หุ่นยนต์ใต้พิภพ (robots)

นนท์กำลังเข้าร่วมการแข่งขันหุ่นยนต์ใต้พิภพ ในการแข่งขันนี้ นนท์จะต้องออกแบบหุ่นยนต์ที่จะต้องเดินผ่านเส้นทางตรงที่ถูกแบ่งเป็น N ช่อง บางช่องจะเป็นหิน แต่บางช่องจะเป็นลาวา ซึ่งกำหนดโดยสตริง S โดยที่สำหรับแต่ละ $i = 0, 1, \ldots, N-1$, $S[i] = \texttt{L}$ ก็ต่อเมื่อช่องที่ i เป็นลาวา และ $S[i] = \texttt{R}$ ก็ต่อเมื่อช่องที่ i เป็นหิน

เพื่อเอาชนะการแข่งขันครั้งนี้ นนท์ได้ออกแบบหุ่นยนต์ที่มีสองขา ได้แก่ขาซ้ายซึ่งวางอยู่บนช่องที่ l ($0 \le l < N$) และขาขวาซึ่งวางอยู่บนช่องที่ r ($0 \le r < N$) เมื่อการแข่งขันเริ่มต้นขึ้น นนท์จะวางหุ่นยนต์โดยให้ขาซ้ายอยู่ในตำแหน่ง $l = 0$ หลังจากนั้น นนท์จะทำการบังคับหุ่นยนต์โดยการขยับขาซ้ายหรือขาขวาอย่างใดอย่างหนึ่งเท่านั้น โดยให้สอดคล้องกับเงื่อนไขต่อไปนี้ตลอดเวลา รวมทั้งตอนเริ่ม

- ขาทั้งสองข้างจะต้องวางอยู่บนช่องที่เป็นหินตลอดเวลา นั้นคือช่องที่ l และ r จะต้องเป็นหิน
- ระยะห่างระหว่างขาทั้งสองข้างจะต้องสอดคล้องกับเงื่อนไข $A \le r - l \le B$ เมื่อ A และ B เป็นจำนวนเต็มบวกที่ผู้จัดการแข่งขันให้มา

ภารกิจของนนท์จะสำเร็จลุล่วงก็ต่อเมื่อขาขวาของเขาอยู่บนช่องสุดท้าย นั่นคือ $r = N - 1$ และนนท์จึงได้ขอให้คุณเขียนโปรแกรมเพื่อหาว่าสำหรับสตริง S และค่า A, B ที่ให้มา นนท์จะสามารถทำภารกิจให้สำเร็จได้หรือมั้ย

แต่เนื่องจากการแข่งขันจัดที่ใต้ผิวโลกซึ่งมีความไม่เสถียรเป็นอย่างยิ่ง สตริง S จึงสามารถเปลี่ยนแปลงได้ โดยที่การเปลี่ยนแปลงแต่ละครั้งจะถูกกำหนดด้วยจำนวนเต็ม K ($0 \le K < N$) ซึ่งจะทำให้ช่องที่ K เปลี่ยนจากลาวาเป็นหินหรือหินเป็นลาวา การเปลี่ยนแปลงนี้จะมีผลต่อการประมวลผลการเปลี่ยนแปลงครั้งถัดๆ ไป

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้:

```
void init_robots(string S, int A, int B)
```

- ฟังก์ชันนี้จะถูกเรียกใช้โดยเกรดเดอร์เพียงครั้งเดียว
- S คือสตริงที่กำหนดช่องที่เป็นหิน/ลาวา ณ ตอนเริ่มต้น
- A คือระยะห่างต่ำสุดระหว่างขาทั้งสองข้างของหุ่นยนต์
- B คือระยะห่างสูงสุดระหว่างขาทั้งสองข้างของหุ่นยนต์

```
bool update(int K)
```

- ฟังก์ชันนี้จะถูกเรียกทั้งสิ้น Q ครั้ง
- K คือตำแหน่งของช่องที่จะถูกเปลี่ยนจากลาวาเป็นหิน หรือจากหินเป็นลาวา
- **หลังจาก**ทำการเปลี่ยนช่องที่กำหนดแล้ว ฟังก์ชันนี้จะต้องคืนค่า `true` ถ้านนท์สามารถทำภารกิจให้สำเร็จ

ได้ และ `false` หากทำไม่ได้

เงื่อนไข

- $3 \leq N \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $S[i]$ มีค่าเป็น L หรือ R ทุกจำนวนเต็ม $0 \leq i < N$
- $S[0] = S[N-1] = \texttt{R}$
- $1 \leq A < B \leq N$
- $1 \leq K \leq N-2$

ปัญหาย่อย

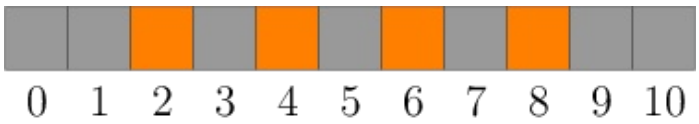
1. (3 คะแนน) $N, Q \leq 100$
2. (2 คะแนน) จำนวนหิน ณ ขณะใดขณะหนึ่งมีค่าไม่เกิน 50 ช่อง
3. (8 คะแนน) $A = 1; B = 2$
4. (9 คะแนน) $A = 1$
5. (14 คะแนน) $N, Q \leq 2\,000$
6. (18 คะแนน) $A = 2$
7. (9 คะแนน) มีเฉพาะการเปลี่ยนจากหินเป็นลาวาเท่านั้น
8. (37 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

