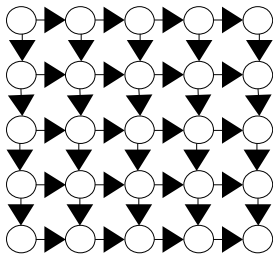


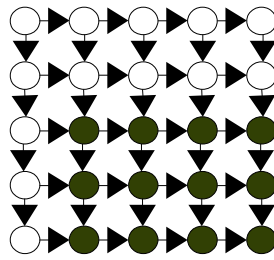
น้ำเสีย

Interactive task, 1second, 64MB

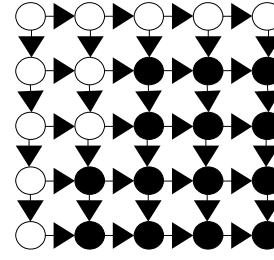
ระบบท่อน้ำของเมืองประกอบไปด้วยหัวต่อส่งน้ำจำนวน $N \times N$ ต่อกันเป็นตารางกริดขนาด N แถว N คอลัมน์ หัวต่อจะมีท่อน้ำแบบทางเดียวเชื่อมอยู่ โดยหัวต่อหัวที่อยู่ตำแหน่งแถวที่ i คอลัมน์ที่ j จะมีท่อน้ำแบบทางเดียวต่อไปยังหัวต่อที่อยู่ตำแหน่งแถวที่ i คอลัมน์ที่ $j+1$ และ ตำแหน่งแถวที่ $i+1$ และคอลัมน์ที่ j ถ้ามีหัวต่อเหล่านั้นอยู่ ด้านล่างในรูปที่ (a) เป็นตัวอย่างของระบบส่งน้ำที่มี 5×5 หัวต่อ



(a)



(b)



(c)

แหล่งน้ำเข้าจะเริ่มที่หัวต่อมุมบนซ้าย ที่หัวต่อใด ๆ น้ำจะไหลต่อไปในทิศทางของท่อน้ำ จนในที่สุดไปสิ้นสุดที่หัวต่อมุมล่างขวา

ในการใช้งานน้ำ บางครั้งผู้ใช้น้ำอาจจะต่อท่อน้ำทิ้งผิดพลาด ทำให้น้ำเสียไหลกลับเข้ามาในระบบส่งน้ำ น้ำเสียนั้นจะไหลไปตามท่อ จนกระทั่งถึงหัวต่อมุมล่างขวา รูปตัวอย่างที่ (b) แสดงหัวต่อที่มีน้ำเสีย ถ้ามีการปล่อยน้ำเสียเข้ามาจากหัวต่อที่อยู่แถวที่ 3 และคอลัมน์ที่ 2 ตัวอย่างที่ (c) แสดงกรณีที่มีการปล่อยน้ำเสียจากสองหัวต่อ ที่อยู่แถวที่ 2 คอลัมน์ 3 และแถวที่ 4 คอลัมน์ 2

เช้าวันนี้คุณวัดคุณภาพน้ำที่หัวต่อมุมล่างขวาแล้วพบว่า มีน้ำเสียอยู่ในหัวต่อ คุณต้องการสืบหาว่าผู้ใช้น้ำที่หัวต่อใดบ้างเป็นผู้เริ่มปล่อยน้ำเสียเข้ามาในระบบน้ำ จำนวนผู้ใช้น้ำที่ปล่อยน้ำเสียมีไม่เกิน K คน โดยที่รับประกันว่าทั้ง K คนจะไม่อยู่ในตำแหน่งที่น้ำเสียจะไหลถึงกันได้

ในการหาที่มานั้น คุณสามารถโทรศัพท์ไปให้เจ้าหน้าที่ที่หัวต่อต่าง ๆ วัดคุณภาพน้ำที่หัวต่อใด ๆ ก็ได้ และใช้ข้อมูลดังกล่าวเพื่อหาที่มาของต้นตอ น้ำเสีย อย่างไรก็ตาม ด้วยงบประมาณในการโทรศัพท์ที่จำกัด คุณจะวัดคุณภาพน้ำได้ไม่เกิน 200 ครั้งเท่านั้น

