International Olympiad in Informatics 2012
23-30 September 2012
Sirmione - Montichiari, Italy
Competition tasks, day 1: Leonardo's inventions and projects

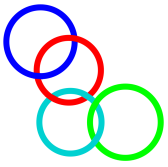
rings
ไทย — 1.2

วงแหวนในร่มชูชีพ

ในหนังสือชื่อดัง Codex Atlanticus (ตั้งแต่ปีค.ศ. 1485) ของลีโอนาร์โด มีการอธิบายถึงวิธีการสร้างร่มชูชีพไว้ โดยร่มชูชีพของลีโอนาร์โดนี้ใช้ผ้าลินินนำมาต่อกัน แล้วใช้ไม้รูปปิระมิดมาถ่างไว้เพื่อให้ผ้ามีช่องว่าง

วงแหวนที่เกี่ยวกัน

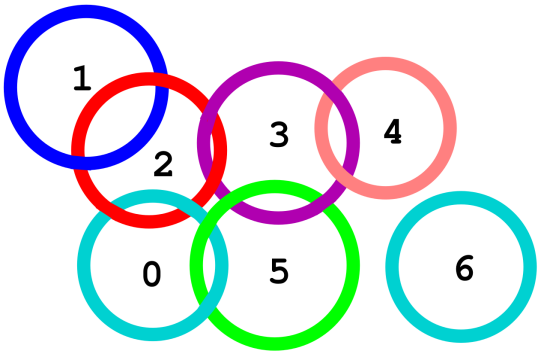
หลังจากที่ลีโอนาร์โดได้ออกแบบร่มชูชีพมากว่า 500 ปี นักดิ่งพสุธา แอเดรียน นิโคลัส ได้ทดสอบสิ่งที่ลีโอนาร์โดได้ออกแบบไว้ โดยใช้อุปกรณ์น้ำหนักเบาผูกร่มของลีโอนาร์โดเข้ากับตัวคน เราจะใช้วงแหวนที่เกี่ยวกันนี้คล้องเข้ากับผ้าลินิน วงแหวนแต่ละอันทำมาจากวัสดุที่ยืดหยุ่นและแข็งแรง วงแหวนเหล่านี้สามารถเกี่ยวเข้ากับวงแหวนอันอื่นโดยใช้การอ้าวงแหวนออก และบีบวงแหวนกลับ เพื่อให้กลายเป็นวงแหวนเหมือนเดิม การนำวงแหวนมาเกี่ยวกันทำได้หลายแบบ การเกี่ยวกันแบบพิเศษแบบหนึ่ง คือ การนำมาเกี่ยวกันเป็นสายโซ่ ซึ่งเป็นลำดับของวงแหวนที่วงแหวนแต่ละอันจะเชื่อมกันกับวงแหวนอันข้าง ๆ (อย่างมากไม่เกินสองอัน) ดังแสดงด้านล่าง ลำดับของวงแหวนนี้จะต้องมีจุดเริ่มต้นและจุดจบ (ซึ่งก็คือวงแหวนที่เชื่อมกับวงแหวนอื่นไม่เกิน 1 อัน) นอกจากนี้ วงแหวนเดี่ยว ก็ถือเป็นสายโซ่อันหนึ่งด้วยเช่นกัน



เนื่องจากวงแหวนหนึ่งอันสามารถเกี่ยวกับวงแหวนอื่น ๆ ได้สามอันหรือมากกว่านั้น ทำให้เราสามารถนำวงแหวนมาเกี่ยวกันเป็นแบบอื่น ๆ ได้อีก นอกจากนี้วงแหวนใด ๆ จะมีคุณสมบัติที่เรียกว่า วงแหวนสำคัญ ถ้าเราดึงเอาวงแหวนอันนั้นออกไปแล้ว วงแหวนอื่น ๆ จะกลายเป็นเซตของเป็นสายโซ่หลาย ๆ อัน (หรือไม่ก็เมื่อดึงออกแล้ว ไม่เหลือวงแหวนอีก) กล่าวอีกอย่างหนึ่งได้ว่า เมื่อดึงออกแล้ว จะเหลือแต่เพียงสายโซ่เท่านั้น

