

## ALGORITMOS Y ESTRUCTURA DE DATOS

- Se desea extraer la mayor cantidad de diamantes "<>". Se debe evitar todas las partículas de arena encontrada (".") en este proceso y luego el diamante se puede extraer y nuevos diamantes se pueden formar. De la siguiente entrada "<... << .. >> ....> .... >>>" tres diamantes se han formado. El primero se toma de <..> resultando "<... <> ....> .... >>>", el segundo diamante es entonces retirado, dejando "<.....> .... >>>", el tercer diamante es entonces retirado, dejando al final "..... >>>", con la posibilidad de extraer nuevos diamantes.

### Entrada

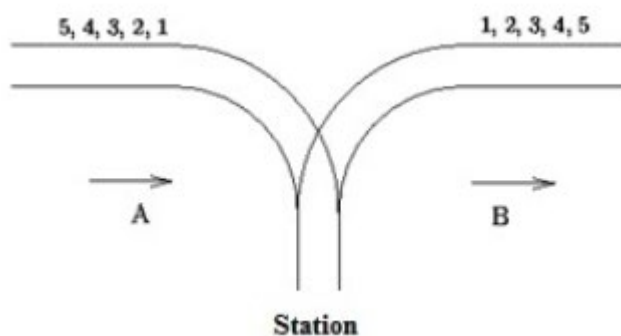
Leer un entero N que es el número de casos de prueba. Entonces siguen las N líneas, cada una hasta 1000 caracteres, incluyendo "<", ">" y ".".

### Salida

Debes imprimir la cantidad de diamantes que se han podido extraer en cada caso de prueba.

Ejemplo de entrada	Ejemplo de salida
2	3
<...><...>>	1
<<<...<.....<<<<...>	

- Se tiene la siguiente estación de tren, cuyas direcciones de desplazamiento son las que indican las letras A y B, tal como indica la figura.



Cada tren que llega desde la dirección A continúa en dirección B con los vagones organizados de alguna manera. Supongamos que el tren que llega desde la dirección A tiene  $N \leq 1000$  vagones numerados en orden creciente  $1, 2, \dots, N$ . El jefe de organización de trenes debe saber si es posible ordenar los vagones, continuando en la dirección B para que su orden sea  $a_1, a_2, a_n$ . Escribir un programa que diga si es posible obtener el orden requerido de vagones.

Se puede asumir que los vagones pueden ser desconectados individualmente del tren antes de que entren a la estación, y, que pueden moverse por su cuenta hasta que estén en la pista de la dirección B. Se puede también suponer que en cualquier momento pueden ser localizados tantos vagones, como sea necesario, en la estación. Pero una vez que un vagón ha entrado en la estación no puede salir por la pista de la dirección A y una vez que ha dejado la estación en la dirección B no puede volver a entrar.

### Entrada

El archivo de entrada consta de varios bloques con varias líneas. Cada bloque, excepto el último, describe un tren. En la primera línea del bloque está el entero N descrito anteriormente. En cada una de las siguientes líneas del bloque hay una permutación de  $1, 2, \dots, N$ . La última línea del bloque contiene sólo cero (0).

### Salida

La salida contiene varias líneas, que se corresponden a cada una de las líneas de permutaciones de la entrada. Una línea de la salida muestra "Yes" si es posible ordenar los vagones en el orden especificado en la línea correspondiente de la entrada. De lo contrario, muestra "No". Además, hay una línea vacía después de las líneas correspondientes a cada bloque de entrada. No hay ninguna línea de salida correspondiente al último bloque "nulo" del archivo de entrada.

Ejemplo de entrada	Ejemplo de salida
5	Yes
5 4 3 2 1	Yes
1 2 3 4 5	No
5 4 1 2 3	
0	Yes
6	
1 3 2 5 4 6	
0	
0	