Assim, apesar do miojo levar apenas três minutos para ser cozido, ele precisa de 10 minutos para ficar pronto.

Faça um programa que, dado o tempo de preparo do miojo, e os tempos das duas ampulhetas (ambos maiores que o tempo do miojo), determina o tempo mínimo necessário para o miojo ficar pronto. Você pode supor que sempre é possível cozinhar o miojo no tempo correto.

Entrada

A entrada contém um único caso de teste, composto por uma única linha, que contém três inteiros T, A e B, representando o tempo necessário para o preparo do miojo, o tempo da primeira ampulheta e o tempo da segunda ampulheta respectivamente.

Saída

Seu programa deve produzir uma única linha na saída, contendo o tempo mínimo para o preparo do miojo.

Restrições

- $0 \leq T \leq 10000$
- $T < A, B \leq 40000$

Exemplo

Entrada
3 5 7

Saída
10

---

3) Albert, Charles e Mary inventaram uma nova versão do clássico jogo de Bingo. Na versão tradicional, o jogo é presidido por um não-jogador conhecido como *caller*. No começo de cada partida, cada jogador recebe uma carta contendo uma única combinação de números de o até **N** dispostos em colunas e linhas. O *caller* opera um globo contendo **N**+1 bolas numeradas de 0 até **N**. Em cada turno, o *caller* sorteia uma bola do globo, anuncia o número sorteado aos jogadores e *não* a coloca novamente no globo. Cada jogador procura pelo número em sua carta e o marca caso o encontre. O primeiro jogador que marcar um padrão pré-definido completo em sua carta (uma linha horizontal, por exemplo) ganha um prêmio.

Na versão **A**lbert-**C**harles-**M**ary, em cada turno, o *caller* sorteia uma primeira bola, coloca-a de volta no globo, sorteia uma segunda bola, coloca-a de volta no globo, e então anuncia a diferença absoluta entre os números das duas bolas. Para aumentar o entusiasmo, antes do início da partida, um subconjunto possivelmente vazio de bolas é retirado do globo, de forma que ao menos duas bolas permaneçam no globo. Eles gostariam de saber se cada número de 0 até **N** podem ainda ser anunciados utilizando a nova regra de sorteio e considerando apenas as bolas que permaneceram dentro do globo.

Entrada

Cada caso de teste é dado em exatamente duas linhas. A primeira linha contém dois inteiros **N** e **B**. O significado de **N** foi descrito acima ($1 \leq \mathbf{N} \leq 90$), enquanto **B** representa o número de bolas que permaneceram no globo ($2 \leq \mathbf{B} \leq \mathbf{N}+1$). A segunda linha contém **B** inteiros distintos $\mathbf{b_i}$, indicando as bolas que permaneceram no globo ($0 \leq \mathbf{b_i} \leq \mathbf{N}$).

O último caso de teste é seguido por uma linha contendo dois zeros.

Saída

Para cada caso de teste, imprima uma única linha contendo um único caractere 'Y' se for possível anunciar todos os números de 0 até **N**, inclusive, ou um único caractere 'N' caso contrário.

Exemplo

Entrada:
6 7
2 1 3 4 0 6 5
5 4
5 3 0 1
5 3
1 5 0
0 0

Saída:
Y
Y
N

---

4) O conglomerado indiano Tutu é um conjunto de empresas que atua nos mais diversos ramos da indústria, produzindo desde sapatos até aviões e foguetes. Por ser tão diversificada, precisa de grandes e rápidos sistemas para cálculos de contabilidade.

Um dos módulos mais importantes desse sistema é o de fornecimento de produtos, onde fica a base de dados de produtos e fornecedores. Um mesmo produto pode ser fornecido por vários fornecedores diferentes.

A outra grande matriz é a **B**, onde cada linha representa um dia do mês e cada coluna é um produto. O valor da matriz na linha *m* e coluna *n* representa a quantidade do produto *n* a ser adquirido no dia *m*.