เกรดเดอร์สามารถเรียก

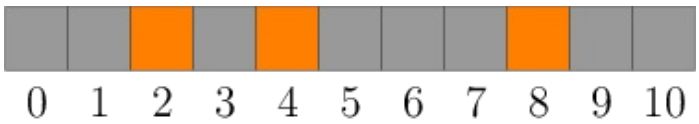
```
init_robots("RRLRLRLRLRR", 2, 5)
```

นั่นคือแต่ละช่องจะเป็นดังรูปต่อไปนี้ (สีส้มแทนลาวา และสีเทาแทนหิน)

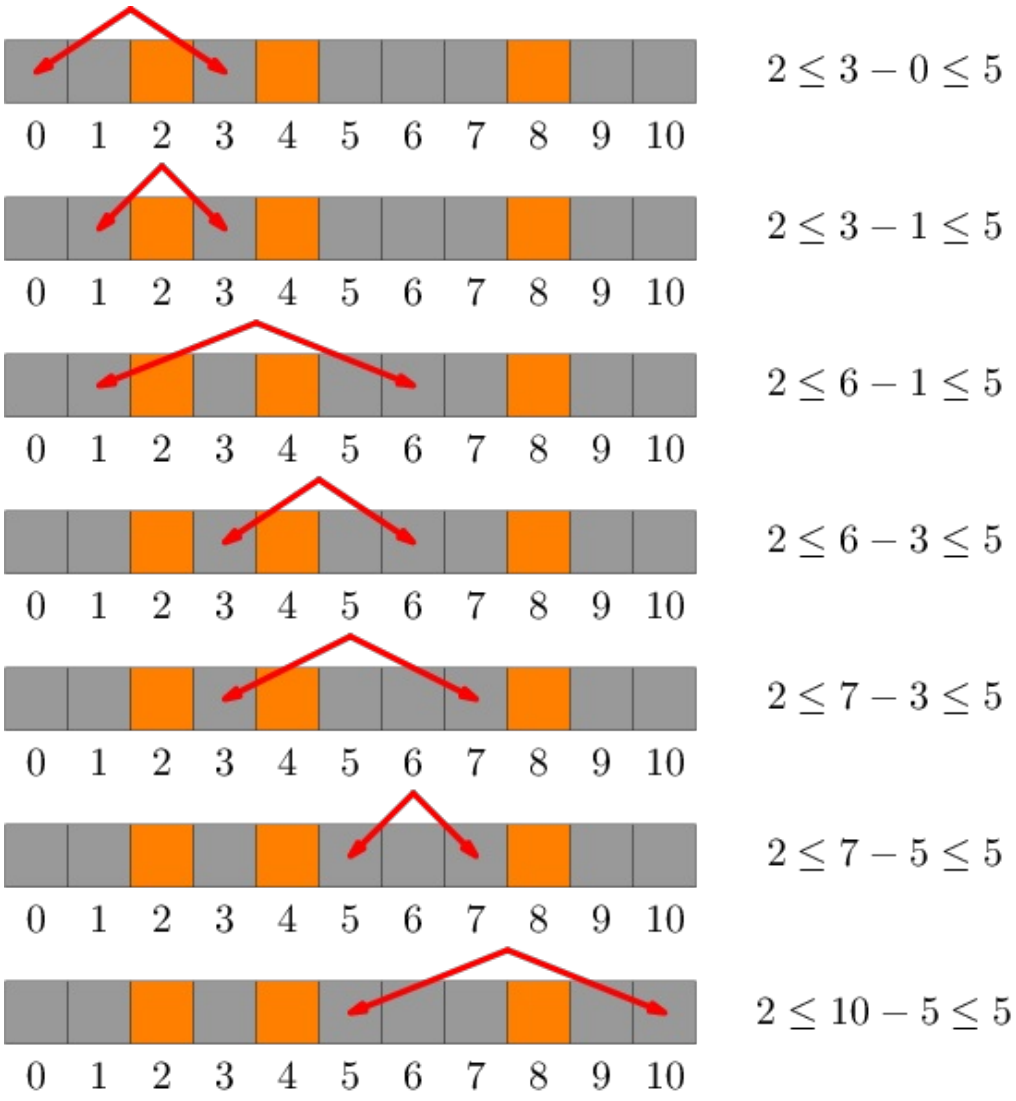



```
update(6)
```

ช่อง 6 ถูกเปลี่ยนจากลาวาเป็นหิน ดังรูป



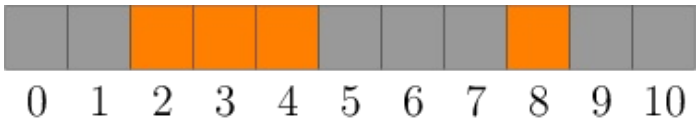
และนนท์สามารถทำภารกิจให้สำเร็จได้ โดยการขยับหุ่นยนต์ดังรูป



