

ปัญหา ร้านอาหารที่อยู่ใกล้ ๆ (nearby_restaurant)

การหาร้านอาหาร ร้านอาหาร และศูนย์บริการที่อยู่ใกล้ ๆ ด้วยการใช้สมาร์ทโฟนหรือโปรแกรมเว็บเป็นสิ่งที่นิยมกันมากในปัจจุบัน และเราต้องการจะเขียนส่วนของการคำนวณแบบง่าย ๆ ในชุดทดสอบนี้

จากแผนที่ในรูปที่ 1 ตัวเมืองถูกแบ่งออกเป็นช่อง (block) อย่างเป็นระเบียบ และเลขในแต่ละช่องระบุพิกัดของช่องนั้น ๆ สมมติว่าผู้ใช้งานโปรแกรมอยู่ที่พิกัด (5, 4) ส่วนร้านอาหารมีทั้งหมด 12 ร้าน (พิกัดที่มีเครื่องหมายดอกจันอยู่ด้วย) ผู้ใช้สามารถเดินทางไปยังช่องที่ติดกับตนเองได้ในทิศแนวตั้งและแนวนอนเท่านั้น ไม่สามารถเดินทางในทิศทแยงมุมได้ เช่น การเดินทางของผู้ใช้ไปยังร้านอาหาร ณ พิกัด (6, 2) ผู้ใช้จะต้องเดินทางผ่านตัวเมืองทั้งหมด 3 ช่อง และถ้าจะไปร้าน (3, 6) ผู้ใช้จะต้องเดินทางทั้งหมด 4 ช่อง

(1, 1)	(1, 2)	(1, 3)**	(1, 4)**	(1, 5)	(1, 6)**	(1, 7)	(1, 8)	(1, 9)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)	(2, 8)	(2, 9)
(3, 1)	(3, 2)	(3, 3)	(3, 4)**	(3, 5)	(3, 6)**	(3, 7)	(3, 8)	(3, 9)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)	(4, 7)	(4, 8)	(4, 9)
(5, 1)	(5, 2)**	(5, 3)	(5, 4)	(5, 5)	(5, 6)	(5, 7)	(5, 8)	(5, 9)
(6, 1)	(6, 2)**	(6, 3)	(6, 4)	(6, 5)	(6, 6)	(6, 7)	(6, 8)	(6, 9)
(7, 1)	(7, 2)	(7, 3)	(7, 4)	(7, 5)**	(7, 6)	(7, 7)**	(7, 8)	(7, 9)
(8, 1)	(8, 2)	(8, 3)	(8, 4)	(8, 5)**	(8, 6)**	(8, 7)**	(8, 8)	(8, 9)
(9, 1)	(9, 2)	(9, 3)	(9, 4)	(9, 5)	(9, 6)	(9, 7)	(9, 8)	(9, 9)

รูปที่ 1 แผนที่แสดงตำแหน่งผู้ใช้และร้านอาหาร

ดังนั้นถ้าผู้ใช้งานต้องการหาร้านอาหารที่ระยะทางห่างจากตนเองไม่เกิน 3 ช่อง ก็จะมีร้านให้ผู้ใช้เลือกทั้งหมด 4 ร้านคือ ร้านที่อยู่พิกัด (3, 4), (5, 2), (6, 2) และ (7, 5) แต่ถ้าผู้ใช้กำหนดระยะห่างจากตนเป็นไม่เกิน 4 ช่องก็จะมีร้านให้เลือกทั้งหมด 7 ร้านคือ (1, 4), (3, 4), (3, 6), (5, 2), (6, 2), (7, 5) และ (8, 5)

จงเขียนโปรแกรมที่รับพิกัดของผู้ใช้, จำนวนช่องสูงสุดที่ผู้ใช้จะสามารถเดินทางได้, จำนวนร้านอาหารในแผนที่ และพิกัดของร้านอาหารทั้งหมด จากนั้นโปรแกรมจะนับจำนวนร้านทั้งหมดที่อยู่ห่างจากผู้ใช้ไม่เกินระยะที่กำหนดและพิมพ์จำนวนร้านออกมาเป็นคำตอบ