Tal empresa tem uma política de fidelidade com seus fornecedores, e uma das práticas efetuadas pela empresa é, em um determinado dia, comprar todos os produtos necessários de um único fornecedor. Isto é, em um dia todos os produtos adquiridos serao comprados do fornecedor *x*, no outro dia do fornecedor *y*, e assim por diante

Para auxiliar a escolha de qual fornecedor será o escolhido no dia, foi gerada outra matriz **C**, que é o resultado da multiplicação das matrizes **A** × **B**. Essa matriz diz o quanto será gasto pela empresa se adquirir todos os produtos de um determinado fornecedor em um determinado dia.

As matrizes **A** e **B** são quadradas (o número de linhas é igual ao número de colunas) e têm valores definidos pelas fórmulas

$\mathbf{A}_{ij} = (P \times i + Q \times j) \pmod X$

$\mathbf{B}_{ij} = (R \times i + S \times j) \pmod Y$

onde i é o índice da linha da matriz e j é o índice da coluna da matriz (todos os índices vão de 1 até N). Os inteiros *P, Q, R, S, X e Y* são parâmetros constantes, que definem as duas matrizes **A** e **B**.

Tarefa

Escreva um programa que, dados os parâmetros das matrizes **A** e **B**, e a posição de uma das entradas as matriz **C**, calcula o valor daquela entrada.

Entrada

A primeira linha da entrada contém um inteiro *N*, indicando as dimensões das matrizes **A**, **B** e **C** ($2 \le N \le 10^5$). A linha seguinte contém seis inteiros *P, Q, R, S, X e Y*, indicando os parâmetros das matrizes **A** e **B** ($2 \le X, Y \le 10^4$; $0 \le P, Q < X$; $0 \le R, S < Y$). Finalmente, a última linha da entrada contém dois inteiros *I* e *J*, indicando a linha e a coluna da matriz **C** a serem consultados ($1 \le I, J \le N$>).

Saída

Seu programa deve imprimir uma única linha contendo o valor da matriz **C** na linha e coluna especificadas.

Exemplo

Entrada
3
4 3 2 3 5 6
2 2

Saída
18

---

5) Um dos jogos de cartas mais divertidos para crianças pequenas, pela simplicidade, é Rouba- Monte. O jogo utiliza um ou mais baralhos normais e tem regras muito simples. Cartas são distingüidas apenas pelo valor (ás, dois, três, . . .), ou seja, os naipes das cartas não são considerados (por exemplo, ás de paus e ás de ouro têm o mesmo valor).

Inicialmente as cartas são embaralhadas e colocadas em uma pilha na mesa de jogo, chamada de pilha de compra, com face voltada para baixo. Durante o jogo, cada jogador mantém um *monte* de cartas, com face voltada para cima; em um dado momento o monte de um jogador pode conter zero ou mais cartas. No início do jogo, todos os montes dos jogadores têm zero cartas. Ao lado da pilha de compras encontra-se uma área denomindada de *área de descarte*, inicialmente vazia, e todas as cartas colocadas na área de descarte são colocadas lado a lado com a face para cima (ou seja, não são empilhadas).

Os jogadores, dispostos em um círculo ao redor da mesa de jogo, jogam em sequência, em sentido horário. As jogadas prosseguem da seguinte forma:

- O jogador que tem a vez de jogar retira a carta de cima da pilha de compras e a mostra aos outros jogadores; vamos chamar essa carta de *carta da vez*.
- Se a carta da vez for igual a alguma carta presente na área de descarte, o jogador retira essa carta da área de descarte colocando-a, juntamente com a carta da vez, no topo de seu monte, com as faces voltada para cima, e continua a jogada (ou seja, retira outra carta da pilha de compras e repete o processo).
- Se a carta da vez for igual à carta de cima de um monte de um outro jogador, o jogador "rouba" esse monte, empilhando-o em seu próprio monte, coloca a carta da vez no topo do seu monte, face para cima, e continua a jogada.
- Se a carta da vez for igual à carta no topo de seu próprio monte, o jogador coloca a carta da vez no topo de seu próprio monte, com a face para cima, e continua a jogada.
- Se a carta da vez for diferente das cartas da área de descarte e das cartas nos topos dos montes, o jogador a coloca na área de descarte, face para cima, e a jogada se encerra (ou seja, o próximo

jogador efetua a sua jogada). Note que esse é o único caso em que o jogador não continua a jogada.

