

Zigzag Series on a Grid

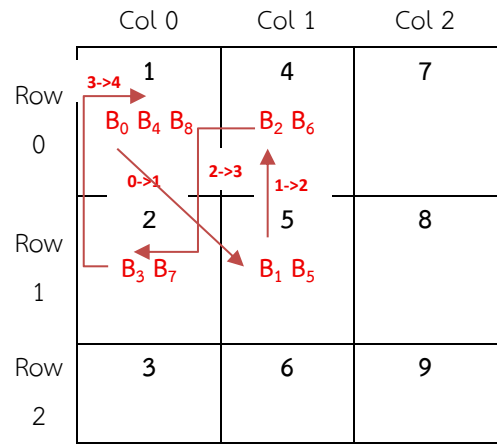
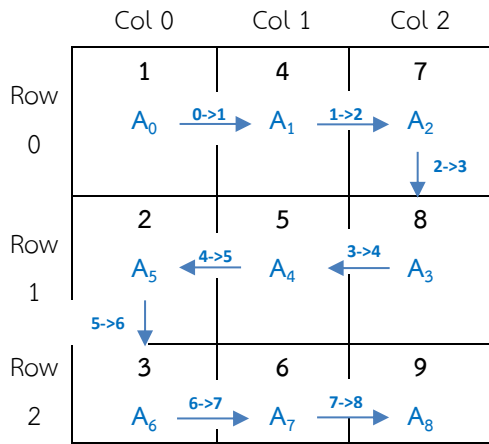
จงเขียนโปรแกรมเพื่อติดตามและคำนวณค่าของอนุกรม 2 ชุด โดยมาจากการเคลื่อนที่ของ Cursor A และ Cursor B บนตารางกริดของตัวเลขขนาด $N \times N$ โดยมีรูปแบบการเคลื่อนที่แบบ "ซิกแซก" เมื่อ Cursor แต่ละตัวเคลื่อนที่ไปหยุดที่ช่องไหน จะเก็บค่าจากช่องที่มันอยู่ แล้วนำไปคำนวณในอนุกรมของตัวเองตามกฎที่แตกต่างกัน

กฎการเคลื่อนที่:

- Cursor A จะเคลื่อนที่แบบซิกแซก โดยแถวที่เป็นเลขคู่ (แถวที่ 0, 2, 4, ...) จะเคลื่อนที่จาก ซ้ายไปขวา (หลัก $0 \rightarrow N-1$) และแถวที่เป็นเลขคี่ (แถวที่ 1, 3, 5, ...) จะเคลื่อนที่จาก ขวาไปซ้าย (หลัก $N-1 \rightarrow 0$)
- Cursor B จะเริ่มต้นที่ (0, 0) และในก้าวถัดไปจะถูกกำหนดโดย **คุณสมบัติ (เลขคู่/คี่)** ของตัวเลขในช่องที่มันอยู่ปัจจุบัน สมมติว่าปัจจุบัน Cursor B อยู่ที่ (r, c) และมีค่าในช่องนั้นเท่ากับ **value**:
 - ถ้า **value** เป็นเลขคู่ : ตำแหน่งใหม่จะคำนวณจากสูตร:
$$\text{newRow} = ((r + \text{value}) \% N + N) \% N, \text{newCol} = ((c + \text{value} + 1) \% N + N) \% N$$
 - ถ้า **value** เป็นเลขคี่ : ตำแหน่งใหม่จะคำนวณจากสูตรที่แตกต่างออกไป:
$$\text{newRow} = ((r + \text{value}) \% N + N) \% N, \text{newCol} = ((c - \text{value} - 1) \% N + N) \% N$$
- เงื่อนไขการหยุดเดิน:
 - การเคลื่อนที่จะดำเนินไปทั้งหมด $N \times N$ ก้าว โดยจะนับลำดับก้าวตั้งแต่ ก้าวที่ 0 ถึง ก้าวที่ $N \times N - 1$
 - Cursor A ซึ่งมีเส้นทางที่แน่นอนในการเคลื่อนที่ให้ครบทุกช่อง จะเป็นตัวกำหนดให้การเคลื่อนที่ทั้งหมดมี $N \times N$ ก้าว เมื่อครบจำนวนก้าวที่กำหนดแล้ว การเคลื่อนที่ของทั้ง Cursor A และ Cursor B จะสิ้นสุดลงพร้อมกัน

กฎการคำนวณ:

