**Problema A: Mamão Com Açúcar**

*TL: 2 segundos (1 caso)*

*ML: 64 MB*

Em cada linha da entrada você receberá uma string **que pode conter espaços**. Você deve imprimí-las na tela até que a linha **mamao com acucar** seja encontrada. O tamanho máximo será de 1000 caracteres.

**Entrada**

Cada linha terá uma string que contém letras do alfabeto e espaços.

**Saída**

Imprima todas as strings até que a string dita anteriormente seja encontrada.

|  |  |
| --- | --- |
| **Entrada de Teste**  a muleke  princesa carlota imperadora dos andes  cada linha pode ter varias palavras  mamao com leite  mamao com banana  mamao com acucar  mamao com ovo | **Saída de Teste**  a muleke  princesa carlota imperadora dos andes  cada linha pode ter varias palavras  mamao com leite  mamao com banana |

**Problema B: Gostei Desse Problema**

*TL: 2 segundos (1 caso)*

*ML: 64 MB*

É dada uma lista de palavras no dicionário. Dadas algumas strings, imprima se elas existem ou não no dicionário.

**Entrada**

O problema tem um único caso de teste. Na primeira linha temos um número **N** (1 <= **N** <= 100000), que indica quantas palavras temos no dicionário. Nas **N** linhas seguintes teremos as palavras listadas uma a uma, de tamanho máximo 100. Depois teremos um número **Q** (1 <= **Q** <= 100000), que indica quantas strings devem ser procuradas no dicionário. Nas **Q** linhas seguintes teremos as palavras a serem procuradas, de tamanho máximo 100.

Todas as palavras contém apenas letras do alfabeto.

**Saída**

Para cada palavra a ser procurada imprima **S** se ela existe ou **N** caso contrário.

|  |  |
| --- | --- |
| **Entrada de Teste**  11  usem  hahaha  map  de  strings  asa  senao  BoLa  vai  dar  TLE  6  asa  bola  BoLa  DE  dE  vaia | **Saída de Teste**  S  N  S  N  N  N |

**Problema C: Duplicados De Novo!**

*TL: 2 segundos (~50 casos)*

*ML: 64 MB*

Uma famosa empresa fabricante de números distintos está à beira da falência! Isso tudo porque no meio de sua grande linha de produção alguns números repetidos acabaram vazando por engano! Você foi contratado para descobrir se existe algum numero repetido na sequência dada.

**Entrada**

A entrada tem vários casos de teste. Para cada caso haverá uma linha contendo um inteiro **N** ( 1 <= N <= 10^4 ) indicando quantos números foram produzidos, a próxima linha conterá **N** inteiros (de 0 até 2^31-1) separados por espaços em branco que representam os números (veja o exemplo da entrada para maior esclarecimento).

**Saída**

Para cada caso de teste, imprima **S** se existe algum número repetido na entrada, caso contrário imprima **N**.

|  |  |
| --- | --- |
| **Entrada**  3  3 1 2  3  1 1 2 | N  S |

**Problema D: A+B = C ?**

*TL: 2 segundos (~50 casos)*

*ML: 64 MB*

Esse problema é tão fácil que dispensa história: Dada uma sequência de **N** inteiros, existe algum par A+B dessa sequência que é igual a C?

**Entrada**

A entrada é composta de vários casos de teste. A primeira linha conterá um número **N**, o tamanho da sequência. A próxima linha conterá **N** inteiros, que são os números. Por fim, a última linha de cada caso, conterá um inteiro **C**. Os números vão de 0 até 10^9.

**Saída**

Imprima para cada caso de teste **S** se existe algum par A+B que seja igual a C, caso contrario imprima **N**.

**Constantes**

1 <= N <= 10^4

|  |  |
| --- | --- |
| **Entrada de Teste**  3  1 2 4  4  2  2 3  5 | **Saída de Teste**  N  S |

Nota: O primeiro caso de teste a resposta é não porque não existem 2 números que somados dão 4, para o segundo caso, 2 + 3 = 5, logo a resposta é sim.

