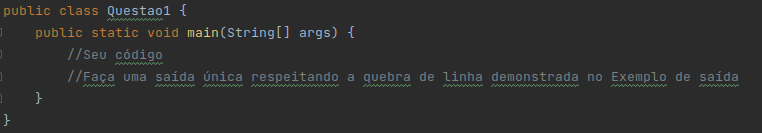
**Nome completo: LUCAS AVANZI**

**Instruções gerais**

Você deverá resolver os problemas abaixo. O código da sua solução será avaliado.

Cada questão deverá ter seu próprio método **main** e deverá seguir o seguinte padrão:



Ao enviar as respostas, exporte a pasta **src** em um arquivo **zip.**

A linguagem de programação usada deverá ser Java.

Boa sorte!

1. Escreva um programa que imprima os números de 1 a 100. Mas para múltiplos de três imprima A em vez do número e para os múltiplos de cinco imprima Z.

Para números que são múltiplos de três e cinco imprimir AZ.

CAMPO TEXTO

1. Muitos sites na internet adicionam uma sessão chamada “Perguntas mais Frequentes” que, como o nome já diz, contém as perguntas mais feitas pelos usuários que utilizam o site.

O portal da AZ costuma receber muitas perguntas de seus usuários, então Neilor imaginou que seria uma boa ideia adicionar uma sessão de Perguntas mais Frequentes no site. Como o Neilor anda muito ocupado ultimamente, ele pediu a sua ajuda para adicionar essa sessão.

Dados os identificadores de perguntas feitas pelos usuários, diga o número de perguntas que serão adicionadas na nova sessão do site. Uma pergunta é classificada como “frequente” quando ela é feita ao menos K vezes.

**Entrada**

Haverá diversos casos de teste. Cada caso de teste inicia com dois inteiros **N** e **K** (1 ≤ **N** ≤ 1000, 1 ≤ **K** ≤ 100), indicando o número de perguntas realizadas, e o número de vezes que uma pergunta deve ser feita para ser considerada “frequente”, respectivamente.

Em seguida haverá **N** inteiros **P** (1 ≤ P ≤ 100), cada um indicando o número de uma determinada pergunta.

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 um inteiro, indicando o número de perguntas que serão adicionadas na nova sessão do site.

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 5 2  1 4 2 1 3  5 2  1 1 2 3 2  8 3  1 1 3 5 4 6 3 4  8 3  1 1 1 1 1 1 1 1  0 0 | 1  2  0  1 |

1. O recente terremoto em Nlogônia não chegou a afetar muito as edificações da capital, principal epicentro do abalo. Mas os cientistas detectaram que o principal dique de contenção teve um dano significativo na sua parte subterrânea que, se não for consertado rapidamente, pode causar o seu desmoronamento, com a consequente inundação de toda a capital.

O conserto deve ser feito por mergulhadores, a uma grande profundidade, em condições extremamente difíceis e perigosas. Mas como é a sobrevivência da própria cidade que está em jogo, seus moradores acudiram em grande número como voluntários para essa perigosa missão.

Como é tradicional em missões perigosas, cada mergulhador recebeu no início do mergulho uma pequena placa com um número de identificação. Ao terminar o mergulho, os voluntários devolviam a placa de identificação, colocando-a em um repositório.

O dique voltou a ser seguro, mas aparentemente alguns voluntários não voltaram do mergulho. Você foi contratado para a penosa tarefa de, dadas as placas colocadas no repositório, determinar quais voluntários perderam a vida salvando a cidade.

**Entrada**

A entrada contém vários casos de teste e termina com EOF. Cada caso de teste é composto de duas linhas. A primeira linha contém dois inteiros **N** e **R** ( 1 ≤ **R** ≤ **N** ≤ 10^4), indicando respectivamente o número de voluntários que mergulhou e o número de voluntários que retornaram do mergulho. Os voluntários são identificados por números de 1 a N. A segunda linha da entrada contém R inteiros, indicando os voluntários que retornaram do mergulho (ao menos um voluntário retorna do mergulho).

**Saída**

Seu programa deve produzir uma única linha para cada caso de teste, contendo os identificadores dos voluntários que não retornaram do mergulho, na ordem crescente de suas identificações. Deixe um espaço em branco após cada identificador (note que isto significa que deve haver um espaço em branco também após o último identificador). Se todos os voluntários retornaram do mergulho, imprima apenas o caractere ‘\*(asterisco).

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 5 3  3 1 5  6 6  6 1 3 2 5 4 | 2 4  \* |

1. João está trabalhando em uma mina, tentando retirar o máximo que consegue de diamantes "<>". Ele deve excluir todas as partículas de areia "." do processo e a cada retirada de diamante, novos diamantes poderão se formar. Se ele tem como uma entrada .<...<<..>>....>....>>>., três diamantes são formados. O primeiro é retirado de <..>, resultando .<...<>....>....>>>. Em seguida o segundo diamante é retirado, restando .<.......>....>>>. O terceiro diamante é então retirado, restando no final .....>>>., sem possibilidade de extração de novo diamante.

## **Entrada**

Deve ser lido um valor inteiro **N** que representa a quantidade de casos de teste. Cada linha a seguir é um caso de teste que contém até 1000 caracteres, incluindo "**<**,**>**, ."

## **Saída**

Você deve imprimir a quantidade de diamantes possíveis de serem extraídos em cada caso de entrada.

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 2  <..><.<..>>  <<<..<......<<<<....> | 3  1 |

1. A divisão de Suprimentos de Botas e Calçados do Exército comprou um grande número de pares de botas de vários tamanhos para seus soldados. No entanto, por uma falha de empacotamento da fábrica contratada, nem todas as caixas entregues continham um par de botas correto, com duas botas do mesmo tamanho, uma para cada pé. O sargento mandou que os recrutas retirassem todas as botas de todas as caixas para reembalá-las, desta vez corretamente.

Quando o sargento descobriu que você sabia programar, ele solicitou com a gentileza habitual que você escrevesse um programa que, dada a lista contendo a descrição de cada bota entregue, determina quantos pares corretos de botas poderão ser formados no total.

## **Entrada**

A entrada é composta por diversos casos de teste e termina com final de arquivo (EOF). A primeira linha de um caso de teste contém um inteiro **N** (2 ≤ **N** ≤ 10^4), **N** é par, indicando o número de botas individuais entregues. Cada uma das N linhas seguintes descreve uma bota, contendo um número inteiro **M** (30 ≤ **M** ≤ 60) e uma letra **L**, separados por uma espaço em branco. **M** indica o número da bota e **L** indica o pé da bota: **L** = 'D' indica que a bota é para o pé direito, **L** = 'E' indica que a bota é para o pé esquerdo.

## **Saída**

Para cada caso de teste imprima uma linha contendo um único número inteiro indicando o número total de pares corretos que podem ser formados.

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 4  40 D  41 E  41 D  40 E  6  38 E  38 E  40 D  38 D  40 D  37 E | 2  1 |