

ข้อสอบทักษะการเขียนโปรแกรม

Associated Programming Contest 2008

วันเสาร์ที่ 16 สิงหาคม 2551

20.00 – 23.00 น. (3 ชั่วโมง)

ข้อกำหนดของการสอบ

- ห้ามสื่อสารกับบุคคลอื่นในหัวข้อเกี่ยวกับการสอบ ไม่ว่าจะเป็นการติดต่อสื่อสารด้วยวิธีใดก็ตาม
- ห้ามปรึกษาหรือแลกเปลี่ยนตัวอย่างชุดข้อมูลทดสอบให้กับบุคคลอื่น ไม่ว่ากรณีใดๆ
- สามารถนำส่วนของโปรแกรมที่เคยเขียนไว้แล้วมาใช้ได้ ตราบเท่าที่คุณเป็นเจ้าของส่วนของโปรแกรมนั้น
- กำหนดให้ตัวแปรชนิด integer บนระบบตรวจ มีขนาด 32 bits
- การสอบมีระยะเวลา 3 ชั่วโมง และระบบจะปิดรับโปรแกรมใดๆ โดยอัตโนมัติเมื่อหมดเวลาสอบ
- เมื่อผู้เข้าทดสอบส่งโปรแกรมในระหว่าง 3 ชั่วโมงของการทดสอบ ระบบจะทำการคอมไพล์และทดสอบโปรแกรมที่ส่งขึ้นมากับชุดข้อมูลทดสอบตัวอย่างจำนวน 1 ถึง 2 ชุด ซึ่งชุดข้อมูลเหล่านี้ไม่ใช่ชุดข้อมูลทดสอบจริงที่จะทำการตรวจให้คะแนนแต่อย่างใด
- โจทย์อาจมีการใช้ภาษาที่ไม่เป็นทางการ ทั้งนี้เพื่ออรรถรสของเนื้อหาโจทย์ ต้องขออภัยมา ณ ที่นี้ด้วย
- หากมีคำถามเกี่ยวกับระบบหรือโจทย์ สามารถสอบถามผ่านทางแบบฟอร์มที่ทางระบบได้จัดไว้ให้

ข้อกำหนดของโจทย์

โจทย์	ดนตรีกวน (melody)	สวนดอกไม้ (harvest)	ระบบใหม่ (apcos)	โกโก้พลอย (garnet)
ข้อมูลนำเข้า	standard input (keyboard)			
ข้อมูลส่งออก	standard output (monitor)			
หน่วยความจำที่ใช้สูงสุด	16 MB			
เวลาที่ใช้สูงสุด	1 วินาที			
จำนวนชุดข้อมูลทดสอบ	25	10	20	20
คะแนนต่อชุดทดสอบ	4	10	5	5
คะแนนเต็ม	100	100	100	100

ดนตรีกวน (melody)

นิยาม โน้ต ๑, โน้ตเพลง น. เครื่องหมายกำหนดเสียงดนตรี. (อ. note).

อ้างอิงจาก พจนานุกรม ฉบับราชบัณฑิตยสถาน พ.ศ. 2542

โน้ตดนตรี มีอยู่ด้วยกันทั้งหมด 12 ตัวโน้ต โดยเรียงลำดับตามความถี่เสียงดังนี้

C C# D D# E F F# G G# A A# B

โดยในแต่ละความถี่เสียงด้านบนนั้น จะห่างกัน 1 semitone (ครึ่งเสียง) กล่าวคือ จาก C ไปยัง C# จะเป็นการเพิ่มความถี่เสียงไปอีก 1 semitone หรือ จาก C ไปยัง D จะเป็นการเพิ่มความถี่เสียงไปอีก 2 semitones ทั้งนี้สังเกตว่าจาก E ไป F นั้น เป็นการเพิ่มความถี่เสียงเพียง 1 semitone ไม่ใช่ 2 semitones นอกจากนั้นเรายังพอจะทราบได้อีกว่าเครื่องหมาย # (sharp) นั้นอาจหมายถึงการเพิ่มความถี่เสียงขึ้นอีก 1 semitone นั้นเอง

