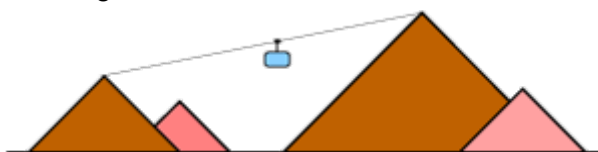


Bondinho

A turma do colégio vai fazer uma excursão na serra e todos os alunos e monitores vão tomar um bondinho para subir até o pico de uma montanha. A cabine do bondinho pode levar 50 pessoas no máximo, contando alunos e monitores, durante uma viagem até o pico. Neste problema, dado como entrada o número de alunos A e o número de monitores M , você deve escrever um programa que diga se é possível ou não levar todos os alunos e monitores em apenas uma viagem!



Entrada

A primeira linha da entrada contém um inteiro A , representando a quantidade de alunos. A segunda linha da entrada contém um inteiro M , representando o número de monitores.

Saída

Seu programa deve imprimir uma linha contendo o caractere S se é possível levar todos os alunos e monitores em apenas uma viagem, ou o caractere N caso não seja possível.

Restrições

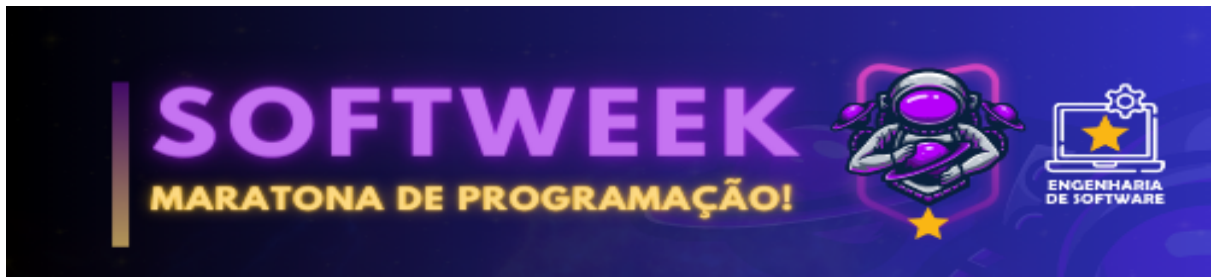
- $1 \leq A \leq 50$
- $1 \leq M \leq 50$

Exemplos

Entrada 10 20	Saída S
---------------------	------------

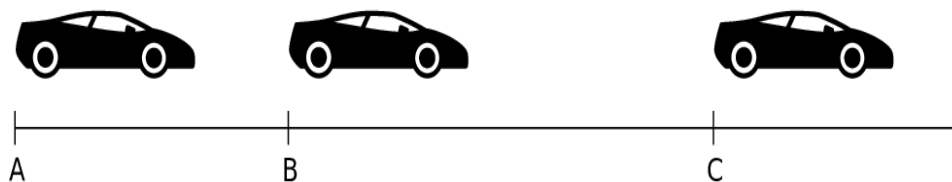
Entrada 12 39	Saída N
---------------------	------------

Entrada 49 1	Saída S
--------------------	------------



Piloto Automático

Uma grande fábrica de carros elétricos está realizando melhorias no sistema de piloto automático e precisa da sua ajuda para implementar um programa que decida se um carro B, que está trafegando no meio de dois carros A e C, precisa acelerar, desacelerar ou manter a velocidade atual. Os carros são iguais e os sensores do piloto automático vão fornecer, como entrada, a posição atual da traseira dos três carros. Veja um exemplo na figura.



O carro B precisa ser acelerado se a distância da sua traseira para a traseira do carro A for menor do que a distância da sua traseira para a traseira do carro C. Se for maior, ele precisa ser desacelerado. Se for igual, precisa manter a velocidade atual. Quer dizer, o carro B precisa ser acelerado se $(B-A) < (C-B)$, desacelerado se $(B-A) > (C-B)$ e manter a velocidade se $(B-A)$ for igual a $(C-B)$.

Entrada

A primeira linha da entrada contém um inteiro A. A segunda linha da entrada contém um inteiro B. A terceira linha da entrada contém um inteiro C. Os três inteiros representam as posições atuais das traseiras dos carros A, B e C, respectivamente.