ข้อมูลเข้า

- บรรทัดแรกเป็นจำนวนเต็มสองตัวคั่นด้วยช่องว่าง ระบุพิกัดของผู้ใช้เป็นหมายเลขแถวและคอลัมน์ตามลำดับ
- บรรทัดที่สองเป็นเลขจำนวนเต็มบวก R ระบุระยะทางสูงสุดที่ผู้ใช้สามารถเดินทางได้ (หน่วยเป็นจำนวนช่อง)
- บรรทัดที่สามเป็นเลขจำนวนเต็มบวก N บอกจำนวนร้านอาหารทั้งหมดที่อยู่ในแผนที่ ($N \leq 10,000$)

4. บรรทัดที่ 4 ถึง $N + 3$ ระบัพิกัดของร้านอาหาร หนึ่งร้านต่อหนึ่งบรรทัด แต่ละบรรทัดมีเลขจำนวนเต็มสองตัวคั่นด้วยช่องว่าง เลขสองตัวนี้ระบัพิกัดของร้านอาหารเป็นหมายเลขแถวและคอลัมน์ตามลำดับ กำหนดให้หมายเลขพิกัดแต่ละตัว มีค่าเป็นบวกและไม่เกิน 5,000

ผลลัพธ์

มีบรรทัดเดียวเป็นเลขจำนวนเต็มหนึ่งตัวระบุจำนวนร้านอาหารทั้งหมดที่อยู่ห่างจากผู้ใช้ไม่เกินระยะ R
หมายเหตุ เป็นไปได้ที่ร้านอาหารมากกว่าหนึ่งร้านจะอยู่ในพิกัดเดียวกัน รวมทั้งอาจอยู่พิกัดเดียวกับผู้ใช้ก็ได้

ตัวอย่าง

ตัวอย่างที่หนึ่ง		ตัวอย่างที่สอง		ตัวอย่างที่สาม		ตัวอย่างที่สี่	
ข้อมูลเข้า	ผลลัพธ์	ข้อมูลเข้า	ผลลัพธ์	ข้อมูลเข้า	ผลลัพธ์	ข้อมูลเข้า	ผลลัพธ์
5 4	4	5 4	7	10 5	6	10 7	0
3		4		3		1	
12		12		11		11	
1 3		1 3		9 3		9 3	
1 4		1 4		9 5		9 5	
1 6		1 6		10 5		10 5	
3 4		3 4		5 10		5 10	
3 6		3 6		5 9		5 9	
5 2		5 2		9 5		9 5	
6 2		6 2		10 5		10 5	
7 5		7 5		10 10		10 10	
7 7		7 7		9 3		9 3	
8 5		8 5		3 9		3 9	
8 6		8 6		5 5		5 5	
8 7		8 7					

วิธีคิดที่เขียนมาเป็นโปรแกรมได้ไม่ยาก (วิธีนี้ไม่ใช่อาเรย์)

- อ่านค่าตำแหน่งพิกัดของผู้ใช้ไว้ในตัวแปร A และ B
- อ่านค่าระยะทางที่เดินทางได้ไว้ในตัวแปร R
- อ่านค่าจำนวนร้านไว้ในตัวแปร N
- มีตัวนับจำนวนร้านอาหารไว้ อย่าลืมหักกำหนดค่าให้เป็น 0 ก่อนด้วย
- วนลูปรับค่าพิกัดของร้านเข้าเก็บไว้ในตัวแปร C และ D เวลาอ่านค่าของร้านแรกเสร็จ อย่าเพิ่งไปปรับวนลูปไปอ่านค่าของร้านถัดไป แต่ให้เอาค่า A, B, C และ D มาทำการคำนวณหาระยะทางก่อน ถ้าระยะทางที่คำนวณได้มันไม่เกิน R ก็ให้บวกหนึ่ง แต่ถ้าเกินก็ไม่ต้องบวก วิธีคำนวณสามารถทำได้โดย
 - คำนวณระยะทางด้วย $K = |C - A| + |D - B|$ (คำนวณความแตกต่างด้วยค่าสัมบูรณ์)
 - ตรวจสอบว่ามันเกินระยะ R หรือไม่
 - ถ้าไม่เกินก็บวกตัวนับเข้าไป
 - วนลูปไปรับค่าพิกัดของร้านอาหารถัดไป