โดยเมื่อเรียงครบ 12 ชุดตัวโน้ตด้านบนแล้ว หลังจากเสียงตัวโน้ต B จะวนกลับไปเป็นตัวโน้ต C อีกครั้ง ซึ่งความถี่เสียงจะห่างกัน 1 semitone แต่ว่าเสียง C นี้จะสูงขึ้นไปอีก 1 octave (คู่แปด)

นอกเหนือจากการใช้เครื่องหมาย sharp เพื่อเป็นการเพิ่มเสียงไป 1 semitone แล้ว ยังมีการใช้อีกเครื่องหมายหนึ่งซึ่งทำหน้าที่ตรงกันข้าม นั่นคือเครื่องหมาย b (flat) เพื่อลดเสียงลง 1 semitone ซึ่งเราอาจจะเขียนลำดับของโน้ตข้างต้นใหม่ ในรูปแบบของเครื่องหมาย flat แทนที่จะเป็น sharp ได้ดังนี้

C Db D Eb E F Gb G Ab A Bb B

และจะวนกลับมาที่ C อีกครั้งหนึ่ง โดยที่ B กับ C จะมีความถี่เสียงต่างกัน 1 semitone แต่เสียงจะสูงขึ้นไปอีก 1 octave เช่นเดียวกับในรูปแบบแรกที่เราได้กล่าวถึงไปแล้ว

เนื่องจากเราเชื่อว่าผู้เข้าทดสอบ APC 2008 เป็นผู้ที่หลงใหลในเสียงดนตรีเป็นอย่างมาก เมื่อมีเสียงดนตรีใดๆ มากระทบโสตประสาท ก็อาจจะจินตนาการเห็นภาพสายของโน้ตดนตรีวาดเป็นเส้นสวยสดงดงาม เราจึงอยากให้คุณช่วยเขียนโปรแกรมเพื่อพิจารณาสายรหัสดนตรีที่กำหนดให้ แล้วบอกได้ว่าเสียงที่มีการเล่นซ้ำมากที่สุดนั้นมีจำนวนการเล่นซ้ำทั้งหมดกี่ครั้ง โดยที่สายรหัสดนตรีที่กำหนดให้นั้น อาจประกอบด้วยชุดของตัวโน้ตที่ดูประหลาดไปจากโน้ตดนตรีทั่วไปที่คุณเคยเห็นมา

ตัวอย่างเช่น หากเขียนว่า

Db จะมีค่าเท่ากับโน้ต C# หรือ Db ก็ได้

C## จะมีค่าเท่ากับโน้ต D

G##### จะมีค่าเท่ากับโน้ต C# หรือ Db ก็ได้

A##b จะมีค่าเท่ากับโน้ต A# หรือ Bb ก็ได้

Cbbbb# จะมีค่าเท่ากับโน้ต A

B# จะมีค่าเท่ากับโน้ต C (สังเกตว่าสูงขึ้น 1 octave แต่ยังถือว่าเป็นโน้ต C เหมือนกัน)

สมมติว่าคุณได้รับสายรหัสดนตรีเป็น CDEFGABCC#b###bD#####F###G#b#C##b###A##bbb

โปรแกรมของคุณจะต้องตอบได้ว่าโน้ตที่มีการเล่นซ้ำมากที่สุด มีการเล่นซ้ำทั้งหมด 3 ครั้ง (หมายถึงโน้ต A)

รับประกันได้ว่าทุกชุดข้อมูลทดสอบนั้น สายรหัสดนตรีอยู่ในรูปแบบที่ถูกต้องและหาคำตอบได้เสมอ

ข้อมูลนำเข้า

บรรทัดเดียว กำหนดสายรหัสดนตรีมา รับประกันว่าความยาวรวมไม่เกิน 1,000 อักขระ และไม่มีช่องว่าง
 ตัวโน้ตจะเป็นตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่
 ใช้ # แทนเครื่องหมาย sharp และใช้ b (บีเล็ก) แทนเครื่องหมาย flat

