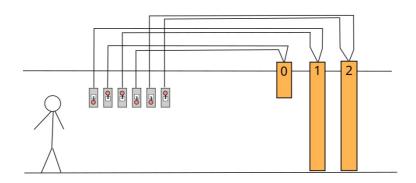
คู่สวิตช์ (switchpairs)

มีถ้ำแห่งหนึ่ง มีประตู N บานที่เรียงต่อกันขวางทางอยู่ ($1 \leq N \leq 500$) เรียกเป็นประตู $0,1,\ldots,N-1$ โดยที่ประตู 0 อยู่หน้าสุด ประตู N-1 อยู่ด้านในสุด ประตูแต่ละบานมีหมายเลขเขียนไว้แต่เนื่องจากคุณไม่ อยากเข้าไปเสี่ยงในถ้ำ การจะสังเกตว่าประตูหน้าสุดที่ปิดอยู่เป็นประตูอะไร จะต้องทำโดยการใช้กล้องพิเศษในการ มอง

ประตูเหล่านี้ควบคุมด้วยสวิตซ์จำนวน 2N อัน (เรียกเป็นสวิตซ์ S_0 ถึงสวิตซ์ S_{2N-1}) สวิตซ์แต่ละสวิตซ์จะมี สถานะได้สองแบบคือ **เปิด** (on) กับ **ปิด** (off) คุณสามารถควบคุมสวิตซ์เหล่านี้ได้ ประตู i จะถูกควบคุมด้วย สวิตซ์จำนวนสองสวิตซ์พอดี สวิตซ์ใด ๆ จะมีผลกับประตูแค่หนึ่งบานเท่านั้น แต่คุณไม่ทราบความสัมพันธ์นี้ ถ้า ประตู i ถูกควบคุมด้วยสวิตซ์ S_{j_1} และ S_{j_2} จะมีรูปแบบของสถานะของสวิตซ์ทั้งสองแค่หนึ่งรูปแบบจาก (on,on), (on,off), (off,on), หรือ (off,off) เท่านั้นที่จะทำให้ประตู i เปิดได้

คุณสามารถปรับเปลี่ยนสวิตช์จากนั้นใช้กล้องพิเศษในการหาว่าหมายเลขของประตูบานหน้าสุดที่ปิดอยู่ หรืออาจ จะเห็นว่าประตูทุกบานเปิดหมดแล้ว ให้คุณหาวิธีเปิดประตูทั้ง N บานนี้ ผ่านทางการสับสวิตช์ คุณสามารถใช้ กล้องพิเศษได้ไม่เกิน Q ครั้ง ($Q=15\,000$)

พิจารณาตัวอย่างดังด้านล่างที่ N=3



รายการสวิตช์ที่ควบคุมประตูเป็นดังนี้

ประตู i	สวิตช์	สถานะที่ทำให้ประตู i เปิด		
0	1,3	$S_1=\mathrm{on},~S_3=\mathrm{off}$		
1	0,2	$S_0={ m off}, \ \ S_2={ m off}$		
2	4,5	$S_4=\mathrm{on},~S_5=\mathrm{on}$		

ถ้าคุณปรับสถานะของสวิตช์เป็นค่าต่าง ๆ ผลลัพธ์และสิ่งที่คุณสังเกตได้จากกล้องพิเศษจะเป็นดังนี้

S_0	S_1	S_2	S_3	S_4	S_5	สถานะประตู (เรียงจาก $0,1,2$)	ประตูบานหน้าสุดที่เห็น
on	on	on	on	on	on	ปิด,ปิด,เปิด	0
on	on	on	off	on	off	เปิด,ปิด,ปิด	1
on	on	on	off	on	on	เปิด,ปิด,เปิด	1
off	on	off	off	on	off	เปิด,เปิด,ปิด	2
off	on	off	off	on	on	เปิด,เปิด,เปิด	ไม่มี (เปิดทุกบานแล้ว)

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

void solve(int N)

- ullet พารามิเตอร์ N แทนจำนวนประตู
- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง
- ullet ฟังก์ชันนี้จะต้องพยายามหาทางเปิดประตูทั้ง N บาน

ฟังก์ชัน solve จะสามารถเรียกฟังก์ชัน control_switches ได้ไม่เกิน Q ครั้ง

int control_switches(vector<int> V)

- ullet ฟังก์ชันนี้จะถูกเรียกได้ไม่เกิน Q ครั้ง ถ้าเรียกเกินโปรแกรมจะจบการทำงานและถือว่าเกิดความผิดพลาด
- ullet อาร์เรย์ V ์เป็นอาร์เรย์ขนาด 2N ที่แต่ละช่องมีค่า 1 หรือ 0 ที่แทนสถานะของสวิตช์ เราจะกำหนดให้ $S_i= ext{on}$ เมื่อ V[i]=1 และ $S_i= ext{off}$ เมื่อ V[i]=0
- ฟังก์ชันจะคืนหมายเลขของประตูหน้าสุดที่ปิดอยู่ ถ้าประตูเปิดหมดแล้วโปรแกรมจะทำงานสำเร็จและจบ การทำงาน