Saída

Seu programa deve imprimir uma linha contendo um inteiro: 1 se o carro B precisa acelerar; -1 se precisa desacelerar; ou 0 se precisa manter a velocidade atual.

Restrições

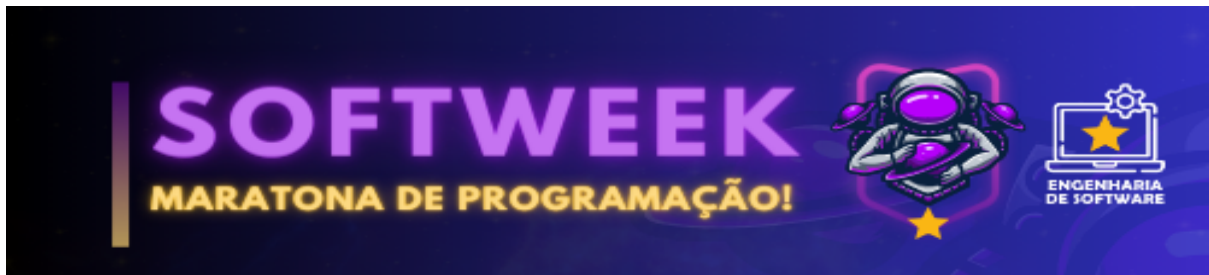
- $0 \leq A < B < C \leq 500$

Exemplos

Entrada 10 23 38	Saída 1
---------------------------	------------

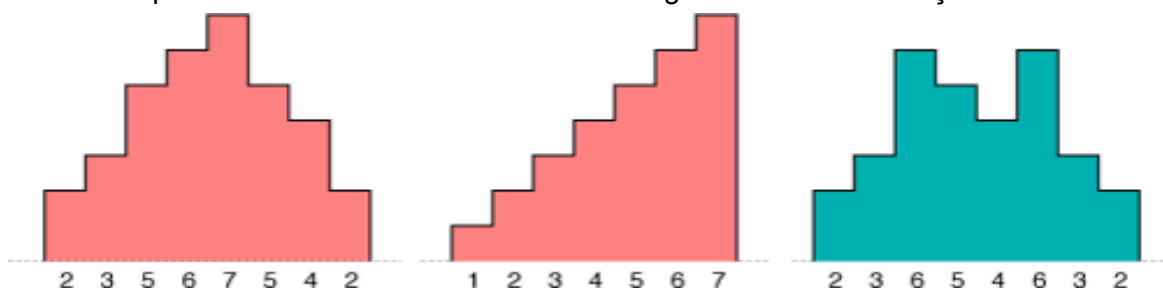
Entrada 105 212 319	Saída 0
------------------------------	------------

Entrada 80 120 132	Saída -1
-----------------------------	-------------



Montanha

Um sistema de informações geográficas computadorizado está representando o perfil de uma montanha através de uma sequência de números inteiros, na qual não há dois números consecutivos iguais, como ilustrado na figura abaixo para três montanhas. Os números representam a altura da montanha ao longo de uma certa direção.



O gerente do sistema de informações geográficas pesquisou e encontrou uma maneira de identificar se uma sequência de números inteiros representa uma montanha com mais de um pico, ou com apenas um pico. Ele observou que, como não há números consecutivos iguais, se houver três números consecutivos na sequência, tal que o número do meio é menor do que os outros dois números, então a montanha tem mais de um pico. Caso contrário, a montanha tem apenas um pico. De forma mais rigorosa, se a sequência é $A=[A_1, A_2, A_3, \dots, A_{N-2}, A_{N-1}, A_N]$, ele quer saber se há uma posição i , para $2 \leq i \leq N-1$, tal que $A_{i-1} > A_i$ e $A_i < A_{i+1}$.

Para ajudar o gerente, seu programa deve determinar, dada a sequência de números inteiros representando a montanha, se ela tem mais de um pico, ou se tem um pico apenas.

Entrada

A primeira linha da entrada contém um inteiro N , representando o tamanho da sequência. A segunda linha contém N inteiros A_i , $1 \leq i \leq N$, representando a sequência de alturas da montanha.

