

Cadena de bits

Una cadena de bits es una secuencia de unos (1) y ceros (0). En computadoras, estas cadenas de bits se encuentran por doquier. Los datos e instrucciones se representan a niveles muy bajos como cadenas de bits. Por ejemplo, el número 14 se puede representar como $1_31_21_10_0$. Esta efectivamente es una representación en base 2 de los números: $2^3 * 1 + 2^2 * 1 + 2^1 * 1 + 2^0 * 0 = 14$.

Los bits fueran inútiles si no se pudiera operar sobre ellos. El operador lógico AND opera sobre dos bits y produce un nuevo bit con valor 1 si los bits operandos tienen el valor 1, en cualquier otro caso, produce 0.

```
AND 0 1
0   0 0
1   0 1
```

Algo interesante de este operador es que puede extenderse a cadenas de bits. Esto es, este operador produce una cadena de bits resultante donde el i -ésimo bit de éste representa el resultado de la aplicación del operador AND a los i -ésimos bits de las cadenas operandos. Un ejemplo es el siguiente:

```
1010101
1101110
=====
1000100
```

Tu problema aquí es bastante sencillo. Recibirás una cadena de bits A, y dos enteros en base 10, B y C. Tu objetivo es determinar si, aplicando $A \text{ AND } B$, puedes obtener C. Para ello, puedes reordenar los bits de la cadena de bits resultante.

1. Para aplicar el operador AND, los operandos necesitan estar en representación de base 2.
2. Siempre se puede aplicar el operador AND a dos cadenas de bits, aunque estas no tengan la misma cantidad de bits. Para ello, primero se igualan en longitud ambas cadenas de bits, llenando de ceros por la izquierda la cadena de bits más pequeña.

Input

La entrada consistirá de dos líneas.

La primera línea contendrá la cadena de bits A, con longitud entre 1 y 60. Cada uno de los caracteres de la primera línea podrá ser '0' o '1'.

La segunda línea tendrá los enteros en base 10, B y C ($0 \leq B, C \leq 10^{18}$), separados por un espacio. Se garantiza que B tendrá, en su representación binaria, una cantidad igual o menor de bits que A.

Output

Si es posible encontrar una permutación de los bits de A AND B que produzca C, imprime en una primera línea "SI". De lo contrario, imprime "NO". En ningún caso imprime comillas.

Example

Input Output

```
1      SI
1 1
10     SI
2 1
10     SI
1 0
```