A. Grande roue (`ferriswheel`)

Limite de temps: 1 secondes
Limite de mémoire: 1024 MiB

La place principale de Cesenatico dispose d'une grande roue colorée, l'une des attractions emblématiques de la ville. Pendant l'hiver, la roue a été démontée pour être stockée, mais maintenant que l'été approche, il est enfin temps de la remonter ! Les pièces viennent d'arriver sur la place, et avec votre aide, nous sommes prêts à tout assembler.

Devant vous se trouvent N cabines individuelles qu'il faut attacher les unes aux autres de manière circulaire, pour construire la grande roue. Les cabines sont numérotées de 0 à $N-1$, mais pas forcément dans l'ordre dans lequel elles doivent être attachées.

Chaque cabine est équipée d'une attache spéciale qui sert à la relier à la cabine suivante dans le sens des aiguilles d'une montre. Chaque attache est de l'un des deux types suivants :

- Type [+] : ne peut être utilisée que pour se connecter à une cabine ayant un numéro plus grand ;
- Type [−] : ne peut être utilisée que pour se connecter à une cabine ayant un numéro plus petit.

Dans l'exemple ci-dessous, la cabine 2 possède une attache de type [+]. Cela signifie que la cabine suivante dans le sens des aiguilles d'une montre doit être soit la cabine 3, soit la cabine 4.

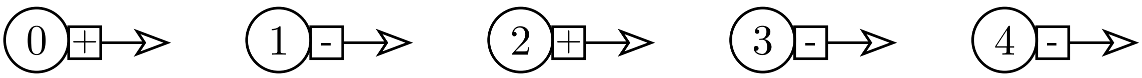

Fig. 1. – $N = 5$ et cinq cabines séparées, chacune avec une attache de type [+] ou [−].

On vous donne le nombre de cabines et leurs types d'attache. Votre objectif est de déterminer s'il est possible d'assembler les N cabines pour former une grande roue. Si oui, vous devez également trouver un ordre dans lequel les cabines peuvent apparaître sur la roue.

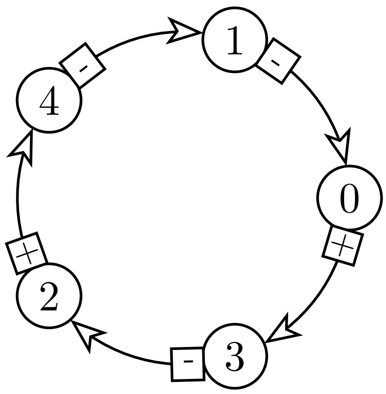

Fig. 2. – Une grande roue valide pouvant être assemblée à partir des cinq cabines montrées ci-dessus.

La figure 2 ci-dessus montre une grande roue valide pouvant être assemblée à partir des cinq cabines montrées figure 1.

Formellement, un ordre valide de cabines est une séquence $C_0, C_1, \ldots, C_{N-1}$ de nombres possédant les propriétés suivantes :

- Chaque nombre de 0 à $N-1$ apparaît exactement une fois dans la séquence.

- Pour chaque $0 \leq i \leq N-2$, la cabine C_{i+1} doit satisfaire la condition imposée par l'attache de la cabine C_i. C'est-à-dire que si l'attache de la cabine C_i est de type [+], alors $C_{i+1} > C_i$; si elle est de type [-], alors $C_{i+1} < C_i$.
- De plus, la cabine C_0 doit satisfaire la condition imposée par l'attache de la cabine C_{N-1}.

Entrée

L'entrée se compose de deux lignes. La première ligne contient un entier N, indiquant le nombre de cabines.

La deuxième ligne contient une chaîne S de longueur N, composée des caractères "+" et "-". Si $S_i =$ "+", alors la cabine i possède une attache de type [+]. Si $S_i =$ "-", alors la cabine i possède une attache de type [-].

Sortie

S'il n'existe aucun ordre satisfaisant les contraintes, affichez `NO`.

Sinon, affichez `YES`, suivi d'une ligne de N entiers, les indices des cabines sur une grande roue valide dans le sens des aiguilles d'une montre, en commençant par n'importe quel indice. S'il y a plusieurs solutions, vous pouvez afficher n'importe laquelle.

Contraintes

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

Score

Votre programme sera évalué sur plusieurs tests, regroupés en sous-tâches. Pour obtenir les points d'une sous-tâche, vous devez résoudre correctement tous les tests qu'elle contient.

- **Sous-tâche 0** [**0 points**]: Exemples.
- **Sous-tâche 1** [**16 points**]: $N = 3$.
- **Sous-tâche 2** [**13 points**]: Il y a exactement un "+" dans la chaîne S.
- **Sous-tâche 3** [**24 points**]: Les caractères "+" et "-" alternent dans la chaîne S ; c'est-à-dire que pour tout $0 \leq i \leq N-2$, on a $S_i \neq S_{i+1}$.
- **Sous-tâche 4** [**23 points**]: $N \leq 1000$.
- **Sous-tâche 5** [**24 points**]: Aucune contrainte supplémentaire.