Saída

Seu programa deve imprimir uma linha contendo o caractere "S" se há mais de um pico, ou o caractere "N" se há apenas um pico.

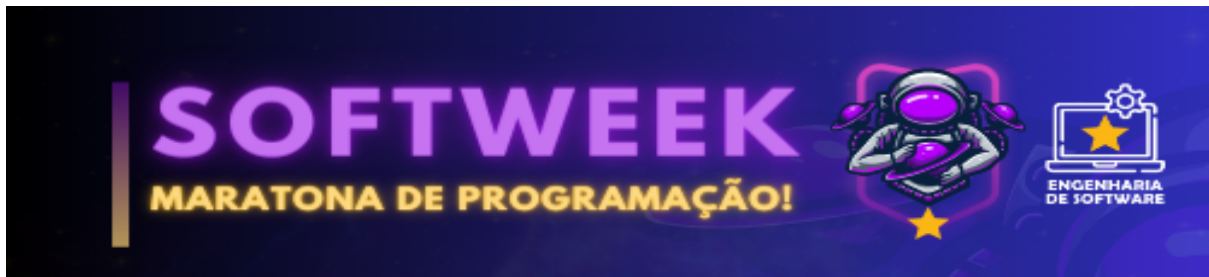
Restrições

- $3 \leq N \leq 1000$
- $1 \leq A_i \leq 1000$, para $1 \leq i \leq N$

Exemplos

Entrada 8 2 3 5 6 7 5 4 2	Saída N
---------------------------------	------------

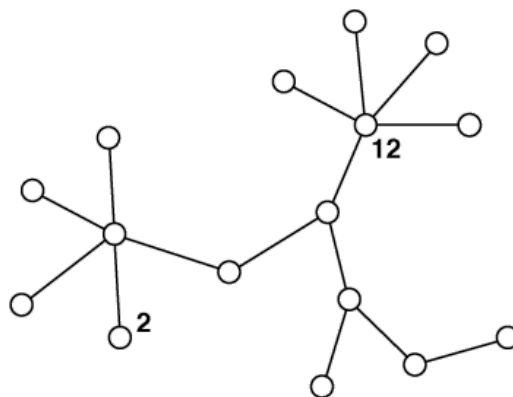
Entrada 8 2 3 6 5 4 6 3 2	Saída S
---------------------------------	------------



Ônibus

A Linearlândia é composta de N cidades, numeradas de 1 até N . Para alguns pares de cidades existe uma linha de ônibus que faz o trajeto de ida e volta diretamente entre as duas cidades do par. Os pares de cidades ligados diretamente por uma linha de ônibus são escolhidos de forma que sempre é possível ir de qualquer cidade para qualquer outra cidade por um, e somente um, caminho (um caminho é uma sequência de linhas de ônibus, sem repetição).

Dada a lista de pares de cidades ligados diretamente por linhas de ônibus, uma cidade origem e uma cidade destino, seu programa deve computar quantos ônibus é preciso pegar para ir da origem ao destino. Por exemplo, na figura, para ir da cidade 2 para a cidade 12 é preciso pegar 4 ônibus.



Entrada

A primeira linha da entrada contém três inteiros N , A e B , representando o número de cidades na Linearlândia, a cidade origem e a cidade destino, respectivamente. As $N-1$ linhas seguintes contém, cada uma, dois inteiros P e Q , indicando que existe uma linha de ônibus ligando diretamente as cidades P e Q .

Saída

Seu programa deve imprimir uma linha contendo um inteiro representando quantos ônibus é preciso pegar para ir de A até B .

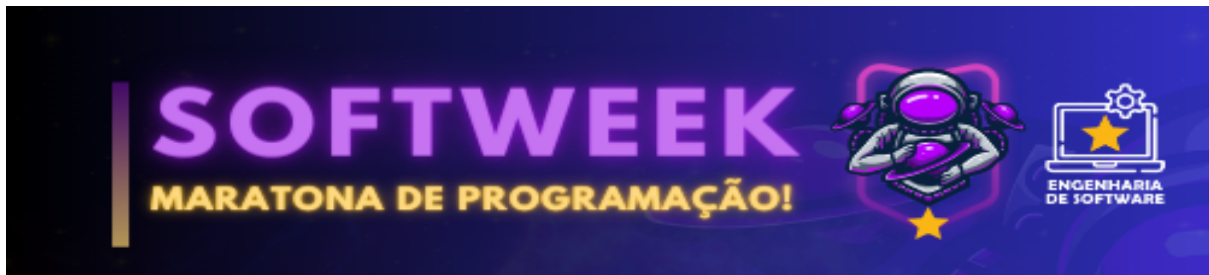
Restrições

- $2 \leq N \leq 10000$
- $1 \leq A \leq N, 1 \leq B \leq N, A \neq B$
- $1 \leq P \leq N, 1 \leq Q \leq N$