เงื่อนไข

- $1 \le N \le 500$
- Q = 15000

ปัญหาย่อย

- 1. (10 points) $N \leq 10$
- 2. (5 points) สำหรับทุกๆ ประตู i จะมีจำนวนเต็ม j ที่สวิตช์ S_{2j} และ S_{2j+1} ควบคุมประตูนั้นอยู่, และ $N \leq 80$
- 3. (7 points) สำหรับทุกๆ ประตู i จะมีจำนวนเต็ม j ที่สวิตช์ S_{2j} และ S_{2j+1} ควบคุมประตูนั้นอยู่, และ $N \leq$

160

- 4. (8 points) สำหรับทุกๆ ประตู i จะมีจำนวนเต็ม j ที่สวิตช์ S_{2j} และ S_{2j+1} ควบคุมประตูนั้นอยู่
- 5. (10 points) $N \le 200$
- 6. (19 points) N < 300
- 7. (20 points) $N \leq 400$
- 8. (21 points) ไม่มีเงื่อนไขเพิ่มเติมอื่น ๆ

ตัวอย่าง

สำหรับตัวอย่างข้างต้น เกรดเดอร์จะเรียก

```
solve(3)
```

ฟังก์ชัน solve จะสามารถเรียกฟังก์ชัน control_switches ได้ โดยค่าที่ตอบมาสำหรับการเรียกแต่ละครั้ง จะเป็นดังต่อไปนี้

```
control_switches([1, 1, 1, 1, 1])
```

สถานะของสวิตช์คือทุกสวิตช์นั้น on ซึ่งจะทำให้ประตู 0 และ 1 ปิด ส่วนประตู 2 นั้นเปิดเนื่องจาก $S_4=S_5=$ on เนื่องจากประตูหน้าสุดที่ปิดอยู่คือประตู 0 ดังนั้นฟังก์ชัน control_switch จะคืนค่า 0

```
control_switches([1, 1, 1, 0, 1, 0])
```

เนื่องจาก $S_1={
m on}$ และ $S_3={
m off}$ ประตู 0 จะเปิด ส่วนประตู 1 และ 2 นั้นปิด $\,$ เนื่องจากประตูหน้าสุดที่ปิดอยู่ คือประตู 1 ดังนั้นฟังก์ชัน ${
m control}_{
m switch}$ จะคืนค่า 1

```
control_switches([1, 1, 1, 0, 1, 1])
```

เนื่องจาก $S_1={
m on}$ และ $S_3={
m off}$ ประตู 0 จะเปิด นอกจากนี้ เนื่องจาก $S_4=S_5={
m on}$ ทำให้ประตู 2 ก็เปิด ส่วนประตู 1 ยังปิดอยู่ เนื่องจากประตูหน้าสุดที่ปิดอยู่คือประตู 1 ดังนั้นฟังก์ชัน ${
m control}_{-}$ switch จะคืนค่า 1

```
control_switches([0, 1, 0, 0, 1, 0])
```

เนื่องจาก $S_1={
m on}$ และ $S_3={
m off}$ ประตู 0 จะเปิด นอกจากนี้ เนื่องจาก $S_0=S_2={
m off}$ ทำให้ประตู 1 ก็เปิด ส่วนประตู 2 ยังปิดอยู่ เนื่องจากประตูหน้าสุดที่ปิดอยู่คือประตู 2 ดังนั้นฟังก์ชัน ${
m control}_{_}$ switch จะคืนค่า 2

```
control_switches([0, 1, 0, 0, 1, 1])
```

เนื่องจากสถานะของสวิตช์ที่ส่งมานี้ ทำให้ทุกประตูเปิดขึ้น ฟังก์ชัน control_switches จะถือว่าคุณทำงาน สำเร็จและจบการทำงาน

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลดังนี้:

- ullet บรรทัด 1: N
- บรรทัด 2+i (สำหรับ $0\leq i < N$): a_i b_i x_i y_i เป็นข้อมูลของประตู i โดยที่ a_i และ b_i แทนหมายเลขสวิตช์ ($0\leq a_i < 2N, 0\leq b_i < 2N$) และประตู i จะเปิดก็ต่อเมื่อ $S_{a_i}=x_i$ และ $S_{b_i}=y_i$ โดยที่ x_i กับ y_i จะเป็น 1 เพื่อแทนสถานะ on และเป็น 0 เพื่อ แทนสถานะ off

ถ้าเรียกฟังก์ชัน control_switches เกิน Q ครั้ง เกรดเดอร์จะแสดงความผิดพลาด ถ้าเรียกไม่เกินและ สามารถเปิดประตูได้ เกรดเดอร์ตัวอย่างจะแสดงจำนวนครั้งที่เรียก control_switches

ขอบเขต

Time limit: 1 secondsMemory limit: 512 MB