

## A. La Noria (ferriswheel)

Límite de tiempo: 1 segundos

Límite de memoria: 1024 MiB

En la plaza principal de Cesenatico hay una colorida noria (rueda de la fortuna), una de las atracciones más icónicas de la ciudad. Durante el invierno, la desmontaron y la guardaron, ¡pero ahora que ya casi llega el verano es momento de armarla de nuevo! Las piezas acaban de llegar a la plaza y, con tu ayuda, estamos listos para poner todo en su lugar.

Frente a ti hay  $N$  cabinas individuales que deben unirse, en orden circular, para formar la noria. Las cabinas están numeradas del 0 al  $N - 1$ , pero no necesariamente en el orden en que deben conectarse.

Cada cabina trae una unión especial que sirve para conectarla con la siguiente cabina en sentido horario. Cada unión tiene uno de dos tipos posibles:

- Tipo  $[+]$ : solo sirve para conectar con una cabina de número mayor;
- Tipo  $[-]$ : solo sirve para conectar con una cabina de número menor.

En el ejemplo de abajo, la cabina 2 tiene una unión de tipo  $[+]$ . Esto significa que la siguiente cabina en sentido horario debe ser la 3 o la 4.

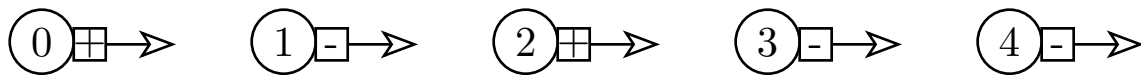


Figura 1:  $N = 5$  y cinco cabinas separadas, cada una con una unión de tipo  $[+]$  o  $[-]$ .

Te damos el número de cabinas y los tipos de sus uniones. Tu trabajo es determinar si es posible armar las  $N$  cabinas para formar la noria. Si es así, también tienes que encontrar un orden en el que las cabinas puedan aparecer en la rueda.

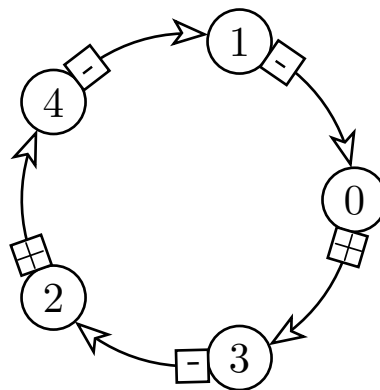


Figura 2: Una noria válida que se puede armar con las cinco cabinas mostradas arriba.

La imagen de arriba muestra una noria válida que se puede armar con las cinco cabinas mostradas arriba.

Formalmente, un orden válido de las cabinas es una secuencia  $C_0, C_1, \dots, C_{N-1}$  de números con las siguientes propiedades:

- Cada número del 0 al  $N - 1$  aparece exactamente una vez en la secuencia.
- Para cada  $0 \leq i \leq N - 2$ , la cabina  $C_{i+1}$  debe cumplir la condición impuesta por el tipo de unión de la cabina  $C_i$ . Es decir, si el tipo de unión de la cabina  $C_i$  es  $[+]$ , entonces  $C_{i+1} > C_i$ ; si es  $[-]$ , entonces  $C_{i+1} < C_i$ .

- Además, la cabina  $C_0$  debe cumplir la condición impuesta por el tipo de unión de la cabina  $C_{N-1}$ .

## Entrada

La entrada consiste en dos líneas. La primera línea contiene un número entero  $N$ , que indica el número de cabinas.

La segunda línea contiene una cadena  $S$  de longitud  $N$ , formada por caracteres “+” y “-”. Si  $S_i = “+”$ , entonces la cabina  $i$  tiene una unión de tipo [+]. Si  $S_i = “-”$ , entonces la cabina  $i$  tiene una unión de tipo [-].

## Salida

Si no hay ningún orden que cumpla las restricciones, imprime NO.