Exemplos

Entrada 4 2 4 1 2 2 3 3 4	Saída 2
---------------------------------------	------------

Entrada 16 2 12 3 5 12 3 5 1 2 1 4 1 6 1 7 1 12 8 12 9 12 10 12 11 3 13 13 14 15 13 15 16	Saída 4
---	------------



Falha de segurança

Rafael foi contratado como programador por um grande banco que está atualizando todo o sistema computacional. O novo sistema vai ser instalado amanhã, mas Rafael acabou de descobrir uma falha grave na nova autenticação para acesso às contas do banco: se um usuário digitar como senha uma cadeia de caracteres que contenha, como sub-cadeia contígua, a senha correta para esse usuário, o sistema se confunde e permite o acesso.

Por exemplo, se a senha correta é 'senhafraca' e o usuário digitar 'quesenhafracameu' ou 'senhafraca123', o sistema permite o acesso. Note que nesse caso o sistema não permite o acesso se o usuário digitar 'senha' ou 'nhafra' ou 'senha123fraca'. O chefe de Rafael chamou um programador mais experiente para alterar a autenticação do novo sistema, mas solicitou que Rafael determinasse, para o conjunto de senhas existentes, quantos pares ordenados (A,B) de usuários distintos existem tal que o usuário A, usando sua senha, consegue acesso à conta do usuário B. Você poderia por favor ajudar Rafael?

Entrada

A primeira linha da entrada contém um número inteiro N , o número de usuários no sistema. Cada uma das N linhas seguintes contém uma senha S_i , a senha do i -ésimo usuário.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o número de pares ordenados (A,B) de usuários distintos tal que o usuário A, usando sua senha, consegue acesso à conta do usuário B.

Restrições

- $1 \leq N \leq 20\,000$
- S_i inicia com letra minúscula sem acento e contém apenas letras minúsculas sem acento e dígitos de 0 a 9, para $1 \leq i \leq N$
- $1 \leq \text{comprimento de } S_i \leq 10$

Informações sobre a pontuação

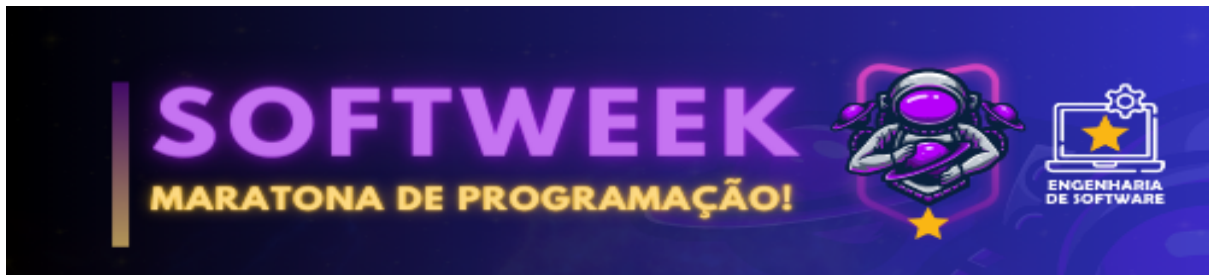
- Para um conjunto de casos de testes valendo 12 pontos, comprimento de $S_i=1$ e $N \leq 1000$.
- Para um conjunto de casos de testes valendo outros 28 pontos, $N \leq 2000$.
- Para um conjunto de casos de testes valendo outros 60 pontos, nenhuma restrição adicional.

Exemplos

Entrada 3 xxx x23 xx	Saída 1
----------------------------------	------------

Entrada 3 a a a8	Saída 4
------------------------------	------------

Entrada 5 jus justa ta us t	Saída 6
---	------------



Fibonacci, How Many Calls?