O jogo termina quando não há mais cartas na pilha de compras. O jogador que tiver o maior monte (o monte contendo o maior número de cartas) ganha o jogo. Se houver empate, todos os jogadores com o monte contendo o maior número de cartas ganham o jogo.

Entrada

A entrada é composta de vários casos de teste. A primeira linha de um caso de teste contém dois inteiros N e J, representando respectivamente o número de cartas no baralho ($2 \leq N \leq 10.000$) e o número de jogadores ($2 \leq J \leq 20$ e $J \leq N$). As cartas do baralho são representadas por números inteiros de 1 a 13 e os jogadores são identificados por inteiros de 1 a J. O primeiro jogador a jogar é o de número 1, seguido no jogador de número 2, . . ., seguido pelo jogador de número J, seguido pelo jogador de número 1, seguido do jogador de número 2, e assim por diante enquanto houver cartas na pilha de compras. A segunda linha de um caso de teste contém N inteiros entre 1 e 13, separados por um espaço em branco, representando as cartas na pilha de compras. As cartas são retiradas da pilha de compras na ordem em que aparecem na entrada. O final da entrada é indicado por uma linha com $N = J = 0$.

Saída

Para cada caso de teste seu programa deve imprimir uma linha, contendo o número de cartas do monte do jogador ou jogadores que ganharam a partida, seguido de um espaço em branco, seguido do(s) identificador(es) dos jogadores que ganharam a partida. Se há mais de um jogador vencedor imprima os identificadores dos jogadores em ordem crescente, separados por um espaço em branco.

Exemplo

Entrada:
4 2
10 7 2 5
6 3
1 2 1 2 1 2
8 2
3 3 1 1 2 3 4 5
0 0

Saída:
0 1 2
5 1
3 2

6) Gene e Gina possuem um tipo peculiar de fazenda. Ao invés de criar animais e plantar vegetais como acontece em fazendas normais, eles cultivam *strings*. Uma string é uma sequência de caracteres. As strings, ao crescerem, adicionam caracteres à esquerda e/ou à direita delas mesmas, mas nunca perdem caracteres nem inserem caracteres no meio.

Gene e Gina possuem uma coleção de fotos de algumas strings durante diferentes etapas de seus crescimentos. O problema é que a coleção não é rotulada, portanto eles esqueceram a qual string pertence cada uma das fotos. Eles querem montar um painel para ilustrar os procedimentos do cultivo de strings, mas eles necessitam sua ajuda para encontrar uma sequência de fotos apropriada.

Cada foto ilustra uma string. A sequência de fotos precisa ter a seguinte propriedade: se s_i aparece imediatamente antes de s_{i+1} na sequência, então s_{i+1} é uma string que pode ter crescido a partir de s_i (ou seja, s_i é uma substring contígua de s_{i+1}). Além disso, eles não querem usar fotos repetidas, portanto todas as strings na sequência devem ser diferentes.

Dado um conjunto de strings representando todas as fotos disponíveis, sua tarefa é calcular o tamanho da maior sequência que pode ser produzida com as restrições acima.

Entrada

Cada caso de teste se estende por várias linhas. A primeira linha contém um inteiro **N** representando o número de strings no conjunto ($1 \leq \mathbf{N} \leq 10^4$). Cada uma das próximas **N** linhas contém uma string não-vazia e única com no máximo 1000 caracteres minúsculos do alfabeto inglês. Em cada caso de teste, a soma dos tamanhos das strings é no máximo 10^6.

O último caso de teste é seguido de uma linha contendo um zero.

Saída

Para cada caso de teste, imprima uma única linha com um único inteiro representando o tamanho da maior sequência de fotos que pode ser produzida.