De lo contrario, imprime YES, seguido de una línea con  $N$  enteros, los índices de las cabinas en una noria válida en sentido horario, empezando por cualquier índice. Si hay varias soluciones, puedes imprimir cualquiera de ellas.

## Restricciones

- $3 \leq N \leq 300\,000$ .
- $S_i = “+”$  o “-”.

## Puntuación

Tu programa será probado en varios casos de prueba agrupados en subtareas. Para obtener el puntaje de una subtarea, debes resolver correctamente todas las pruebas que contiene.

- **Subtask 0 [ 0 puntos]:** Ejemplos.
- **Subtask 1 [16 puntos]:**  $N = 3$ .
- **Subtask 2 [13 puntos]:** Hay exactamente un “+” en la cadena.
- **Subtask 3 [24 puntos]:** Los caracteres “+” y “-” se alternan en la cadena  $S$ ; es decir, para cada  $i = 0, \dots, N - 2$ , se cumple que  $S_i \neq S_{i+1}$ .
- **Subtask 4 [23 puntos]:**  $N \leq 1000$ .
- **Subtask 5 [24 puntos]:** Sin restricciones adicionales.

## Ejemplos de entrada/salida

stdin	stdout
3 +++	NO
5 +-+--	YES 0 3 2 4 1
7 -----+	NO
8 +-+--+--	YES 3 2 4 6 7 1 0 5
11 ++++-+--+--	YES 10 0 5 8 9 4 2 6 3 1 7

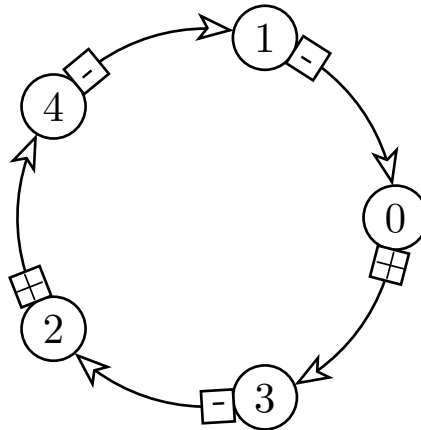
## Explicación

**Primer Ejemplo.** Nos dan tres cabinas. Como todas las uniones son de tipo [+], debemos organizar las cabinas de modo que cada una vaya seguida de una cabina con un número mayor. Se puede demostrar que ningún orden de las tres cabinas cumple esta condición, por lo tanto la respuesta es NO.

**Segundo Ejemplo.** Mira la figura en el enunciado del problema. Nos dan cinco cabinas. Debemos organizarlas en sentido horario de modo que:

- las cabinas 0 y 2 (unión tipo [+]) vayan seguidas inmediatamente de una cabina con un número mayor;
- las cabinas 1, 3 y 4 (unión tipo [-]) vayan seguidas inmediatamente de una cabina con un número menor.

Una noria que cumple todas estas condiciones se muestra en la figura de abajo. Para las uniones de tipo [+], los requisitos se cumplen ya que  $0 < 3$  y  $2 < 4$ . Para las uniones de tipo [-], los requisitos se cumplen ya que  $1 > 0$ ,  $3 > 2$ , y  $4 > 1$ . Hay varias salidas que corresponden a esta noria: en lugar de 0 3 2 4 1 también puedes imprimir 3 2 4 1 0, 2 4 1 0 3, 4 1 0 3 2, o 1 0 3 2 4.



En el tercer ejemplo, nos dan siete cabinas: todas las uniones son de tipo [-], excepto la última, que es de tipo [+]. Por lo tanto, debemos organizar las cabinas de modo que cada cabina vaya seguida de una con un número menor, excepto la cabina 6, que debe ir seguida de una con un número mayor. Se puede demostrar que no existe tal orden, por lo que la respuesta es NO.

Las figuras de abajo muestran las norias que corresponden a las salidas de los dos últimos ejemplos.