การปฏิสัมพันธ์กับระบบ

โปรแกรมที่คุณเขียนจะได้รับข้อมูลทดสอบหลายชุด โดยจะใช้ฟังก์ชันดังนี้

- `init_all()`
ฟังก์ชันนี้จะคืนจำนวนเต็ม T แทนจำนวนข้อมูลทดสอบซึ่งโปรแกรมของคุณจะต้องหาคำตอบ ($1 \leq T \leq 100$)

สำหรับแต่ละข้อมูลทดสอบ คุณจะใช้ฟังก์ชันเหล่านี้

- `init()`
ก่อนเริ่มแต่ละข้อมูลทดสอบ ให้เรียกฟังก์ชันนี้ ซึ่งจะคืนจำนวนเต็ม N แทนขนาดของระบบส่งน้ำ สำหรับแต่ละชุดข้อมูลทดสอบ คุณจะเรียกฟังก์ชัน `init` ได้ครั้งเดียว และต้องเรียกเป็นฟังก์ชันแรก
- `bool check(int r, int c)`

ฟังก์ชันนี้ใช้ในการตรวจสอบหัวต่อที่แถว r คอลัมน์ c โดยฟังก์ชันจะคืนค่า true ถ้าน้ำในหัวต่อนั้นเป็นน้ำเสีย คุณสามารถเรียกฟังก์ชัน check ได้ไม่เกิน 200 ครั้ง

- `report(int m, int rs[], int cs[])`
เมื่อคุณได้ตรวจสอบแล้ว ให้เรียกฟังก์ชัน report เพื่อรายงาน โดยให้ใส่พารามิเตอร์ดังนี้ m แทนจำนวนต้นตอที่มีการปล่อยน้ำเสีย ($1 \leq m \leq K$) จากนั้นในอาร์เรย์ rs และ cs จะเป็นแถวและคอลัมน์ของหัวต่อที่มีการปล่อยน้ำเสีย กล่าวคือ สำหรับ $0 \leq j \leq m-1$, หัวต่อที่มีการปล่อยน้ำเสียที่ j อยู่ในแถว `rs[j]` และคอลัมน์ `cs[j]`
ในการเรียกนั้น ลำดับของหัวต่อที่มีการปล่อยน้ำเสียจะเป็นอย่างไรก็ได้ สำหรับแต่ละชุดข้อมูลทดสอบคุณจะถูกเรียกฟังก์ชัน report ครั้งเดียว และเป็นฟังก์ชันสุดท้ายสำหรับข้อมูลทดสอบนั้น ๆ

ดาวน์โหลดไลบรารีได้ที่ <http://theory.cpe.ku.ac.th/~jittat/oi/2013/waste/>

ตัวอย่างการติดต่อกับไลบรารี

ฟังก์ชัน	ค่าที่คืน	คำอธิบาย
<code>init_all()</code>	2	มีสองข้อมูลทดสอบ
<code>init()</code>	5	ขนาด = 5 (ใช้ตัวอย่างในรูป (b))
<code>check(1,1)</code> <code>check(2,2)</code> <code>check(3,3)</code> <code>check(4,4)</code>	false false true true	
<code>report(1,{3},{2})</code>		รายงานตำแหน่งของข้อมูลทดสอบแรก
<code>init()</code>	5	ขนาด = 5 (ใช้ตัวอย่างในรูป (c))
<code>check(1,1)</code> <code>check(2,3)</code> <code>check(3,3)</code> <code>check(4,1)</code>	false true true false	
<code>report(2,{2,4},{3,2})</code>		รายงานตำแหน่งว่าอยู่ที่ 2,3 และ 4,2

ปัญหาย่อย

ปัญหาย่อย 1 (10%): $N \leq 10, K = 1$	ปัญหาย่อย 4 (20%): $N \leq 1\,000, K = 2$
ปัญหาย่อย 2 (10%): $N \leq 1\,000, K = 1$	ปัญหาย่อย 5 (20%): $N \leq 1\,000\,000, K = 2$
ปัญหาย่อย 3 (10%): $N \leq 1\,000\,000, K = 1$	ปัญหาย่อย 6 (30%): $N \leq 1\,000\,000, K = 3$