**Desafio1 – Diamantes e Areias**Beecrow - Problema ID: 1069  
[**https://judge.beecrowd.com/en/problems/view/1069**](https://judge.beecrowd.com/en/problems/view/1069)

John trabalha em uma mina de diamantes e tem como objetivo extrair a maior quantidade possível de diamantes “**<>**”. Neste processo, partículas de areia também podem ser encontradas entre os diamantes, sendo representadas por “**.**”. Para formar um diamante, as partículas de areia podem ser desconsideradas. Por exemplo, se John encontrou uma amostra representada por **<...<<..>>....>....>>>**., três diamantes podem ser formados:

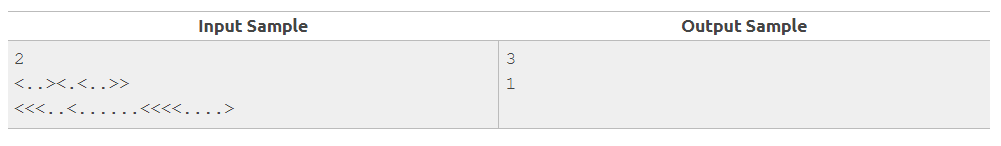
* Os três diamantes que podem ser formados neste exemplo estão destacados em cores diferentes: **<...<<..>>....>....>>>**.
* O primeiro diamante formado é **<..>**, resultando em **<... <> ....> .... >>>.**
* O segundo diamante formado é **<>**, resultando em **<.......> .... >>>.**
* O terceiro diamante formado é **<.......>,** resultando em **.... >>>.**

**Entrada**

O algoritmo deve ler um número inteiro **N**, este número representará a quantidade de amostras que serão consideradas. Cada amostra consiste em um conjunto de caracteres, variando entre **<**, **>** e **.**

**Saída**

O algoritmo deve imprimir o número total de diamantes que podem ser extraídos em cada amostra.



Lógica de Resolução utilizando **Pilha**:

Vamos percorrer todos os caracteres da amostra utilizando um laço de repetição. Para cada caractere da amostra:

* 1. Se **<**, adicionamos o caractere na Pilha;
  2. Se **.**, ignoramos o caractere e vamos para a próximo;
  3. Se **>**, verificamos se pelo menos um **<** já foi empilhado. Se sim, desempilhamos um **<** e computamos que +1 diamante foi encontrado. Se não, vamos para o próximo caractere.

Teste de Mesa para o Exemplo: **<...<<..>>....>....>>>**

1ª iteração: **<...<<..>>....>....>>>**

Primeiro **<** entra na Pilha: |**<**|

2ª até 4ª iteração: **<...<<..>>....>....>>>**

Caracteres **.** são ignorados.

5ª iteração: **<...<<..>>....>....>>>**

Segundo **<** entra na Pilha: |**<**| -- |**<**|

6ª iteração: **<...<<..>>....>....>>>**

Terceiro **<** entra na Pilha: |**<**| -- |**<**| -- |**<**|

7ª até 8ª iteração: **<...<<..>>....>....>>>**

Caracteres **.** são ignorados.

9ª iteração: **<...<<..>>....>....>>>**

Sendo o caractere igual à **>**, verifica-se que a pilha não está vazia, então:

O número de diamantes é incrementado em +1 (diamantes = 1) e um dos sinais **<** é desempilhado: |**<**| -- |**<**|

10ª iteração: **<...<<..>>....>....>>>**

Sendo o caractere igual à **>**, verifica-se que a pilha não está vazia, então:

O número de diamantes é incrementado em +1 (diamantes = 2) e um dos sinais **<** é desempilhado: |**<**|

11ª até 14ª iteração: **<...<<..>>....>....>>>**

Caracteres **.** são ignorados.

15ª iteração: **<...<<..>>....>....>>>**

Sendo o caractere igual à **>**, verifica-se que a pilha não está vazia, então:

O número de diamantes é incrementado em +1 (diamantes = 3) e um dos sinais **<** é desempilhado: pilha vazia.

16ª até 19ª iteração: **<...<<..>>....>....>>>**

Caracteres **.** são ignorados.

20ª até 22ª iteração: **<...<<..>>....>....>>>**

Sendo o caractere igual à **>**, verifica-se que a pilha está vazia, então:

O caractere **>** é ignorado.