ข้อมูลส่งออก

บรรทัดเดียว พิมพ์ว่าตัวโน้ตที่มีการเล่นซ้ำมากที่สุด มีจำนวนการเล่นซ้ำกี่ครั้ง
 อาจมีตัวโน้ตมากกว่า 1 ตัวที่มีจำนวนครั้งการเล่นซ้ำมากที่สุดเท่ากัน

ตัวอย่างข้อมูลนำเข้า

CDEFGABCC#b###bD#####F####G#b##C##b###A##bbb

ตัวอย่างข้อมูลส่งออก

3

คำอธิบายตัวอย่างข้อมูล

จากสายรหัสดนตรีที่กำหนดมานั้น จะพบว่าสามารถแบ่งกลุ่มของตัวโน้ตที่ถูกเล่นได้ดังนี้

C D E F G A B C C#b###b D##### F#### G#b## C##b### A##bbb

ซึ่งมีค่าเท่ากับ

C D E F G A B C D G A A E G#

จะพบว่าตัวโน้ตที่ถูกเล่นซ้ำมากที่สุดคือตัวโน้ต A ซึ่งถูกเล่นทั้งหมด 3 ครั้งด้วยกัน

สวนดอกไม้ (harvest)

กาลครั้งหนึ่งนานมาแล้ว ณ หมู่บ้าน Mineral มีเด็กหนุ่มบ้านไร่ไฟแรงผู้ชอบเก็บเห็ดมาทำเกาเหลาเป็นชีวิตจิตใจ ได้เอาบัวรดน้ำไปอัพเกรดด้วยแร่มีสทริลที่ร้านอัพเกรดประจำเมือง ซึ่งบัวรดน้ำที่ได้ถูกอัพเกรดมาแล้วนี้ มีทั้งข้อดีและข้อเสียอยู่ด้วยกัน กล่าวคือ ผู้ที่ใช้บัวรดน้ำนี้ สามารถทำการรดน้ำได้ทุกทิศทางรอบจุดที่ผู้ใช้ยืนอยู่ ไม่รวมพื้นที่ที่ผู้ใช้ยืนอยู่ (รวมเป็นจำนวน 8 ช่อง) แต่เนื่องจากแร่มีสทริลนั้น เป็นแร่ที่มีข้อเสียคือจะทำให้เกิดปฏิกิริยา Mystrilization กับน้ำและทำให้เกิดการสุกก่อนได้ ดังนั้นบัวรดน้ำที่อัพเกรดมานี้ ไม่ควรใส่น้ำทิ้งไว้ข้ามวัน มิเช่นนั้นแล้ว บัวรดน้ำที่เพิ่งอัพเกรดมาใหม่ๆ อาจจะสุกก่อนและเสียหายได้ และหนุ่มน้อยจะเสียใจเป็นอย่างยิ่ง

น้ำที่ใช้ในหมู่บ้าน Mineral นี้ มีคุณสมบัติที่แตกต่างจากน้ำที่เราใช้กันอยู่ปัจจุบัน นั่นคือ หากน้ำโดนแร่มีสทริลแล้ว เมื่อบัวรดน้ำนั้นไปยังดอกไม้ที่เหี่ยวเฉา มันสามารถทำให้ดอกไม้เหล่านั้นกลับมากระปรี้กระเปร่าได้อีกครั้ง ในขณะเดียวกัน หากรดน้ำพลาดไปโดนดอกไม้ที่สดแสนจะกระปรี้กระเปร่าอยู่แล้ว มันก็จะกลับกลายเป็นดอกไม้ที่เหี่ยวเฉาในทันที

เช้าวันถัดมา หนุ่มไฟแรงคนนั้นได้นำบัวรดน้ำให้เต็มเหมือนที่เคยทำเป็นกิจวัตรประจำวัน แต่ทว่าเขาลืมตัวไปว่าบัวรดน้ำที่สดแสนจะวิเศษนี้เพิ่งถูกอัพเกรดด้วยแร่มีสทริลนั้นแตกต่างไปจากเมื่อก่อนแล้ว เขายังต้องการดอกไม้ที่กระปรี้กระเปร่าให้มากที่สุดในแต่ละวันเพื่อนำไปขายทำกำไรให้ได้มากๆ จะได้นำเงินทุนไปก่อตั้งสมาคมนักเก็บเห็ดมาทำเกาเหลาประจำเมือง เขาจึงได้มาขอร้องให้โปรแกรมเมอร์ประจำเมือง (ซึ่งก็คือคุณนั่นแหละ) ได้ช่วยเขียนโปรแกรมเพื่อช่วยเขาคิดให้น้อย ว่าหากเขาเลือกลำดับของจุดสำหรับยืนรดน้ำได้อย่างไรที่สุดจากสวนดอกไม้ขนาด 9x9 ช่องแล้ว เขาสามารถจะทำให้ดอกไม้กระปรี้กระเปร่ามากที่สุดก็ดอก

