ภูเขา (Mountain)

ไอซี่เป็นผู้จัดการแข่งขันวิ่งขึ้น-ลง ภูเขาในปี 3024 ไอซี่ต้องการจัดการแข่งขันที่มีการขึ้นและลงอย่างน้อยอย่างละหนึ่งครั้ง แต่เขาก็ไม่เชี่ยวชาญด้านภูเขาแถว ๆ นี้ เขาจึงต้องไปถามปลื้มผู้เป็นนักปีนเขาที่ปีนเขาแถว ๆ นี้มาแล้วนับครั้งไม่ถ้วน

หลังจากที่ไอซี่ถามปลื้มแล้วได้ข้อมูลมาว่า ภูเขาแถว ๆ นี้มีอยู่ N ลูก เรียงตัวกันเป็นเส้นตรงโดยปลื้มได้กำหนดภูเขาลูกแรกเป็นภูเขาลูกที่ 0 ลูกถัดมาเป็นลูกที่ 1 ถัดไปเป็นลูกที่ 2 ไปเรื่อย ๆ จนถึงภูเขาลูกที่ $N-1$ โดยภูเขาลูกที่ i จะมีราคาค่าผ่านทางเท่ากับ $A[i]$

ในตอนแรกไอซี่ได้วางแผนไว้ว่าภูเขาลูกที่ i จะมีสถานะ 3 รูปแบบได้แก่

1. ภูเขาที่เป็นจุดเริ่มการแข่งขัน
2. ภูเขาที่เป็นจุดจบการแข่งขัน
3. ภูเขาธรรมดาที่ไม่เป็นทั้งจุดเริ่มและจุดจบการแข่งขัน

โดยจุดเริ่มและจุดจบอาจมีหลายจุดได้

เนื่องจากมาร์คผู้เป็นนายทุนของการแข่งขันนี้มีเงินมากมายมหาศาล เขาจึงต้องให้การจัดการแข่งขันขึ้นทุกแบบที่เป็นไปได้ แต่มาร์คก็ยังไม่อยากจะเสียเงินมากนักจึงสั่งให้ ไอซี่ไปจัดการต่อรองกับทางเทพเจ้าแพะเพื่อลดราคาค่าผ่านทาง เทพเจ้าแพะจึงลดค่าผ่านทางจากผลรวมของค่าผ่านทางของเส้นทางการแข่งขันให้เหลือเพียงค่าผ่านทางที่สูงที่สุดของเส้นทางการแข่งขันเท่านั้น (รวมจุดเริ่มและจุดจบด้วย)

ไม่เพียงเท่านี้มาร์คยังชอบแกล้งไอซี่ด้วยการเปลี่ยนสถานะของภูเขาหมายเลข X ให้กลายเป็นสถานะ S แล้วให้คำนวณราคาค่าใช้จ่ายทั้งหมดให้เขาอีกครั้ง โดยเขาจะทำการเปลี่ยนสถานะของภูเขาทั้งหมด Q ครั้ง ซึ่งหากมาร์คเปลี่ยนสถานะของภูเขาลูกใด ๆ แล้วการเปลี่ยนครั้งถัด ๆ ไปจะคำนึงถึงการเปลี่ยนครั้งนี้ด้วย เนื่องจากไอซี่ก็เตรียมงานจนเหนื่อยแล้วยังต้องโดนมาร์คแกล้งอีก เขาจึงต้องการให้**คุณ**ช่วยเขาคำนวณค่าใช้จ่ายของการแข่งขันนี้ให้

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้

```
void initialize(int N, vector<int> A, vector<int> C)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว ก่อนการเรียกฟังก์ชัน `race_cost` ทุกครั้ง
- ตัวแปร N ระบุจำนวนภูเขา โดยภูเขาจะมีหมายเลขตั้งแต่ 0 ไปจนถึง $N-1$
- สำหรับ $i = 0, 1, 2, \ldots, N-1$, ภูเขาลูกที่ i ใช้ค่าจัดงาน $A[i]$ และเป็นภูเขาประเภท $C[i]$ โดย หากค่า

ของ $C[i]$ มีได้ 3 รูปแบบได้แก่

1. $C[i] = 0$ แทน i เป็นภูเขาธรรมดา
2. $C[i] = 1$ แทน i เป็นภูเขาจุดเริ่ม
3. $C[i] = 2$ แทน i เป็นภูเขาจุดจบ