Exemplo

Entrada:
6
plant
ant
cant
decant
deca

an
2
supercalifragilisticexpialidocious
rag
0

Saída:
4
2

---

7) Mário é dono de uma empresa de guarda-volumes, a Armários a Custos Moderados (ACM). Mário conquistou sua clientela graças à rapidez no processo de armazenar os volumes. Para isso, ele tem duas técnicas:

- Todos os armários estão dispostos numa fila e são numerados com inteiros positivos a partir de 1. Isso permite a Mário economizar tempo na hora de procurar um armário;
- Todos os armários têm rodinhas, o que lhe dá grande flexibilidade na hora de rearranjar seus armários (naturalmente, quando Mário troca dois armários de posição, ele também troca suas numerações, para que eles continuem numerados seqüencialmente a partir de 1).

Para alugar armários para um novo cliente, Mário gosta de utilizar armários contíguos, pois no início da locação um novo cliente em geral faz muitas requisições para acessar o conteúdo armazenado, e o fato de os armários estarem contíguos facilita o acesso para o cliente e para Mário.

Desde que Mário tenha armários livres em quantidade suficiente, ele sempre pode conseguir isso. Por exemplo, se a requisição de um novo cliente necessita de quatro armários, mas apenas os armários de número 1, 3, 5, 6, 8 estiverem disponíveis, Mário pode trocar os armários 5 e 2 e os armários 6 e 4 de posição: assim, ele pode alugar o intervalo de armários de 1 até 4.

No entanto, para minimizar o tempo de atendimento a um novo cliente, Mário quer fazer o menor número de trocas possível para armazenar cada volume. No exemplo acima, ele poderia simplesmente trocar os armários 1 e 4 de posição, e alugar o intervalo de 3 até 6.

Mário está muito ocupado com seus clientes e pediu que você fizesse um programa para determinar o número mínimo de trocas necessário para satisfazer o pedido de locação de um novo cliente.

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém dois números inteiros N e L ($1 \leq N \leq L \leq 100.000$), indicando quantos armários são necessários para acomodar o pedido de

locação do novo cliente e quantos armários estão disponíveis, respectivamente. A linha seguinte contém L números inteiros positivos separados por espaços em branco, nenhum deles maior do que 1.000.000.000, indicando as posições dos armários disponíveis. Os números dos armários livres são dados em ordem crescente.

O final da entrada é indicado por um caso onde N = L = 0.

Saída

Para cada caso de teste, imprima uma linha contendo um único número inteiro, indicando o número mínimo de trocas que Mário precisa efetuar para satisfazer o pedido do novo cliente (ou seja, ter N armários consecutivos livres).

Exemplo

Entrada:
5 6
1 3 4 5 6 8
5 5
1 3 5 6 8
3 4
1 4 5 6
0 0

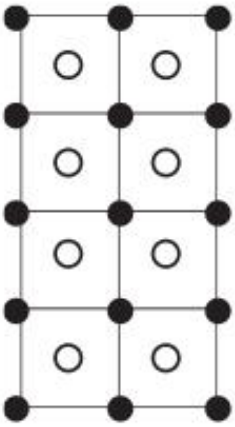
Saída:
1
2
0

---

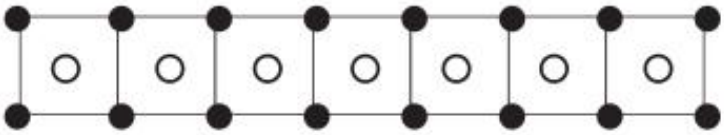
8) O desmatamento é um dos maiores problemas enfrentados pelo Brasil hoje; estima-se que mais de 10 mil km² de vegetação sejam desflorestados todo ano. Além de destruir os habitats de várias espécies em risco de extinção, o desmatamento promove a emissão de gás carbônico, principal responsável pelo efeito estufa e pelo aquecimento global.

A Fundação de Conservação dos Carvalhos (FCC) tenta combater esta tendência, promovendo o reflorestamento das regiões desmatadas. Para isso, eles pretendem plantar carvalhos formando um quadriculado (um carvalho em cada vértice); no centro de cada quadrado formado por eles, a FCC também plantará um eucalipto. Para preservar a biodiversidade da área plantada, pelo menos uma árvore de cada espécie deve ser plantada durante o reflorestamento.

