

Run-Length Encoding (RLE)

Dada una cadena s compuesta por letras minúsculas ($a-z$), implementa:

$rle_encode(s)$: comprime s usando *Run-Length Encoding* escribiendo, para cada *run* de un mismo carácter, el carácter seguido del conteo en decimal. Ejemplo: $aaabbc \rightarrow a3b2c1$.

$rle_decode(s)$: descomprime una cadena en formato RLE válido (carácter $[a-z]$ seguido de un entero decimal positivo) y devuelve la original. Ejemplo: $a3b2c1 \rightarrow aaabbc$.

Reglas

En encode, siempre emite el conteo (incluido 1).

En decode, el formato de entrada es válido (no necesitas validar errores): cada bloque es $\langle \text{letra} \rangle \langle \text{entero} \rangle = \langle \text{conteo} \rangle$, con $\text{conteo} \geq 1$. Las cadenas pueden ser grandes se debe manejar las reservas de memoria.

Formato de entrada

Archivo cuya, primera línea contiene una palabra MODE, que será ENCODE o DECODE. La segunda línea contiene la cadena s (solo letras $a-z$; puede ser vacía).

Formato de salida

Imprime una única línea con el resultado de aplicar la operación indicada por MODE a s .

Restricciones

$0 \leq |s| \leq 10^6$. s solo contiene $[a-z]$ (para DECODE, la cadena está en formato RLE válido). Se debe usar memoria dinámica y liberar lo reservado.

Ejemplos

Input	Output
ENCODE aaabbc	A3b2c1
DECODE a12b1	aaaaaaaaaaaab
ENCODE	(línea vacía)