โดยมีข้อกำหนดว่า

- เมื่อบัวรดน้ำไปหนึ่งครั้ง น้ำที่อยู่ในบัวจะลดลงไป 1 ใน 3 ของปริมาณบัวรดน้ำ
- เขาจำเป็นต้องรดน้ำจนหมดบัว เนื่องจากไม่ต้องการให้บัวสุกก่อนจากปฏิกิริยาที่ได้กล่าวถึง
- เนื่องจากว่าสวนดอกไม้อยู่ห่างจากบ่อน้ำมาก เขาจึงไม่สามารถเติมน้ำได้อีก
- การรดน้ำที่ดี ต้องรดน้ำโดยคำนึงถึงการประหยัดน้ำเพื่อลดสถานะโลกร้อน น้ำที่รดจะต้องโดนดอกไม้ครบทั้ง 8 ช่องรอบทิศของบัวรดน้ำ และห้ามเทน้ำจากบัวทิ้งไปเปล่าๆ
- เมื่อยืนรดน้ำที่จุดใดไปแล้ว ไม่สามารถกลับมายืนซ้ำที่จุดนั้นได้อีก

ข้อมูลนำเข้า

มีจำนวนทั้งหมด 9 บรรทัด แต่ละบรรทัดประกอบด้วยตัวเลข 0 หรือ 1 จำนวน 9 ตัว กันด้วยช่องว่าง โดยที่ เลข 0 หมายถึงดอกไม้ที่เหี่ยวเฉา และเลข 1 หมายถึงดอกไม้ที่กระปรี้กระเปร่า

ข้อมูลส่งออก

บรรทัดเดียว พิมพ์จำนวนดอกไม้ที่กระปรี้กระเปร่าที่มากที่สุดที่สามารถทำได้ จากเงื่อนไขข้างต้น

ตัวอย่างข้อมูลนำเข้า

```

0 0 0 1 1 0 0 1 1
0 1 0 1 1 0 1 1 1
0 0 0 1 1 0 1 1 1
1 1 1 1 1 1 1 1 1
1 1 0 0 0 1 1 1 1
1 1 0 1 0 1 0 1 1
1 1 0 0 0 1 1 1 1
1 1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 0 1

```

ตัวอย่างข้อมูลส่งออก

```

73

```

คำอธิบายตัวอย่างข้อมูล

การเลือกที่ดีที่สุดในข้อตัวอย่างข้อมูลนี้ คือเลือกยื่นรคน้ำที่จุด (2, 2), (7, 2) และ (4, 6) ตามลำดับ จะได้

```

      1 2 3 4 5 6 7 8 9
      -----
1 | 1 1 1 1 1 1 1 0 1
2 | 1 1 1 1 1 1 1 0 1
3 | 1 1 1 1 1 1 0 0 1
4 | 1 1 1 1 1 1 1 1 1
5 | 1 1 1 1 1 1 1 1 1
6 | 1 1 1 1 1 1 0 1 1
7 | 1 1 1 1 1 1 1 1 1
8 | 1 1 1 1 1 1 1 0 0
9 | 1 1 1 1 1 1 1 0 1

```

จะได้จำนวนดอกไม้ที่กระปรี่กระเปร่าเป็นจำนวน 73 ดอกด้วยกัน

ระบบใหม่ (apcos)