Por exemplo, se a FCC quiser plantar 23 árvores, ela poderá fazê-lo de duas maneiras: ou formando um retângulo *3 × 5* com os carvalhos, como na figura (a), ou formando um retângulo *2 × 8*, como na figura (b).



(a)



(b)

Considere que, para os propósitos deste problema, um retângulo $x \times y$ é equivalente a um retângulo $y \times x$.

Tarefa

Escreva um programa que, dado o número total de árvores que devem ser plantadas, de quantas maneiras diferentes elas podem ser dispostas.

Entrada

A única linha da entrada contém um único inteiro N, que indica o número total de árvores que devem ser plantadas ($1 \leq N \leq 10^9$) .

Saída

Seu programa deve imprimir uma única linha, contendo um único inteiro, indicando o número de arranjos distintos que podem ser feitos para o reflorestamento.

Exemplo

Entrada
7

Saída
0

---

9) Definimos a paridade de um inteiro **n** como a soma dos seus bits em representação binária computada módulo dois. Como exemplo, o número $21 = 10101_2$ possui três *1s* na sua representação binária e portanto ele possui paridade *3 (mod 2)*, ou *1*.

Neste problema, você deverá calcular a paridade de um inteiro **1 ≤ I ≤ 2147483647**.

Entrada

Cada linha da entrada possui um inteiro **I** e o fim da entrada é indicado por uma linha onde **I = 0**, a qual não deve ser processada.

Saída

Para cada inteiro **I** na entrada você deve imprimir uma linha **A paridade de B eh P (mod 2).**, onde **B** é a representação binária de **I**.

Exemplo de Entrada

```
1
2
10
21
0
```

Exemplo de Saída

```
A paridade de 1 eh 1 (mod 2).
A paridade de 10 eh 1 (mod 2).
A paridade de 1010 eh 2 (mod 2).
A paridade de 10101 eh 3 (mod 2).
```

---

10) A Subindo Bem Confortavelmente (SBC) é uma empresa tradicional, com mais de 50 anos de experiência na fabricação de elevadores. Todos os projetos da SBC seguem as mais estritas normas de segurança, mas infelizmente uma série de acidentes com seus elevadores manchou a reputação da empresa.

Ao estudar os acidentes, os engenheiros da companhia concluíram que, em vários casos, o acidente foi causado pelo excesso de passageiros no elevador. Por isso, a SBC decidiu fiscalizar com mais rigor o uso de

seus elevadores: foi instalado um sensor em cada porta que detecta a quantidade de pessoas que saem e entram em cada andar do elevador.

A SBC tem os registros do sensor de todo um dia de funcionamento do elevador (que sempre começa vazio). Eles sabem que as pessoas são educadas e sempre deixam todos os passageiros que irão sair em um andar saírem antes de outros passageiros entrarem no elevador, mas ainda assim eles têm tido dificuldade em decidir se a capacidade máxima do elevador foi excedida ou não.

Tarefa

Escreva um programa que, dada uma sequência de leituras do sensor e a capacidade máxima do elevador, determina se a capacidade máxima do elevador foi excedida em algum momento.

Entrada

A primeira linha da entrada contém dois inteiros N e C, indicando o número de leituras realizadas pelo sensor e a capacidade máxima do elevador, respectivamente ($1 \leq N \leq 1000$ e $1 \leq C \leq 1000$). As N linhas seguintes contêm, cada uma, uma leitura do sensor. Cada uma dessas linhas contém dois inteiros S e E, indicando quantas pessoas saíram e quantas pessoas entraram naquele andar, respectivamente ($0 \leq S \leq 1000$ e $0 \leq E \leq 1000$)

Saída

Seu programa deve imprimir uma única linha contendo o caractere 'S', caso a capacidade do elevador tenha sido excedida em algum momento, ou o caractere 'N' caso contrário.

Exemplo

Entrada
5 10
0 5
2 7
3 3
5 2
7 0

Saída
N

Entrada
5 10
0 3
0 5
0 2
3 4

6 4

Saída
S

Entrada
6 4
0 5
3 5
4 5
1 0
1 1
1 1

Saída
S

A TecnoJr lhes deseja boa sorte!