

Problem D. Laberintuerca

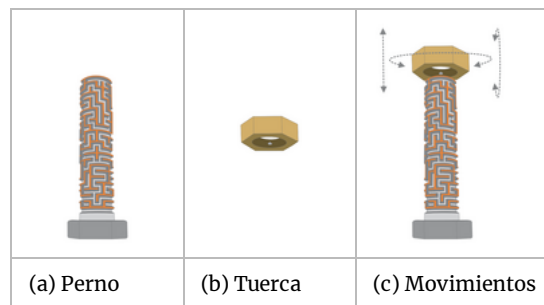
Time limit 300 ms
Mem limit 1048576 kB
Statement [Statements \(PDF\)_\(en\)](#)
OS Windows

En el FabLab de la Universidad, diseñaron un novedoso puzzle usando impresión 3D, llamado Laberintuerca.

El puzzle está compuesto de dos partes: una pieza con forma de perno con un laberinto en relieve grabado en él, y una tuerca. La parte interna de la tuerca puede contener puntas, las cuales restringen que la tuerca se deslice solo por los pasillos del laberinto.

Inicialmente, las dos piezas están separadas. El desafío es deslizar la tuerca a través de todo el perno hasta alcanzar la cabeza. La tuerca puede girarse en sentido horario, antihorario, y también puede ser bajada (hacia la cabeza del perno) y subida (alejándose de la cabeza). Cada uno de estos movimientos solo son posibles cuando todas las puntas de la parte interior de la tuerca no chocan con alguna muralla del laberinto.

Además de los movimientos anteriores, se permite uno más: cuando las piezas están separadas, el perno se puede voltear. La siguiente ilustración muestra ambas partes del puzzle y los movimientos permitidos:



Un cliente compró una gran cantidad de Laberintuercas. Cada puzzle diseñado por el FabLab es aleatorio y único, pero dado el volumen de compras y una fecha límite cercana, en el FabLab no creen ser capaces de verificar manualmente si todos los puzzles tienen solución o no.

Desde el FabLab te pidieron a ti, persona experta en algoritmos y teoría de grafos, diseñar un algoritmo que sea capaz de rápidamente verificar si un puzzle tiene solución. Para esto, la parte interior de la tuerca y cada fila del perno son modelados como strings binarios circulares (los extremos están conectados).

Entrada

La primera línea contiene dos enteros R ($1 \leq R \leq 1000$) y C ($3 \leq C \leq 1000$), indicando respectivamente la cantidad de filas y columnas del laberinto. La segunda línea contiene un string S de largo C , representando la parte interior de la tuerca. Cada caracter de S es **1** si la tuerca tiene una punta en la posición correspondiente y un **0** si no.

Cada una de las siguientes R líneas contienen un string circular de largo C indicando una fila del laberinto del perno. En este caso, se indica con un **1** si hay una muralla en la posición y un **0** si no. Las filas se dan desde arriba (la punta del perno) hasta abajo (la cabeza).

Salida

Si el laberinto tiene solución, imprime una letra **Y** mayúscula. Si no, imprime una **N** mayúscula.

Ejemplo 1

Entrada	Salida
<pre> 8 13 0110010000000 1100101110100 1001101000100 1100101110100 1000100010000 1010111011001 0000001010000 1001101111101 0001001100100 </pre>	Y