นอกเหนือจากระบบตรวจออนไลน์ของ APC 2008 ที่ถูกเขียนขึ้นมาใหม่แล้วนั้น ทางคณะกรรมการได้เขียนระบบปฏิบัติการ APCos มาทดลองใช้ ด้วยเหตุผลที่ว่าเมื่อระบบปฏิบัติการยอดนิยมที่มีใช้กันอยู่ทั่วไปในปัจจุบันนี้ แต่ด้วยเหตุที่คณะกรรมการทุกคนล้วนมีภาระหน้าที่ที่ต้องรับผิดชอบมากมายนัก จึงทำให้แต่ละคนต้องเปิดหน้าต่างโปรแกรมขึ้นมาหลายตัว จนยุ่งกันไปหมด ทั้งหน้าต่างแชทกับสาว ๆ หน้าต่าง Hi5 หน้าต่างเกม GTA และอีกสารพัดหน้าต่างจนเต็มหน้าจอ

ที่เป็นปัญหาไปมากกว่านั้นคือคณะกรรมการแต่ละคนนั้นมีจอชน สลับสับเปลี่ยนหน้าต่างอยู่ตลอดเวลา หรือบ้างก็ปิดหน้าต่างบางอันไปเลย จนทำให้ระบบปฏิบัติการเกิดอาการรวนขึ้น เพราะไม่รู้ว่าจะต้องนำหน้าต่างไหนขึ้นมาแสดงด้านบนสุด จึงเป็นปัญหาของคุณซึ่งเป็นผู้เข้าทดสอบ APC 2008 ที่จะต้องช่วยแก้ไขปัญหามาของระบบปฏิบัติการ APCos โดยข้อมูลที่ทำให้มีดังนี้

เริ่มแรกจากไม่มีหน้าต่างใดๆ เปิดหน้าต่างที่ 1 ถึง N เรียงตามลำดับ โดยที่หน้าต่างที่เปิดทีหลังจะขึ้นมามีทับหน้าต่างที่เปิดก่อน ดังนั้นหลังจากเปิดเสร็จแล้ว หน้าต่างที่ N จะอยู่บนสุด และหน้าต่างที่ 1 จะอยู่ล่างสุด ซึ่งหน้าต่างแต่ละหน้าต่างจะมีตำแหน่งและขนาด โดยบอกด้วยพิกัดมุมบนซ้าย และพิกัดมุมล่างขวา และมีปุ่มปิดอยู่ที่มุมบนซ้ายสุด (ตำแหน่งเดียวกันกับพิกัดมุมบนซ้ายของหน้าต่าง)

หลังจากนั้นทำการคลิกเมาส์ T ครั้ง ในแต่ละครั้งหากคลิกโดนหน้าต่างใด หน้าต่างนั้นจะถูกนำขึ้นมาแสดงด้านบนสุดทับหน้าต่างอื่นๆ โดยการคลิกโดนขอบหน้าต่างก็ถือเป็นการคลิกโดนหน้าต่างนั้นๆ ด้วย แต่ถ้าคลิกโดนปุ่มปิด (พิกัดมุมบนซ้าย) จะเป็นการปิดหน้าต่างนั้นแทน

กรณีคลิกตำแหน่งที่มีหน้าต่างมากกว่า 1 หน้าต่างซ้อนทับกัน ให้ถือว่าหน้าต่างที่อยู่บนกว่าจะบังทับหน้าต่างที่อยู่ล่าง ดังนั้นการคลิกนั้น จะโดนหน้าต่างที่อยู่บนสุด ณ ตำแหน่งพิกัดนั้นๆ เพียงหน้าต่างเดียว

รับประกันได้ว่าพิกัดของทุกหน้าต่างที่เปิด และพิกัดตำแหน่งการคลิกเมาส์เป็นจำนวนเต็มเสมอ

ให้เขียนโปรแกรมหาว่าหน้าต่างที่เท่าไรอยู่บนสุด หลังจากคลิกเมาส์ T ครั้งแล้ว

ข้อมูลนำเข้า

บรรทัดแรก จำนวนเต็ม N แทนจำนวนหน้าต่างทั้งหมดที่ถูกเปิดขึ้นมาในตอนแรก

N บรรทัดถัดมา ตำแหน่งของหน้าต่างที่ 1 ถึงหน้าต่างที่ N ตามลำดับ (บอกเป็นจำนวนเต็ม)

