

## Tiling\_Puzzle

คงเคยเล่นเกมเลื่อนแผ่นพลาสติกเล็กๆ จำนวน 15 อัน ให้เรียง 1 ถึง 15 จากซ้ายไปขวาลงล่าง ดังตัวอย่างในรูปข้างล่างนี้

1	7	2	3
6		8	4
5	9	10	11
13	14	15	12



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

ให้สังเกตว่า ไม่ใช่ทุกตารางจะสามารถเลื่อนไปสู่เป้าหมายที่ต้องการได้ เช่น การสลับ 14 กับ 15 ในตารางทางขวานี้ ก็ไม่สามารถเลื่อนไปสู่เป้าหมายได้ (ขอไม่พิสูจน์) โจทย์ข้อนี้เกี่ยวกับการตรวจตาราง ว่าเป็นตารางที่สามารถเลื่อนไปหาเป้าหมายได้หรือไม่ เราตรวจได้โดยไม่ต้องลงมือเลื่อนหมายเลขในตาราง ทำได้ดังนี้ (การตรวจสอบนี้ใช้ได้กับตารางที่มีขนาด  $n \times n$  ใดๆ และขอแทนช่องที่ว่างด้วย 0)

- นำจำนวนตารางในแต่ละแถวมาเขียนเรียงเป็นแถวเดียว (โดยไม่รวม 0) เช่น ตาราง  $3 \times 3$  ข้างล่างนี้ ทางซ้าย เขียนเรียงได้เป็นทางขวา

1 2 0	1 2 3 5 6 4 7 8
3 5 6	
4 7 8	

- หาจำนวน **inversion** ซึ่งคือจำนวนคู่ของข้อมูลในรายการ (ที่ได้เขียนเรียงเป็นแถวเดียว) ที่ตัวซ้ายมากกว่าตัวขวา

เช่น จากตัวอย่างข้างบนนี้ มีข้อมูล 8 ตัว ลองพิจารณาทุกคู่ ก็มีทั้งหมด  $8 \times 7 / 2 = 28$  คู่ ดังนี้

(1, 2), (1, 3), (1, 5), (1, 6), (1, 4), (1, 7), (1, 8), (2, 3), (2, 5), (2, 6), (2, 4), (2, 7), (2, 8), (3, 5), (3, 6), (3, 4), (3, 7), (3, 8), (5, 6), (5, 4), (5, 7), (5, 8), (6, 4), (6, 7), (6, 8), (4, 7), (4, 8), (7, 8)

จะเห็นว่า มี 2 คู่ที่ตัวซ้ายมากกว่าตัวขวา (แสดงด้วยสีแดง) ดังนั้น จำนวน **inversion** จึงมีค่าเป็น 2

- จากตารางที่ได้รับ จะสรุปว่า **สามารถเลื่อนไปสู่เป้าหมายได้** ก็เมื่อมีลักษณะตรงตามเงื่อนไขข้างล่างนี้

จำนวนแถวของตาราง	จำนวน <b>inversions</b>	หมายเลขแถวของตารางที่เลข 0 อยู่ (แถวบนสุดคือแถวที่ 0)
เลขคี่	เลขคู่	อยู่แถวใดก็ได้
เลขคู่	เลขคี่	เลขคู่
	เลขคู่	เลขคี่

จากตัวอย่าง ตารางมีจำนวน 3 แถวเป็นเลขคี่ จำนวน **inversions** คือ 2 เป็นเลขคู่ (และ 0 อยู่แถวที่ 0 ซึ่งในกรณีนี้เป็นแถวคู่แถวคี่ก็ได้) สรุปว่าสามารถเลื่อนไปยังเป้าหมายได้

จึงเขียนโปรแกรมอ่านตาราง แล้วหาว่า จะสามารถเลื่อนตัวเลขต่าง ๆ ไปสู่เป้าหมายได้หรือไม่

### ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนเต็ม **M** เพื่อบอกว่าเป็นตาราง  $M \times M$

**M** บรรทัดต่อมา แต่ละบรรทัดมีตัวเลข **M** ตัว ระบุหมายเลขที่อยู่ตามช่องต่างในแถวนั้น

### ข้อมูลส่งออก

ถ้าเราสามารถเลื่อนเลขในตารางที่ได้รับไปสู่เป้าหมายได้ ให้แสดง **YES** ถ้าเลื่อนไม่ได้ ให้แสดง **NO**

### ตัวอย่าง

input	output (ทางจอภาพ)
3 1 2 0 3 5 6 4 7 8	YES
3 1 2 0 3 6 5 4 7 8	NO
4 1 2 5 6 7 10 9 11 0 8 4 12 15 13 14 3	YES
4 1 2 6 5 7 10 9 11 0 8 4 12 15 13 14 3	NO