เครื่องสร้างอนุภาค (Particle)

มีเครื่องสร้างอนุภาคทั้งหมด N เครื่อง ในตอนเริ่มต้นแต่ละเครื่องจะมีอนุภาคขนาด 0 อยู่ จะมีคำสั่งทั้งหมด 3 ประเภทดังนี้

1. สั่งให้เครื่องสร้างอนุภาคปรับขนาดอนุภาคจาก S_i เป็น $P - S_i$ เมื่อ S_i คือขนาดอนุภาคที่เครื่องที่ i และ $L \leq i \leq R$
2. ถามค่าผลรวมของขนาดอนุภาคตั้งแต่เครื่องที่ L ถึง R
3. แบ่งอนุภาคออกเป็น 2 กลุ่ม โดยที่ให้กลุ่มแรกมีอนุภาคของเครื่องที่ L ถึง A เป็น**ประจุบวก** และกลุ่มที่ 2 มีอนุภาคของเครื่องที่ $A+1$ ถึง R เป็น**ประจุลบ** เมื่อ $L \leq A \leq R$ เราต้องการจะหา**ค่า A ที่น้อยที่สุด**ที่ทำให้เกิด**แรงลัพธ์น้อยที่สุด** โดยแรงลัพธ์คำนวณได้จาก

$$\sum_{i=L}^{R} \sum_{j=i+1}^{R} (F_{i,j} \times S_i S_j)$$

เมื่อ S_i คือขนาดของอนุภาค $F_{i,j}$ มีค่า 1 ถ้าอนุภาค i, j มีประจุเดียวกัน และ -1 ถ้ามีประจุต่างกัน

เนื่องจากเครื่องสร้างอนุภาคนั้นยังถูกออกแบบให้ใช้งานได้ไม่เต็มประสิทธิภาพ ทำให้การสั่งเครื่องสร้างอนุภาคจำเป็นต้อง**มีค่า P ไม่น้อยกว่าการสั่งในรอบก่อนหน้า** จงหาผลลัพธ์จากทุกๆ คำสั่งประเภทที่ 2 และ 3

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
void init(int N)
```

- N แทนจำนวนเครื่องสร้างอนุภาค
- ฟังก์ชัน `init` ถูกเรียกเพียงครั้งเดียวเท่านั้น และก่อนการเรียกฟังก์ชันอื่นๆ ทุกครั้ง

```
void generateParticle(int L, int R, int P)
```

- L, R, P แทนช่วงที่จะสั่งงานเครื่องสร้างอนุภาค และค่าที่สั่งกับแต่ละเครื่องตามลำดับ
- ฟังก์ชัน `generateParticle` แทนคำสั่งสร้างอนุภาคของเครื่องที่ L ถึง R
- P ในแต่ละคำสั่ง**ไม่น้อยกว่า** P ในรอบก่อนหน้า

```
long long countSize(int L, int R)
```

```

```

- L, R แทนช่วงที่จะถามขนาดอนุภาคของเครื่องสร้างอนุภาค
- ฟังก์ชัน count แทนคำสั่งถามค่าผลรวมของขนาดอนุภาค
- ฟังก์ชันนี้จะต้องคืนค่าจำนวนเต็ม แทนผลรวมของขนาดอนุภาค

```
int bestPartition(int L, int R)

```

- L, R แทนช่วงของเครื่องสร้างอนุภาคที่พิจารณา
- ฟังก์ชัน bestPartition แทนคำสั่งถามค่าตำแหน่ง(A) ของการแบ่งกลุ่มที่ทำให้มีแรงลัพธ์น้อยที่สุด
- ฟังก์ชันนี้จะต้องคืนค่าจำนวนเต็ม แทนตำแหน่งที่ดีที่สุด

ฟังก์ชัน generateParticle, countSize, bestPartition จะถูกเรียกรวมกัน Q ครั้งพอดี

ขอบเขต

- $2 \leq N \leq 1\,000\,000$
- $1 \leq Q \leq 200\,000$
- $1 \leq P \leq 1\,000\,000\,000$
- $0 \leq L \leq R \leq N-1$

ปัญหาย่อย

1. (9 คะแนน) $N, Q \leq 1\,000$ และไม่มีคำสั่งประเภทที่ 3
2. (8 คะแนน) P เท่ากันทุกคำสั่ง คำสั่งประเภทที่ 2 จะเกิดขึ้นหลังประเภทที่ 1 ทั้งหมดและไม่มีคำสั่งประเภทที่ 3
3. (13 คะแนน) P เท่ากันทุกคำสั่ง คำสั่งประเภทที่ 2,3 จะเกิดขึ้นหลังประเภทที่ 1 ทั้งหมด และ $P \leq 10\,000$
4. (9 คะแนน) $L = R$ สำหรับคำสั่งประเภทที่ 1 และไม่มีคำสั่งประเภทที่ 3
5. (8 คะแนน) $L = R$ สำหรับคำสั่งประเภทที่ 2 และไม่มีคำสั่งประเภทที่ 3
6. (14 คะแนน) ไม่มีคำสั่งประเภทที่ 3
7. (20 คะแนน) $Q \leq 50\,000$
8. (19 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

พิจารณาตัวอย่างต่อไปนี้

```
init(6)

```

```
generateParticle(0, 4, 50)
```

```

```

ทำให้ในแต่ละตำแหน่งมีขนาดอนุภาคคือ $50, 50, 50, 50, 50, 0$

```
generateParticle(1, 5, 150)
```

ทำให้ในแต่ละตำแหน่งมีขนาดอนุภาคคือ $50, 100, 100, 100, 100, 150$

```
generateParticle(4, 4, 150)
```

ทำให้ในแต่ละตำแหน่งมีขนาดอนุภาคคือ $50, 100, 100, 100, 50, 150$

```
countSize(1, 5)
```

จะคืนค่า 500

```
bestPartition(3, 5)
```

จะคืนค่า 4 เพราะได้ค่าแรงลัพธ์ $100 \times 50 - 100 \times 150 - 50 \times 150 = -17500$ ซึ่งมีค่าน้อยที่สุดแล้ว

```
bestPartition(1, 3)
```

จะคืนค่า 1 เพราะมีตำแหน่งที่ทำให้ค่าแรงลัพธ์น้อยที่สุดได้ 2 ตำแหน่งคือ 1,2 ซึ่งจะตอบตำแหน่งที่ค่าต่ำที่สุด

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลดังต่อไปนี้:

- บรรทัดที่ 1: $N \;\; Q$
- บรรทัดที่ $1+i$: $type$
  - หาก $type = 1$ รับค่า $L \;\; R \;\; P$
  - หาก $type = 2$ รับค่า $L \;\; R$
  - หาก $type = 3$ รับค่า $L \;\; R$

เกรดเดอร์ตัวอย่างจะส่งออกข้อมูลจำนวนเต็ม 1 จำนวนในทุกบรรทัดสำหรับคำสั่งประเภทที่ 2,3

ข้อจำกัด

- Time limit: 1.25 seconds
- Memory limit: 512 MB