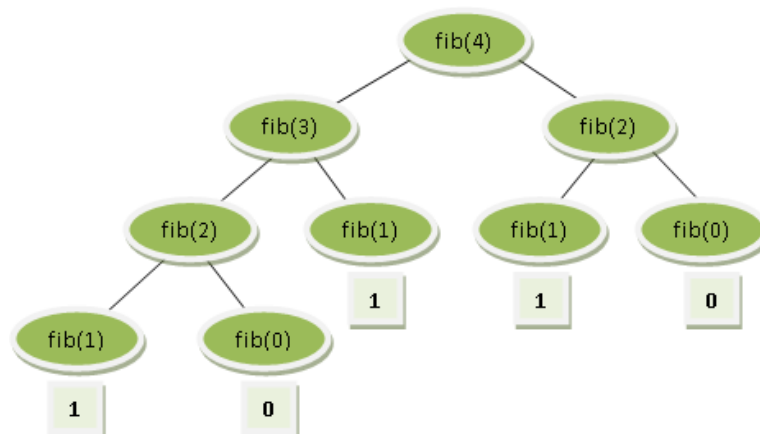
Às vezes, quando você é um estudante de Ciência da Computação, verá um exercício ou um problema envolvendo a sequência de Fibonacci. Esta sequência tem os dois primeiros valores 0 (zero) e 1 (um) e cada próximo valor será sempre a soma dos dois números anteriores. Por definição, a fórmula para encontrar qualquer número de Fibonacci é:

$$\text{fib}(0) = 0$$

$$\text{fib}(1) = 1$$

$$\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2);$$

Uma maneira de encontrar os números de Fibonacci é por meio de chamadas recursivas. Isso é ilustrado abaixo, apresentando a árvore de derivação quando calculamos $\text{fib}(4)$, ou seja, o quinto valor desta sequência:



Dessa maneira,

- $\text{fib}(4) = 1+0+1+1+0 = 3$
- 8 chamadas recursivas devem ser feitas.

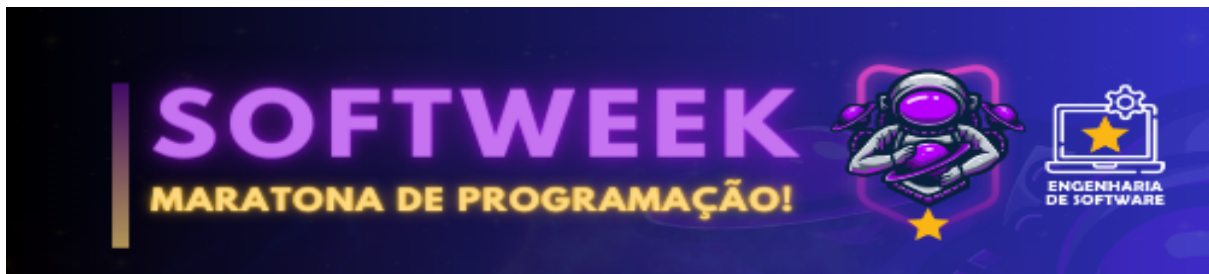
Entrada

A primeira linha de entrada contém um único número inteiro N , indicando o número de casos de teste. Cada caso de teste contém um número inteiro X ($1 \leq X \leq 39$).

Saída

Para cada caso de teste, teremos uma linha de saída, no seguinte formato: `fib(n) = num_calls chamadas = resultado`, onde `num_calls` é o número de chamadas recursivas, sempre com um espaço antes e depois do sinal de igual, conforme mostrado abaixo.

Entradas	Saídas
2	fib(5) = 14 chamadas = 5
5	fib(4) = 8 chamadas = 3
4	



How Many Calls?

O número de Fibonacci é definido pela seguinte recorrência:

- $\text{fib}(0) = 0$
- $\text{fib}(1) = 1$
- $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$

Mas não estamos interessados nos números de Fibonacci aqui. Gostaríamos de saber quantas chamadas são necessárias para avaliar o número de Fibonacci n se seguirmos a recorrência dada. Como os números serão bastante grandes, gostaríamos de facilitar o trabalho para você. Precisariamos apenas do último dígito do número de chamadas, quando esse número é representado na base b .