ดังนั้น ฟังก์ชันควรจะคืนค่า `true`

```
update(3)
```

ช่อง 3 ถูกเปลี่ยนจากหินเป็นลาวา ดังรูป

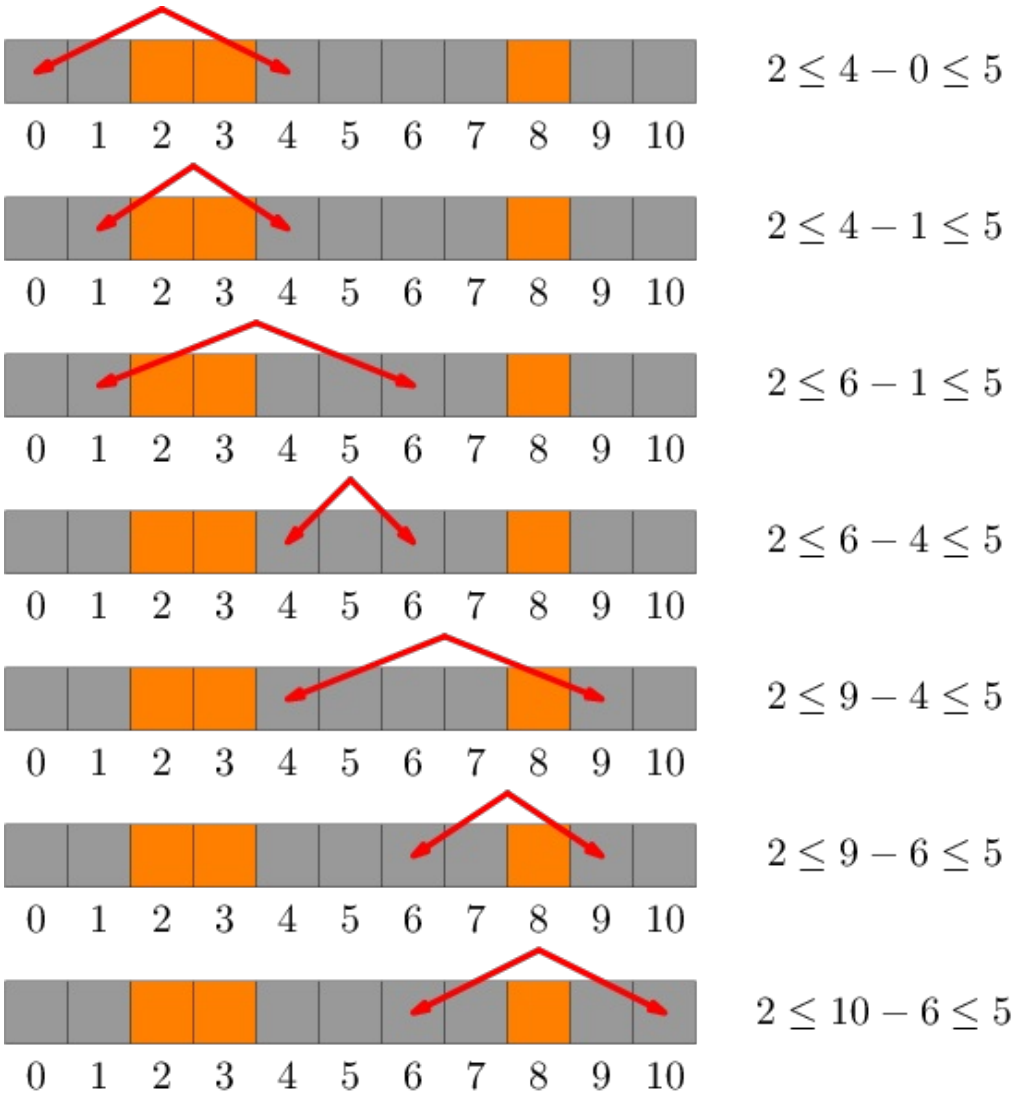


ในกรณีนี้ สามารถแสดงได้ไม่ยากว่านนท์ไม่สามารถทำให้ภารกิจสำเร็จได้ ดังนั้น ฟังก์ชันควรจะคืนค่า `false`

```
update(4)
```

ช่องที่ 4 ถูกเปลี่ยนจากลาวาเป็นหิน ดังรูป

และนนท์สามารถทำภารกิจให้สำเร็จได้ โดยการขยับหุ่นยนต์ดังรูป



ดังนั้น ฟังก์ชันควรจะคืนค่า `true`

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังนี้

- บรรทัดที่ 1: $Q \quad A \quad B$
- บรรทัดที่ 2: S
- บรรทัดที่ 3 ถึง $Q + 2$: K

เกรดเดอร์จะส่งออก Q ค่าที่ได้รับจากการเรียกฟังก์ชัน `update` แต่ละครั้ง

ข้อจำกัด

- Time limit: 1.5 seconds
- Memory limit: 512 MB