**Problema E: Ordonhando**

*TL: 2 segundos (~50 casos)*

*ML: 64 MB*

Você receberá uma lista de números. Seu objetivo é simples, contar quantos números únicos esta lista tem! Atenção, o tempo limite deste problema é bem apertado, tome cuidado com o que você irá programar!!

**Entrada**

Há vários casos de teste. Cada caso começa com um número **N** (1 <= **N** <= 10^4), indicando quantos números a lista tem. Na próxima linha teremos os números separados por espaço.

**Saída**

Para cada caso imprima quantos números distintos há na lista.

|  |  |
| --- | --- |
| **Entrada de Teste**  3  3 2 1  5  1 1 1 1 1  4  1 2 1 2 | **Saída de Teste**  3  1  2 |

**Problema F: Votação**

*TL: 2 segundos (~50 casos)*

*ML: 64 MB*

Em um sistema de votação de um país, são feitas quatro votações. Um candidato tem quatro números que identificam sua posição na primeira votação, na segunda votação, na terceira e quarta.

Um candidato A é superior à um candidato B se e somente se ele ficou na frente em mais de duas votações. Analogamente, se A é inferior à um candidato em mais de duas ele é dito inferior.

Dado as posições dos candidatos, sua tarefa é dizer se existe um candidato que é superior à todos os outros, ou seja, se existe um candidato vencedor.

**Entrada**

A entrada tem vários casos de teste. Para cada caso haverá uma linha contendo um inteiro **N** ( 1 <= N <= 10^4 ) indicando quantos candidatos existem. Nas próximas N linhas, temos as informações sobre os candidatos: cada linha terá quatro números representando a posição do candidato na primeira, na segunda votação, na terceira e na quarta. A posição na votação é um número de 1 a **N** e é única na votação correspondente.

**Saída**

Para cada número da entrada imprima **S** se existe tal candidato ou **N** caso contrário.

|  |  |
| --- | --- |
| **Entrada de Teste**  2  1 1 2 1  2 2 1 2  2  1 1 2 2  2 2 1 1  5  2 4 2 2  1 2 3 3  3 5 1 5  4 1 5 1  5 3 4 4 | **Saída de Teste**  S  N  S |

**Problema G: Giullian O Imperador**

*TL: 2 segundos (~50 casos)*

*ML: 64 MB*

Giullian o Imperador irá comprar ouro. O ouro é vendido em forma de barras. Cada barra tem um valor. O modo como é computado o preço das barras a serem compradas é bem inusitado. Para melhor entender como se é calculado o preço, considere o seguinte exemplo:

Suponha que temos as barras na seguinte ordem: 10, 5, 3 e 5. O preço desta configuração de barras é calculado somando-se a subtração de elementos consecutivos: (10-5=5) + (5-3=2) + (3-5=-2) = 5+2-2 = 5. Suponha outro conjunto de barras: 1, 2 e 3. O preço desta vez vale (1-2=-1) + (2-3=-1) = -1 + -1 = -2. Mesmo sendo negativo este preço é válido!

Como o imperador não é bobo, ele percebeu que reordenando as barras, podemos alterar o preço a se pagar. Seu objetivo aqui é ajudar o imperador a reordenar as barras de modo que o preço final seja o menor possível.

**Entrada**

A entrada tem vários casos de teste. Para cada caso haverá uma linha contendo um inteiro **N** ( 1 <= N <= 10^5 ) indicando quantas barras temos. Nas próximas linhas, temos os preços das barras, separados por espaços. O valor das barras será menor do que 10^9 e maior do que 0.

**Saída**

Para cada caso imprima o menor valor possível que podemos atingir.

|  |  |
| --- | --- |
| **Entrada de Teste**  3  1 2 3  3  3 2 1  2  2 0  4  10 5 3 5 | **Saída de Teste**  -2  -2  -2  -7 |