Entrada

A entrada consiste em vários casos de teste. Para cada teste, haverá dois números inteiros n ($0 \leq n < (263 - 1)$) e b ($1 < b \leq 10000$). A entrada é terminada por um caso de teste onde $n=0$ e $b=0$, você não deve processar este caso de teste.

Saída

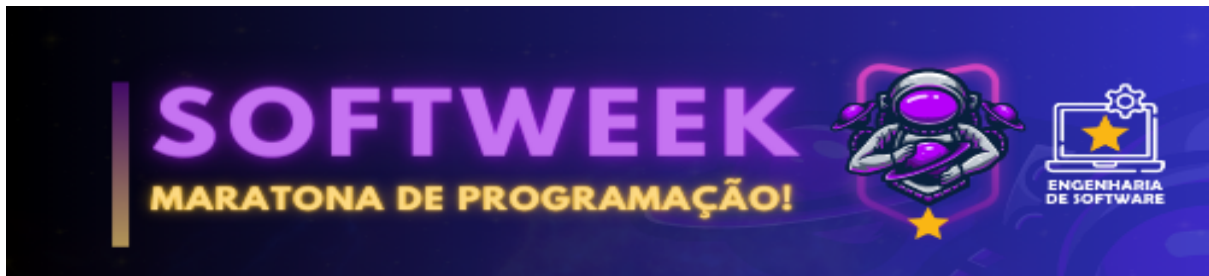
Para cada caso de teste, imprima o número do caso de teste primeiro. Em seguida, imprima n , b e o último dígito (na base b) do número de chamadas. Deve haver um único espaço entre os dois números de uma linha. Observe que o último dígito deve ser representado no sistema de números decimais.

Entradas

0 100
1 100
2 100
3 100
10 10
3467 9350
0 0

Saídas

Caso 1: 0 100 1
Caso 2: 1 100 1
Caso 3: 2 100 3
Caso 4: 3 100 5
Caso 5: 10 10 7
Caso 6: 3467 9350 7631



Eliminating Ballons

Um enorme número de balões estão flutuando no Salão de Convenções após a Cerimônia de Premiação do Concurso Subregional da ICPC. O gerente do Salão de Convenções está irritado, porque ele vai sediar outro evento amanhã e os balões devem ser removidos. Felizmente este ano Carlinhos veio preparado com seu arco e flechas para estourar os balões. Além disso, felizmente, devido ao fluxo de ar condicionado, os balões estão todos no mesmo plano vertical (ou seja, um plano paralelo a uma parede), embora em alturas e posições distintas. Carlinhos vai atirar do lado esquerdo do salão de convenções, a uma altura escolhida, na direção do lado direito do Salão de Convenções. Cada flecha se move da esquerda para a direita, na altura em que foi atirada, no mesmo plano vertical dos balões. Quando uma flecha toca um balão, o balão estoura e a flecha continua seu movimento para a direita, a uma altura diminuída por 1. Em outras palavras, se a flecha estava na altura H , após estourar um balão ela continua na altura $H-1$. Carlinhos quer estourar todos os balões atirando o mínimo de flechas possível. Você pode ajudá-lo?

Entrada

A primeira linha de entrada contém um número inteiro N , o número de balões ($1 \leq N \leq 5 \cdot 10^5$). Como todos os balões estão no mesmo plano vertical, vamos definir que a altura de um balão é dada em relação ao eixo y e a posição de um balão é dada em relação ao eixo x . Os balões são numerados de 1 a N . Os números dos balões indicam suas posições, do mais à esquerda (balão número 1) ao mais à direita (balão número N), independentemente de suas alturas. A posição do balão número i é diferente da posição do balão número $i+1$, para todos i . A segunda linha contém N inteiros H_i , onde H_i indica a altura do balão número i ($1 \leq H_i \leq 10^6$ para $1 \leq i \leq N$).

Saída

Seu programa deve gerar uma única linha, contendo um único número inteiro, o número mínimo de flechas que Carlinhos precisa atirar para estourar todos os balões.

Entradas

Saídas

5

2

3 2 1 5 4

4

4

1 2 3 4

6

3

5 3 2 4 6 1