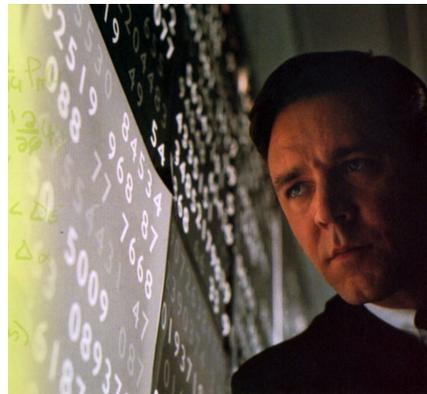


# Codificación espejo

Tiempo máximo: 4,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=228>

El envío de mensajes cifrados para evitar miradas indiscretas se lleva estudiando desde la antigüedad. El método más simple consiste en manejar tablas de traducción que contienen, para cada letra, por qué otra letra se sustituirá en el mensaje cifrado. El descifrado maneja las tablas inversas que permiten recomponer el mensaje original. Además, para dificultar el trabajo a los posibles espías, muchos de estos mecanismos primitivos *no* consideran el espacio separador de palabras como parte del mensaje por lo que el receptor al decodificar ve todas las letras seguidas y tiene que ser él, a la vista de las mismas, el que separe las palabras.

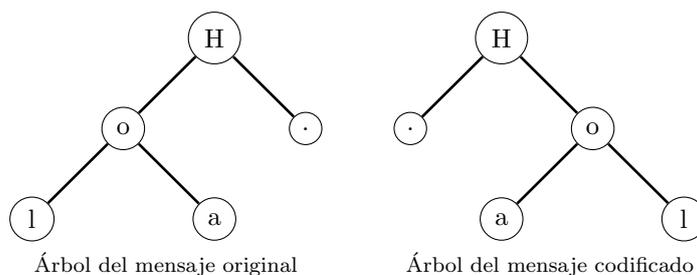


Existe otro mecanismo simple que consiste en, sencillamente, *cambiar de orden* las letras, siguiendo unas determinadas reglas. Si esas reglas están bien elegidas, el proceso de codificación y el de decodificación es el mismo, por lo que no se necesita implementar algoritmos distintos en cada lado de la comunicación.

El método que hoy proponemos cumple, parcialmente, esta propiedad. Y, como decíamos antes, no codifica los espacios, por lo que el receptor tendrá que colocarlos él mismo a partir del significado de lo que vaya leyendo.

El mecanismo de codificación/decodificación se basa en representar el mensaje como un árbol binario tal que su recorrido en *preorden* es el mensaje a codificar (sin espacios); lo normal es que existan muchos posibles árboles que cumplan lo anterior, cualquiera de ellos vale. La codificación espejo entonces construye un nuevo árbol invirtiendo el original para conseguir su *imagen especular*. El recorrido en preorden de ese nuevo árbol es el mensaje que se envía.

A modo de ejemplo, imaginemos que queremos enviar el mensaje “H01a.”. En la figura de la izquierda aparece un posible árbol binario cuyo recorrido en preorden coincide con el mensaje que queremos transmitir. A la derecha aparece su imagen especular, cuyo recorrido en preorden es el que se envía.



Para que el receptor del mensaje pueda recomponer el árbol enviado a partir únicamente del recorrido en preorden, se aprovecha que los espacios del propio mensaje no se envían. Gracias a eso, se pueden utilizar de forma segura para indicar ausencia de hijo. Es por esto que la forma de transmitir el árbol de la derecha es:

- Primero aparece la raíz del árbol, la letra H.
- A continuación aparece el hijo izquierdo completo. Es decir el carácter . seguido de dos espacios que indican que ese nodo no tiene hijos.
- Posteriormente aparece el hijo derecho, cuya codificación comienza con su raíz, o, seguida del hijo izquierdo (a y dos espacios) y el hijo derecho (l y dos espacios adicionales).

## Entrada

La entrada consiste en diversas líneas, cada una con un mensaje codificado utilizando la codificación espejo. Se garantiza que cada línea representará un árbol binario válido de un máximo de 5.000 nodos y cuya altura no será mayor que 3.000.

**Importante:** en el ejemplo de entrada las tres líneas terminan con *dos* espacios tras el último carácter visible; en caso contrario la línea no representaría correctamente el árbol binario.

## Salida

Para cada caso de prueba, el programa escribirá una línea con el mensaje descifrado. Ten en cuenta que lo que escribimos es el mensaje descifrado y no el recorrido en preorden del árbol espejo, por lo que en la salida *no* aparecerán los espacios.

## Entrada de ejemplo

```
H.  oa  l  
Hol  a  .  
as  do  i
```

## Salida de ejemplo

```
Hola.  
H.oal  
adios
```

**Autor:** Marco Antonio Gómez Martín.

**Revisor:** Pedro Pablo Gómez Martín.