ตัวอย่าง

พิจารณาวงแหวน 7 อันในรูปถัดไป วงแหวนทั้งเจ็ดอันมีหมายเลข 0 ถึง 6 กำกับอยู่ ในรูปมีวงแหวนที่เป็นวงแหวนสำคัญอยู่สองอัน อันหนึ่ง คือ วงแหวนหมายเลข 2 เพราะหลังจากดึงวงแหวนหมายเลข 2 ออกไปแล้ว วงแหวนที่เหลือจะกลายเป็นสายโซ่ [1], [0, 5, 3, 4] และ [6] วงแหวนสำคัญอีกอันหนึ่ง คือ วงแหวนหมายเลข 3 เพราะเมื่อดึงวงแหวนหมายเลข 3 ออกไปแล้ว วงแหวนที่เหลือจะกลายเป็นสายโซ่ [1, 2, 0, 5], [4] และ [6] แต่ถ้าเราดึงวงแหวนอื่น ๆ ออก เราจะไม่ได้สายโซ่ที่แยกจากกันในแบบอื่น ๆ อีก ตัวอย่างเช่น ถ้าเราดึงวงแหวนหมายเลข 5 ออกไป แม้ว่าเรามีสายโซ่ [6] อยู่ แต่วงแหวนหมายเลข 0, 1, 2, 3 และ 4 ไม่ได้เกี่ยวกันเป็นสายโซ่



ปัญหา

งานของคุณ คือ นับจำนวนวงแหวนสำคัญในรูปแบบการเกี่ยวกันของวงแหวนที่จะถูกส่งไปให้โปรแกรมของคุณ

ในตอนเริ่มต้น มีวงแหวนอยู่จำนวนหนึ่งซึ่งอยู่แยกกัน หลังจากนั้นมีการนำวงแหวนมาเกี่ยวกัน ในเวลาที่กำหนดให้ใด ๆ คุณอาจจะถูกร้องขอให้ส่งคืนจำนวนวงแหวนสำคัญที่ปรากฏอยู่ในการเกี่ยวกันตามแบบที่เป็น ณ เวลานั้น ในการนี้คุณจะต้องเขียนโปรแกรมย่อยสามอัน

- `Init`(N) — Init จะถูกเรียกเพียงครั้งเดียวในตอนแรก เพื่อบอกจำนวนวงแหวน N อันที่อยู่แยกกัน วงแหวนแต่ละวงจะมีหมายเลข 0 ถึง N - 1 (รวมหัวท้าย) กำกับอยู่ ซึ่งวงแหวนที่อยู่แยกกันนี้จัดเป็นรูปแบบเริ่มต้น
- `Link`(A, B) — ให้นำวงแหวนหมายเลข A และ B มาเกี่ยวกัน รับประกันว่า A และ B แตกต่างกัน และยังไม่ได้เกี่ยวกันโดยตรง นอกเหนือจากนี้ไม่มีเงื่อนไขอื่น ๆ สำหรับ A และ B อีก โดยเฉพาะอย่างยิ่งจะไม่มีเงื่อนไขอื่นใดเกิดขึ้นจากข้อจำกัดทางด้านกายภาพ นอกจากนี้แล้ว ทั้ง `Link`(A, B) และ `Link`(B, A) เหมือนกัน
- `CountCritical`() — ส่งคืนจำนวนวงแหวนสำคัญที่ปรากฏในการเกี่ยวกันตามแบบที่เป็นอยู่ในปัจจุบัน

