Rotaciones

Dados dos vectores $V[0..N), W[0..N), N \geq 0$, y *cada uno con todos sus elementos distintos entre sí*, implementar una función en $\Theta(N)$ que decida si el uno es rotación del otro, esto es, fijando las posibles posiciones un elemento en común a los dos vectores (si lo tienen), el resto de valores aparece en el mismo orden en ambos, tomando el primer valor como el siguiente al último.

El problema formalmente queda especificado como sigue:

$\{ P : V[0..N) \wedge W[0..N) \wedge N \geq 0\}$
fun rotate(**int** V,**int** W, **int** N) **return b** boolean
$\{Q : b = \forall i : 0 \leq i < N : V[i] = W[(i + M)\%N]\}$

where $(0 \leq M \leq N) \wedge (M < N \rightarrow V[0] = W[M]))$

Expresar el(los) invariante(s) que puedas precisar, así como las funciones de cuota que permitan probar tanto la *corrección parcial* como la *terminación*.

Entrada

Cada caso de prueba consta de tres líneas. La primera línea indica el número de componentes N por vector. La segunda línea y tercera indican los vectores V, W cumpliendo la precondición descrita anteriormente (todos componentes son distintos). El programa acaba cuando no hay más casos de prueba.

Salida

Para cada caso de prueba se escribirá, en una línea diferente, `TRUE` si son rotación y `FALSE` en caso contrario.

Entrada de ejemplo

```
5
8 1 4 6 2
4 6 2 8 1
5
2 8 1 4 6
1 4 6 2 8
5
1 5 8 3 2
2 1 5 8 3
5
1 5 8 3 2
1 5 8 2 3
0
```

Salida de ejemplo

```
TRUE
TRUE
TRUE
FALSE
TRUE
```