ในแต่ละบรรทัดระบุพิกัดของแต่ละหน้าต่าง $X1\ Y1\ X2\ Y2$ มีความหมายดังนี้

$(X1, Y1)$ คือพิกัดมุมบนซ้ายของหน้าต่าง และ

$(X2, Y2)$ คือพิกัดมุมล่างขวาของหน้าต่าง

บรรทัดที่ $N+2$ จำนวนเต็ม T หมายถึงจำนวนครั้งของการคลิกเมาส์

T บรรทัดถัดมา พิกัดตำแหน่งของการคลิกเมาส์ครั้งที่ 1 ถึงครั้งที่ T (บอกเป็นจำนวนเต็ม)

ในแต่ละบรรทัดระบุพิกัดของการคลิกเมาส์แต่ละครั้ง **(X, Y)**

ข้อมูลส่งออก

บรรทัดเดียว พิมพ์หมายเลขหน้าต่างที่อยู่บนสุด หลังจากการคลิกเมาส์ **T** ครั้ง
แต่หากไม่มีหน้าต่างใดๆ เหลืออยู่เลย ให้พิมพ์ว่า "DESKTOP" (ไม่ต้องมีเครื่องหมายคำพูด)

ช่วงของข้อมูล

- $0 \leq N \leq 1,000$
- $0 \leq X1 < X2 \leq 128$
- $0 \leq Y1 < Y2 \leq 96$
- $0 \leq X \leq 128$
- $0 \leq Y \leq 96$
- $0 \leq T \leq 100,000$

ตัวอย่างข้อมูลนำเข้า (ชุดที่ 1)

```
2
0 0 60 60
50 50 120 90
5
50 51
25 25
60 60
61 30
0 0
```

ตัวอย่างข้อมูลส่งออก (ชุดที่ 1)

```
2
```

ตัวอย่างข้อมูลนำเข้า (ชุดที่ 2)

```
2
21 21 40 40
10 10 20 20
4
20 20
21 20
21 21
10 10
```

ตัวอย่างข้อมูลส่งออก (ชุดที่ 2)

```
DESKTOP
```

หมายเหตุ

30% ของชุดข้อมูลทดสอบ $0 \leq N \leq 20$ และ $0 \leq T \leq 1,000$

โกโก้พลอย (garnet)

คนไทยเรามีสำนวนเกี่ยวกับโกโก้มากมาย เช่น “เขียนแบบโกโก้” หมายถึง ลักษณะเขียนที่หัดจนไม่มีใครอ่านออก หรือ “โกโก้แม่ปลาช่อน” จากการที่รู้ว่าโกโก้มีหนังเหนียวและมีเล่ห์เหลี่ยมแพรวพราว ดังนั้น โกโก้จึงถูกนำมาเปรียบเทียบกับสตรีที่สูงวัย จัดเจนสิ่งเวียง และมีมารยาทมาก หรือสำนวน “โกโก้รองบ่อน” นั้นหมายถึง ผู้ที่อยู่ในฐานะตัวสำรอง ซึ่งจะถูกรับมาใช้แทนเมื่อใดก็ได้ คนๆ นั้นจึงไม่มีความสำคัญ สำนวน “โกโก้เห็นดินงู เห็นนมโกโก้” หมายถึงคนที่ต่างฝ่ายต่างก็รู้ความลับของฝ่ายตรงกันข้าม ทั้งๆ ที่โกโก้ไม่มีนม และงูก็ไม่มีตีน หรือสุดท้ายสำนวน “โกโก้ได้พลอย” หมายถึงคนที่ได้ของมีค่า แต่ไม่เป็นประโยชน์สำหรับตน ดังนั้นจึงผู้การได้ของไม่มีค่า แต่มีประโยชน์ไม่ได้