ตัวอย่าง

พิจารณารูปที่มี N = 7 และเริ่มต้นด้วยการที่ยังไม่มีวงแหวนคู่ไหนเกี่ยวกันเลย เราจะแสดงลำดับที่เป็นไปได้ของการเรียกโปรแกรมย่อย โดยหลังการเรียกโปรแกรมย่อยครั้ง เราจะได้สถานการณ์ดังแสดงในรูป

คำสั่ง	ค่าที่คืนกลับมา
`Init`(7)	
`CountCritical`()	7
`Link`(1, 2)	
`CountCritical`()	7
`Link`(0, 5)	
`CountCritical`()	7
`Link`(2, 0)	
`CountCritical`()	7
`Link`(3, 2)	
`CountCritical`()	4
`Link`(3, 5)	
`CountCritical`()	3
`Link`(4, 3)	
`CountCritical`()	2

งานย่อยที่ 1 [20 คะแนน]

- $N \leq 5,000$
- **ฟังก์ชัน** `CountCritical` **จะถูกเรียกเพียงครั้งเดียวหลังจากการเรียกใช้ฟังก์ชันอื่น ๆ ทั้งหมด และฟังก์ชัน** `Link` **ถูกเรียกไม่เกิน** 5,000 **ครั้ง**

งานย่อยที่ 2 [17 คะแนน]

- $N \leq 1,000,000$.
- **ฟังก์ชัน** `CountCritical` **จะถูกเรียกเพียงครั้งเดียวหลังจากการเรียกใช้ฟังก์ชันอื่น ๆ ทั้งหมด และฟังก์ชัน** `Link` **ถูกเรียกไม่เกิน** 1,000,000 **ครั้ง**

งานย่อยที่ 3 [18 คะแนน]

- $N \leq 20,000$
- **ฟังก์ชัน** `CountCritical` **ถูกเรียกไม่เกิน** 100 **ครั้ง และฟังก์ชัน** `Link` **ถูกเรียกไม่เกิน** 10,000 **ครั้ง**

งานย่อยที่ 4 [14 คะแนน]

- $N \leq 100,000$
- **ฟังก์ชัน** `CountCritical` **และฟังก์ชัน** `Link` **ถูกเรียกรวมกันไม่เกิน** 100,000 **ครั้ง**

งานย่อยที่ 5 [31 คะแนน]

- $N \leq 1,000,000$.
- **ฟังก์ชัน** `CountCritical` **และฟังก์ชัน** `Link` **ถูกเรียกรวมกันไม่เกิน** 1,000,000 **ครั้ง**

รายละเอียดสำหรับการเขียนโปรแกรม

คุณจะต้องส่งไฟล์หนึ่งไฟล์ในชื่อ `rings.c, rings.cpp` **หรือ** `rings.pas` **ไฟล์นี้จะต้องเขียนโปรแกรมย่อยดังที่กล่าวไว้ข้างต้นโดยใช้รูปแบบดังต่อไปนี้**

โปรแกรมภาษา C/C++

```
void Init(int N);
void Link(int A, int B);
int CountCritical();
```

โปรแกรมภาษา Pascal

```
procedure Init(N : LongInt);
procedure Link(A, B : LongInt);
function CountCritical() : LongInt;
```

โปรแกรมย่อยเหล่านี้จะต้องทำงานตามที่ได้ระบุไว้ข้างต้น คุณสามารถเขียนโปรแกรมย่อยอื่น ๆ สำหรับใช้งานได้ โปรแกรมของคุณจะต้องไม่ยุ่งเกี่ยวกับ standard input/output หรือกับไฟล์ใด ๆ

grader ตัวอย่าง

grader ตัวอย่าง จะอ่านอินพุตในรูปแบบดังต่อไปนี้

- บรรทัดที่ 1: N, L;
- บรรทัดที่ 2, …, L + 1:
  - -1 เพื่อเรียกใช้ `CountCritical`;
  - A, B เป็นพารามิเตอร์สำหรับ `Link`.

grader ตัวอย่างจะพิมพ์ผลลัพธ์ทั้งหมดจาก `CountCritical`