Exemples

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

Explication

Premier exemple. On nous donne trois cabines. Comme toutes les attaches sont de type [+], nous devons disposer les cabines de manière à ce que chaque cabine soit suivie d'une cabine ayant un numéro plus grand. On peut montrer qu'aucun ordre des trois cabines ne satisfait cette condition, par conséquent la réponse est `NO`.

Deuxième exemple. L'exemple correspond aux figures 1 et 2 dans l'énoncé du problème. On nous donne cinq cabines. Nous devons les disposer dans le sens des aiguilles d'une montre de telle sorte que :

- les cabines 0 et 2 (attache de type [`+`]) soient immédiatement suivies d'une cabine ayant un numéro plus grand ;
- les cabines 1, 3 et 4 (attache de type [`-`]) soient immédiatement suivies d'une cabine ayant un numéro plus petit.

Une grande roue satisfaisant toutes ces conditions est montrée dans la figure ci-dessous. Pour les attaches de type [`+`], les conditions sont respectées car $0 < 3$ et $2 < 4$. Pour les attaches de type [`-`], les conditions sont respectées car $1 > 0$, $3 > 2$ et $4 > 1$. Il y a plusieurs sorties possibles correspondant à cette grande roue : au lieu de `0 3 2 4 1` vous pouvez aussi afficher `3 2 4 1 0`, `2 4 1 0 3`, `4 1 0 3 2`, ou `1 0 3 2 4`.

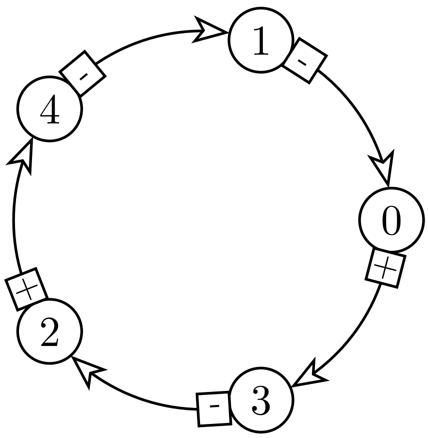

Fig. 3. – La grande roue du deuxième exemple (cette figure est identique à la figure 2.)

Dans le troisième exemple, on nous donne sept cabines : toutes les attaches sont de type [`-`], sauf la dernière qui est de type [`+`]. Ainsi, nous devons disposer les cabines de telle sorte que chaque cabine soit suivie d'une cabine ayant un numéro plus petit, sauf la cabine 6, qui doit être suivie d'une cabine ayant un numéro plus grand. On peut montrer qu'aucun ordre de ce type n'existe, donc la réponse est `NO`.

Les figures ci-dessous montrent une grande roue pour chacun des deux derniers exemples.

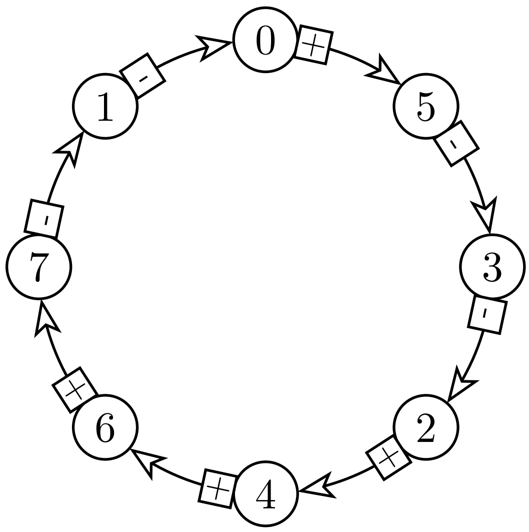

Fig. 4. – La grande roue du quatrième exemple.

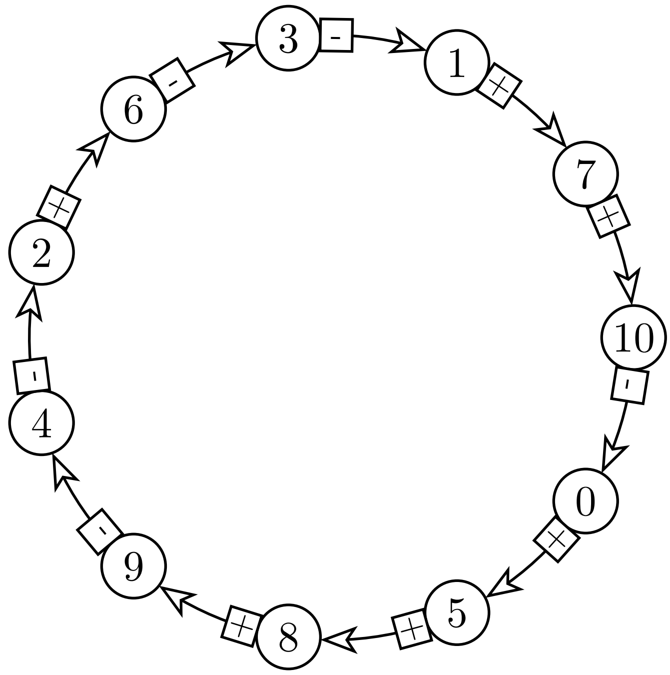

Fig. 5. – La grande roue du cinquième exemple.