```
long long race_cost(int S, int X)
```

- ฟังก์ชันนี้จะถูกเรียกทั้งหมด Q ครั้ง
- ฟังก์ชันนี้จะทำการเปลี่ยนประเภทของภูเขา X จากเดิมให้กลายเป็น S โดยรับประกันว่าจะเปลี่ยนสถานะของภูเขาจากเดิมเสมอ

ข้อจำกัด

- $2 \leq N, Q \leq 100\,000$
- $1 \leq A[i] \leq 100\,000\,000$ สำหรับ $0 \leq i \leq N-1$
- $0 \leq S, C[i] \leq 2$ สำหรับ $0 \leq i \leq N-1$
- $0 \leq X \leq N-1$

ปัญหาย่อย

1. (6 คะแนน) $N, Q \leq 1\,000$
2. (9 คะแนน) $N, Q \leq 5\,000$
3. (13 คะแนน) $A[i-1] < A[i]$ สำหรับ $1 \leq i \leq N-1$
4. (21 คะแนน) $A[i] \leq 20$ สำหรับ $0 \leq i \leq N-1$
5. (51 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

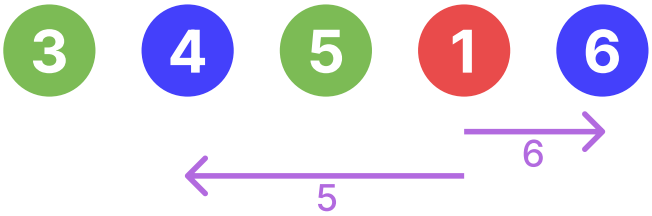
```
initialize(5, [3, 4, 5, 1, 6], [0, 0, 0, 1, 2])
```

เมื่อวงกลมสีแดงแทนการเป็นจุดเริ่ม, วงกลมสีน้ำเงินแทนการเป็นจุดจบ และวงกลมสีเขียวแทนการไม่เป็นจุดเริ่มและจุดจบ

ถัดมา จะมีการเรียก `race_cost` สองครั้ง ดังนี้

```
race_cost(2, 1)
```

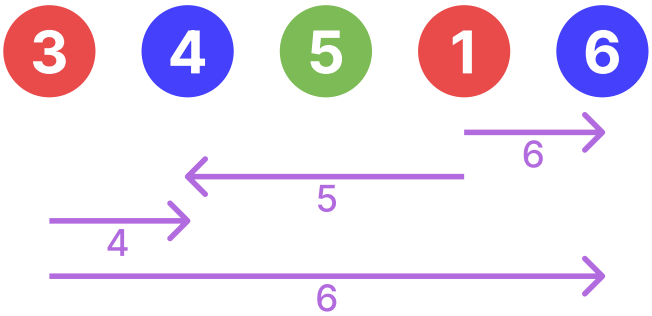
ทำการเปลี่ยนให้ภูเขาลูกที่ 1 กลายเป็นจุดจบ



ฟังก์ชันจะต้องคืนค่า 11 ซึ่งเป็นคำตอบที่ถูกต้อง

```
race_cost(1, 0)
```

ทำการเปลี่ยนให้ภูเขาลูกที่ 0 กลายเป็นจุดเริ่ม



ฟังก์ชันจะต้องคืนค่า 21 ซึ่งเป็นคำตอบที่ถูกต้อง

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลดังต่อไปนี้:

- บรรทัดที่ 1: $N \quad Q$
- บรรทัดที่ 2: $A[0] \quad A[1] \quad A[2] \quad \ldots \quad A[N-1]$
- บรรทัดที่ 3: $C[0] \quad C[1] \quad C[2] \quad \ldots \quad C[N-1]$
- บรรทัดที่ $3+1$ ถึง $3+Q$: $S \quad X$

เกรดเดอร์ตัวอย่างจะพิมพ์ค่าที่คืนจากฟังก์ชัน `race_cost`

ขอบเขต

- Time limit: 2 seconds
- Memory limit: 128 MB