



## Codificación

Encodelia quiere enviar un mensaje a Decodelio. El mensaje consiste en una cadena binaria de bits. Por desgracia, la malvada Errorlina interferirá en la comunicación. Errorlina seleccionará aleatoriamente un intervalo de bits contiguos de la cadena enviada por Encodelia y lo invertirá, es decir, cambiará los unos a ceros y los ceros a unos.

Por ejemplo, si Encodelia envía el mensaje `01011001`, Errorlina puede seleccionar el intervalo `01011001`, de forma que Decodelio recibirá el mensaje `01100111`.

Ayuda a Encodelia y a Decodelio a comunicarse pese a la interferencia de Errorlina. Tu programa se ejecutará tanto desde el punto de vista de Encodelia como de Decodelio: en el primer caso se te dará el mensaje que Encodelia quiere transmitir y debes imprimir una codificación del mensaje que enviar a Decodelio, y en el segundo recibirás la codificación tras la interferencia de Errorlina y debes recuperar el mensaje original.

## Entrada y Salida

**Este es un problema de ejecución múltiple.** Por cada caso de prueba, tu programa se ejecutará dos veces seguidas de forma independiente.

En la primera ejecución, la entrada consistirá en una línea con la palabra `ENC` seguida de una cadena binaria, el mensaje que se quiere enviar. La palabra `ENC` y la cadena binaria están separadas por un espacio. Tu programa debe imprimir una línea con una cadena binaria, la codificación que se desea enviar.

En la segunda ejecución, la entrada consistirá en una línea con la palabra `DEC` seguida de una cadena binaria, el mensaje codificado recibido. La palabra `DEC` y la cadena binaria están separadas por un espacio. Tu programa debe imprimir una línea con una cadena binaria correspondiente al mensaje original.

## Ejemplo

Una posible secuencia de ejecuciones sería la siguiente. En la primera ejecución, la entrada es:

```
ENC 011
```

Y la salida que imprime el programa es:

```
01011001
```

En la segunda ejecución, la entrada es:

```
DEC 01100111
```

Nótese que se ha invertido un intervalo, como se ha explicado antes en el enunciado. La salida debería ser:

```
011
```

## Restricciones

La longitud del mensaje a codificar será de como mucho  $10^4$  bits.

La longitud de la codificación debe ser de como mucho  $10^5$  bits.

Tu programa será probado en un número limitado de casos de prueba. El intervalo de la codificación que se invertirá se seleccionará (pseudo)aleatoriamente de forma independiente



en cada caso, con una distribución uniforme sobre todos los  $\frac{\ell \cdot (\ell + 1)}{2}$  intervalos posibles (donde  $\ell$  es la longitud de la codificación).

### Subtareas

1. (25 puntos) La longitud del mensaje a codificar será de como mucho 10 bits.
2. (75 puntos) Sin restricciones adicionales.

Adicionalmente, la puntuación que obtienes depende de la longitud del mensaje codificado: para obtener una puntuación completa la longitud debe ser como máximo  $2 \cdot 10^4$  bits y para obtener una puntuación positiva la longitud debe ser como máximo  $10^5$  bits. La puntuación de cada subtarea es multiplicada por un multiplicador  $M(\ell)$ , donde  $\ell$  es la máxima longitud en bits de una de las codificaciones producidas en los casos de esa subtarea. El valor de  $M(\ell)$  viene dado por:

$$M(\ell) = \begin{cases} 0 & \ell > 10^5 \\ 0,9 + \frac{10^5 - \ell}{8 \cdot 10^5} & 10^5 \geq \ell > 2 \cdot 10^4 \\ 1,0 & 2 \cdot 10^4 \geq \ell \end{cases}$$