Ejemplo 2

Entrada	Salida
1 3 100 101	Y

Ejemplo 3

Entrada	Salida
2 3 100 101 010	N

Ejemplo 4

Entrada	Salida
4 6 001000 011111 010001 010100 110111	Y

Ejemplo 5

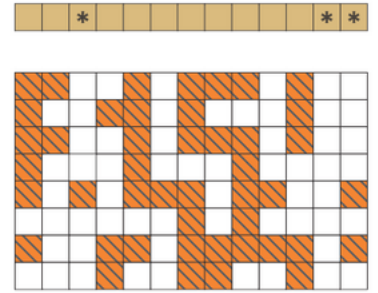
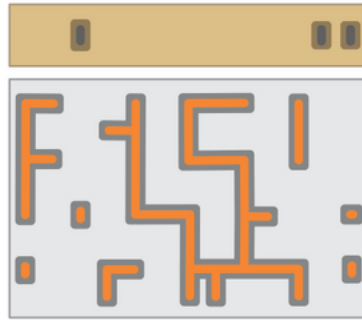
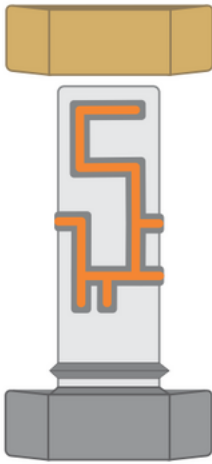
Entrada	Salida
1 6 001011 001011	Y

Note

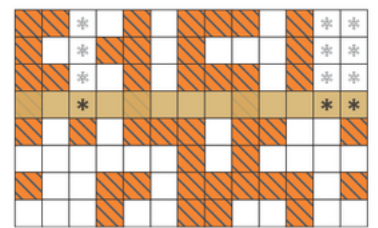
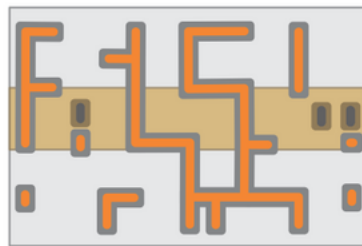
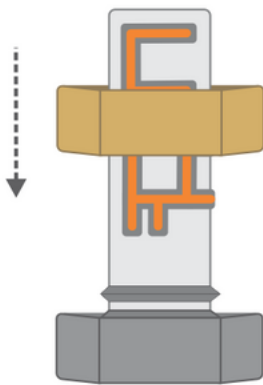
La siguiente imagen explica el primer caso de prueba. Las imagene sen la izquierda corresponden a vistas 3D de diferentes etapas del puzzle. En el medio, las imagenes son versiones 2D aplanadas de la vista 3D correspondiente. Finalmente, las imagenes en la derecha representan las etapas del puzzle acorde al modelo descrito (aunque por claridad, cada caracter 1 ha sido reemplazado por un símbolo y cada 0 se muestra como un espacio vacío).

Se puede ver en las primeras tres imágenes que la tuerca (con tres puntas) y el perno comienzan separados. El segundo grupo de tres imágenes muestran la situación luego de que el perno se ha movido cuatro filas hacia la cabeza. Luego, el perno es rotado una posición, movido hacia abajo dos filas más, rotado cuatro posiciones en la dirección opuesta y finalmente movido tres filas más hacia abajo, lo que resuelve el puzzle. Nota que en este caso el perno no fue volteado al inicio ni movido hacia arriba durante el puzzle.

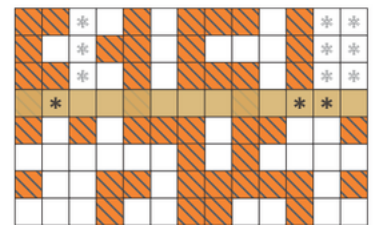
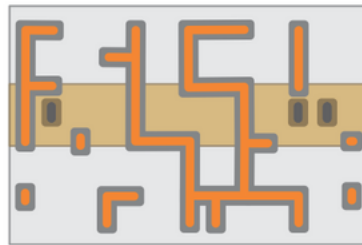
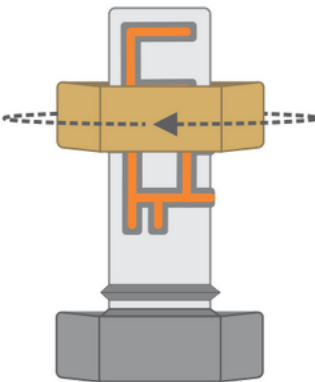
0x



4x



1x



2x