ย่อหน้าด้านบนยกมาจากบทความเรื่องโลกของระกา โดย ศ.ดร.สุทัศน์ ย์ก้าน

ด้วยความอยากรู้อยากเห็นของชาว IT กลุ่มหนึ่ง ที่ต้องทำรายงานวิจัยเกี่ยวกับลักษณะความเป็นมาและความหมายของสำนวน “โกโก้ได้พลอย” บุคคลกลุ่มนี้จึงได้คิดเกมขึ้นมาหนึ่งเกม เพื่อใช้ประกอบรายงานวิจัยนั้น โดยในลานกว้างแห่งหนึ่ง ซึ่งมีขนาด $C \times R$ ช่อง จะมีพลอยจำนวน N ชิ้น วางกระจัดกระจาย โดยที่แต่ละชิ้นจะถูกวางอยู่ที่ช่องพิกัด (X_i, Y_i) ที่แตกต่างกัน และหลังจากนั้นพวกเขาจะนำโกโก้ตัวหนึ่งที่ได้ถูกคัดเลือกมาแล้วว่ามีคุณสมบัติอันแสนพิเศษกว่าโกโก้ตัวใดๆ นั่นคือ โกโก้ที่ถูกเลือกมานี้ สามารถฟังคำสั่งภาษามนุษย์รู้เรื่อง

โกโก้จะถูกวางไว้ ณ จุดเริ่มต้นที่ตำแหน่งหนึ่ง (X_s, Y_s) และหันหน้าไปทางใดทางหนึ่งในลานขนาด $C \times R$ นี้ โดยจะมีคนสั่งอยู่หนึ่งคนยืนอยู่นอกลานกว้าง คนสั่งนี้จะตะโกนทั้งหมด T ครั้ง แต่ละครั้งสามารถมีได้เพียง 4 แบบ ได้แก่ หันซ้าย หันขวา เดินหน้า ถอยหลัง ซึ่งเป็นคำสั่งที่โกโก้สามารถฟังรู้เรื่อง การเดินหน้าหรือถอยหลังของโกโก้ จะเดินจากช่องเดิมที่มันยืนอยู่ ไปยังช่องถัดไปทางด้านหน้าหรือด้านหลังเพียง 1 ช่องเท่านั้น และไม่สามารถเดินสไลด์ไปทางช่องด้านข้างได้ ทั้งนี้หากว่าคนสั่งตะโกนบอกให้โกโก้เดินหน้าหรือถอยหลังออกนอกบริเวณลานกว้าง โกโก้พิเศษนี้จะรู้ตัวโดยสัญชาตญาณว่าโดนแกล้ง และจะไม่สนใจคำสั่งนั้นๆ

เมื่อโกโก้เดินเข้าไปในช่องที่มีพลอยชิ้นใดๆ วางไว้อยู่ จะถือว่าโกโก้สามารถเก็บพลอยชิ้นนั้นได้ทันที ก่อนที่จะรับคำสั่งถัดไปจากคนสั่ง

รับประกันได้ว่าจะไม่มีการพลอยชิ้นใดๆ วางอยู่ ณ จุดเริ่มต้นของโกโก้

หากว่าสิ้นสุดคำสั่ง และโกโก้ไปจอดอยู่ที่ตำแหน่งที่มีพลอยพอดี ให้ถือว่าโกโก้สามารถเก็บพลอยชิ้นนั้นได้ เมื่อสิ้นสุด T คำสั่งแล้ว ถามว่าโกโก้ตัวนี้จะสามารถเก็บพลอยได้ทั้งหมดกี่ชิ้น

ข้อมูลนำเข้า

บรรทัดแรก	จำนวนเต็มบวก C และ R (หลัก x แถว) โดยที่ $1 \leq C \leq 100,000$ และ $1 \leq R \leq 100,000$
บรรทัดที่ 2	$X_s Y_s$ คั่นด้วยช่องว่าง เป็นตำแหน่งเริ่มต้น (หลัก, แถว) ของโกโก้ โดย $1 \leq X_s \leq C$ และ $1 \leq Y_s \leq R$
บรรทัดที่ 3	อักขระ 1 ตัวแทนทิศทางการหันหน้าโกโก้ ณ ตำแหน่งเริ่มต้น กำหนดให้ N =ทิศเหนือ E =ทิศตะวันออก S =ทิศใต้ W =ทิศตะวันตก
บรรทัดที่ 4	จำนวนเต็มบวก N โดยที่ $1 \leq N \leq 100,000$

