A. Ferris Wheel (ferriswheel)

Límite de tiempo: 1 segundos
Límite de memoria: 1024 MiB

La plaza principal de Cesenatico cuenta con una colorida noria, una de las atracciones emblemáticas de la ciudad. Durante el invierno, la noria fue desmontada y almacenada, pero ahora que el verano ya casi está aquí, ¡finalmente es hora de construirla de nuevo! Las piezas acaban de llegar a la plaza, y con tu ayuda estamos listos para montarlas todas.

Ante ti hay N cabinas individuales que deben unirse entre sí, de forma circular, para formar la noria. Las cabinas están numeradas del 0 al $N-1$, pero no necesariamente en el orden en que deben ser unidas.

Cada cabina viene con una unión especial que se utiliza para conectarla a la siguiente cabina en sentido horario. Cada unión puede ser de dos tipos posibles:

- Tipo [+]: solo puede usarse para conectar con una cabina de número mayor;
- Tipo [−]: solo puede usarse 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 cabina 3 o la 4.

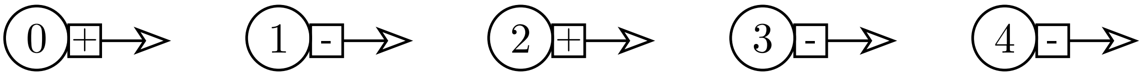

Figura 1: $N = 5$ y cinco cabinas separadas, cada una con una unión de tipo [+] o [−].

Te dan el número de cabinas y los tipos de unión de cada una. Tu tarea es determinar si es posible montar las N cabinas para formar una noria. Si es así, también debes encontrar un orden en el que las cabinas pueden aparecer en la noria.

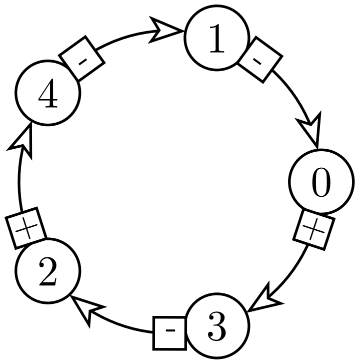

Figura 2: Una noria válida que se puede montar usando las cinco cabinas mostradas arriba.

La Figura 2 muestra una noria válida que se puede montar a partir de las cinco cabinas mostradas en la Figura 1.

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

- Cada número del 0 al $N-1$ aparece exactamente una vez en la secuencia.
- Para cada $0 \le i \le N-2$, 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 entero N, que denota el número de cabinas.

La segunda línea contiene una cadena S de longitud N, formada por los 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 existe un orden que satisfaga las restricciones, muestra `NO`.

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

Restricciones

- $3 \leq N \leq 300\,000$.
- S_i = "`+`" o "`-`".

Puntuación

Tu programa será probado con varios casos de prueba agrupados en subtareas. Para obtener la puntuación de una subtarea, debes resolver correctamente todos los tests que contiene.

- **Subtask 0** [**0 puntos**]: Ejemplos.
- **Subtask 1** [**16 puntos**]: $N = 3$.
- **Subtask 2** [**13 puntos**]: Hay exactamente un "`+`" en la cadena S.
- **Subtask 3** [**24 puntos**]: Los caracteres "`+`" y "`-`" se alternan en la cadena S; es decir, para todo $0 \leq i \leq N-2$, 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. Dado que 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 satisface esta condición, por lo tanto la respuesta es `NO`.

Segundo Ejemplo. Ver las Figuras 1 y 2 en el enunciado del problema. Nos dan cinco cabinas. Debemos organizarlas en sentido horario de tal manera que:

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

Una noria que satisface 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 múltiples salidas que corresponden a esta noria: en lugar de `0 3 2 4 1` también puedes mostrar `3 2 4 1 0`, `2 4 1 0 3`, `4 1 0 3 2`, o `1 0 3 2 4`.

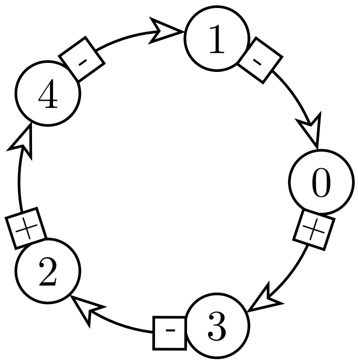

Figura 3: La noria del segundo Ejemplo (esta figura es idéntica a la Figura 2).

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 tal orden no existe, por lo que la respuesta es `NO`.

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

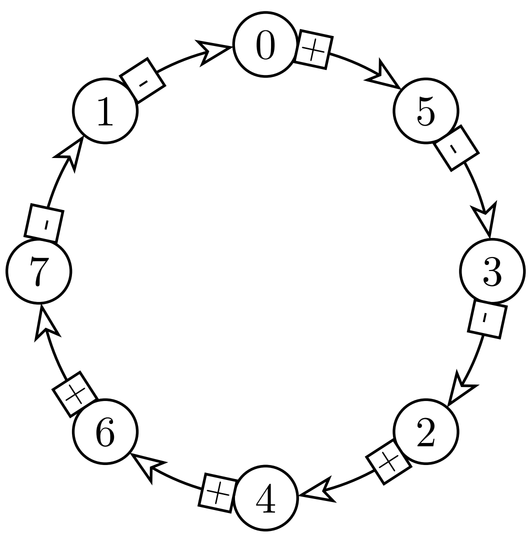

Figura 4: La noria del cuarto Ejemplo.

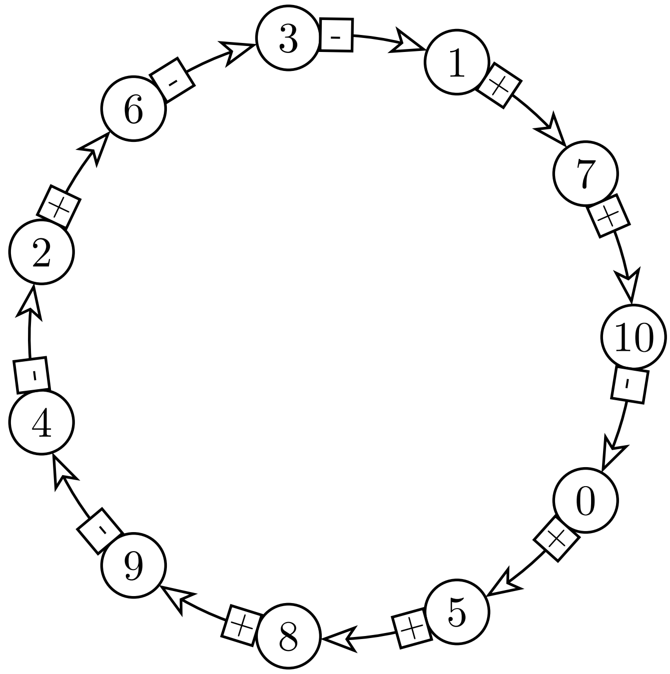

Figura 5: La noria del quinto Ejemplo.