- ค่าตั้งต้น:** ผลรวมของอนุกรม A และ B เริ่มต้นที่ 0
- อนุกรม A (ตามลำดับก้าว):**
 - ในก้าวที่เป็น **ลำดับคู่** (ก้าวที่ 0, 2, 4, ...): นำค่าที่เคอร์เซอร์ A อยู่ **ลบ** ออกจากผลรวมของอนุกรม A
 - ในก้าวที่เป็น **ลำดับคี่** (ก้าวที่ 1, 3, 5, ...): นำค่าที่เคอร์เซอร์ A อยู่ **บวก** เข้ากับผลรวมของอนุกรม A
- อนุกรม B (ตามคุณสมบัติตัวเลข):**
 - ถ้าค่าในช่องที่ Cursor B อยู่เป็น **เลขคู่** : นำค่านั้น **บวก** เข้ากับผลรวมของอนุกรม B
 - ถ้าค่าในช่องที่ Cursor B อยู่เป็น **เลขคี่** : นำค่านั้น **ลบ** ออกจากผลรวมของอนุกรม B
- กฎพิเศษ: การทับซ้อน (Collision Rule):** ในแต่ละก้าว หากตำแหน่งของ Cursor A และ Cursor B เป็นตำแหน่งเดียวกัน ให้ถือว่า Cursor A ได้รับค่าในช่องนั้นไปคนเดียว ส่วนการคำนวณของอนุกรม B ในก้าวนั้น ให้ใช้ค่าเป็น 0 (เสมือนว่า Cursor B ไม่ได้เก็บค่าใดๆ)



A_0 และ B_0 หมายถึงเส้นทางการเดินของ Cursor A ก้าวที่ 0 (จุดเริ่มต้น) และ Cursor B ก้าวที่ 0 (จุดเริ่มต้น) ตามลำดับ โดย Cursor A จะเป็นตัวกำหนดให้การเคลื่อนที่ทั้งหมดมี $N \times N$ ก้าว เริ่มจาก A_0 ไล่ไปเรื่อยๆ จนถึง $A_{N \times N - 1}$ เช่นเดียวกันกับ Cursor B เริ่มจาก B_0 ไล่ไปเรื่อยๆ จนถึง $B_{N \times N - 1}$ เมื่อครบจำนวนก้าวที่กำหนดแล้ว การเคลื่อนที่ของทั้ง Cursor A และ Cursor B จะสิ้นสุดลงพร้อมกัน จากรูปทางซ้าย A มีจำนวนก้าวทั้งหมดคือ 9 ก้าว (A_0 ถึง A_8) เนื่องจากตารางกริดมีขนาด 3×3

ข้อมูลนำเข้า

- บรรทัดแรก: จำนวนเต็ม N (ขนาดของกริด, $1 \leq N \leq 10$)
- N บรรทัดถัดมา: แต่ละบรรทัดมีจำนวนเต็ม N ตัว คั่นด้วยช่องว่าง แทนค่าในแต่ละแถวของกริด
- บรรทัดสุดท้าย: รหัสคำสั่ง (1 หรือ 2)

ข้อมูลส่งออก

ผลลัพธ์จะขึ้นอยู่กับ รหัสคำสั่ง ที่รับเข้ามา:

- ถ้าคำสั่งคือ 1 (เปรียบเทียบค่าสุดท้ายของอนุกรม A และ B):
 - ถ้า $A > B$ ให้แสดงผล A ตามด้วยค่าผลรวมของอนุกรม A และค่าผลรวมของอนุกรม B คั่นแต่ละค่าด้วยช่องว่าง
 - ถ้า $B > A$ ให้แสดงผล B ตามด้วยค่าผลรวมของอนุกรม A และค่าผลรวมของอนุกรม B คั่นแต่ละค่าด้วยช่องว่าง
 - ถ้า $A == B$ ให้แสดงผล EQUAL ตามด้วยค่าผลรวมของอนุกรม A และค่าผลรวมของอนุกรม B คั่นแต่ละค่าด้วยช่องว่าง
- ถ้าคำสั่งคือ 2 (นับการทับซ้อน):
 - แสดงจำนวนครั้งที่ทั้งหมดที่ตำแหน่งของ Cursor A และ B ทับซ้อนกัน

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
2 1 3 4 2 2	3
3 1 4 7 2 5 8 3 6 9 2	1
3 2 5 1 6 3 8 4 7 9 1	B 7 10

<div>3</div> <div>1 100 1</div> <div>1 9 100</div> <div>1 1 1</div> <div>1</div>	<div>A 189 -15</div>
<div>4</div> <div>2 2 2 2</div> <div>1 1 1 2</div> <div>1 1 1 1</div> <div>1 1 1 1</div> <div>1</div>	<div>EQUAL -1 -1</div>