



ดินแดนทองคำ

ในอาณาจักรอันอุดมสมบูรณ์แห่งหนึ่ง ในน้ำมีปลา ในนามีข้าว พื้นแผ่นดินนี้มี *ทำเลทอง* อยู่หลากหลายพื้นที่ (จริง ๆ แล้ว อาจไม่มี หรืออาจมีเพียงที่เดียวก็เป็นได้)

เราสามารถมองพื้นที่อาณาบริเวณของอาณาจักรนี้ เป็นตารางสี่เหลี่ยมผืนผ้าหรือจัตุรัสขนาด N แถว M คอลัมน์ได้ โดยแต่ละช่องของตารางจะมีจำนวนเต็มบวก แทนค่าจำเพาะคุณภาพดินของพื้นที่ย่อยนั้น ๆ อาณาจักรนี้มีความพิเศษอยู่อย่างหนึ่ง คือ เนื่องจากอาณาบริเวณมีขนาดใหญ่มากถึง $N \times M$ อำมาตย์ของอาณาจักรจึงวิจัยค่าจำเพาะคุณภาพดินของแต่ละพื้นที่ แล้วพบความพิเศษคือ ค่าจำเพาะคุณภาพดินของพื้นที่ย่อยในแถวที่ i คอลัมน์ที่ j สำหรับจำนวนเต็ม $0 \leq i < N$ และ $0 \leq j < M$ จะมีค่าเท่ากับ $A[i] \times B[j]$ เมื่อ A และ B เป็นอาเรย์แทนคุณลักษณะของอาณาจักรนี้ มีขนาด N และ M ตามลำดับ

แน่นอนว่า คงไม่ใช่ทุกพื้นที่ ที่จะอุดมสมบูรณ์ตลอดเวลา อำมาตย์พบว่าพื้นที่ย่อยจะเป็นพื้นที่อุดมสมบูรณ์ ก็ต่อเมื่อค่าจำเพาะคุณภาพดินของพื้นที่ย่อยนั้น เป็นจำนวนกำลังสองสมบูรณ์ (perfect square) กล่าวคือ ค่าจำเพาะคุณภาพดินสามารถเขียนในรูป x^2 สำหรับบางจำนวนนับ x เพื่อความง่าย จะเรียกพื้นที่ที่อุดมสมบูรณ์เหล่านี้ว่า *ทำเลทอง*

ราชาของอาณาจักรนี้ มีแผนจะแบ่งพื้นที่ให้กับขุนนางทั้ง P คน เขาจะมอบพื้นที่รูปสี่เหลี่ยมผืนผ้าหรือจัตุรัสให้กับขุนนางแต่ละคน โดยที่ **ไม่มีใครได้พื้นที่ทับซ้อนกัน** (แต่บางคนอาจได้พื้นที่ติดกัน) เพื่อไม่ให้เกิดปัญหาการแย่งชิงพื้นที่ เขาจึงมอบหมายให้อำมาตย์ทำการแบ่งพื้นที่เหล่านี้ออก อำมาตย์จึงได้ทำการแบ่งพื้นที่ไว้ในใจแล้ว แต่เนื่องจากอาณาบริเวณของอาณาจักรนั้นยิ่งใหญ่มากเกินไป เขาจึงนับจำนวน *ทำเลทอง* ไม่ไหว เขาจึงมอบหมายให้ข้าราชการอย่างคุณ นับจำนวน *ทำเลทอง* ที่อยู่ในพื้นที่ของแต่ละคน ให้กับราชา

ตัวอย่าง สำหรับอาณาจักรพื้นที่ 8×8 ที่มี $A = [5, 4, 1, 2, 8, 9, 3, 7]$ และ $B = [2, 1, 4, 3, 8, 6, 5, 2]$ จะมีค่าจำเพาะคุณภาพดินดังภาพดังต่อไปนี้

	2	1	4	3	8	6	5	2
5	10	5	20	15	40	30	25	10
4	8	4	16	12	32	24	20	8
1	2	1	4	3	8	6	5	2
2	4	2	8	6	16	12	10	4
8	16	8	32	24	64	48	40	16
9	18	9	36	27	72	54	45	18
3	6	3	12	9	24	18	15	6
7	14	7	28	21	56	42	35	14

ต่อมา ราชอาณาจักรแบ่งอาณาจักรให้กับขุนนางจำนวน $P = 10$ คน แต่ละคนจะได้ที่ดินดังนี้ (ตั้งแต่ $L1$ ถึง $L10$)

	2	1	4	3	8	6	5	2
5	10	5	20	15	40	30	25	10
4	8	4	16	12	32	24	20	8
1	2	1	4	3	8	6	5	2
2	4	2	8	6	16	12	10	4
8	16	8	32	24	64	48	40	16
9	18	9	36	27	72	54	45	18
3	6	3	12	9	24	18	15	6
7	14	7	28	21	56	42	35	14

เมื่อตรวจสอบ *ทำเลทอง* แล้วจะได้ดังนี้

		2	1	4	3	8	6	5	2
5	10	5	20	15	40	30	25	10	
4	8	4	16	12	32	24	20	8	
1	2	1	4	3	8	6	5	2	
2	4	2	8	6	16	12	10	4	
8	16	8	32	24	64	48	40	16	
9	18	9	36	27	72	54	45	18	
3	6	3	12	9	24	18	15	6	
7	14	7	28	21	56	42	35	14	

จึงได้ว่าขุนนางแต่ละคน ได้ทำเลทองไป 2, 1, 0, 0, 2, 0, 0, 2, 1, 3 ส่วน ตามลำดับ (เรียงตั้งแต่ที่ดิน $L1$ ถึง $L10$)

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
void init_land(int N, int M, int P, vector<int> A, vector<int> B)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง
- A จะมีขนาด N และ B จะมีขนาด M แทนอาเรย์คุณลักษณะของอาณาจักร ดังที่กล่าวไว้ข้างต้น

```
long long answer_query(int r1, int c1, int r2, int c2)
```

- ฟังก์ชันนี้จะถูกเรียกทั้งหมด P ครั้ง
- สำหรับแต่ละครั้งจะต้องตอบคำถามของอำมาตย์ ว่าภายในอาณาบริเวณย่อยตั้งแต่แถวที่ r_1 ไปจนถึงแถวที่ r_2 คอลัมน์ c_1 จนถึงคอลัมน์ c_2 มี ทำเลทอง อยู่ทั้งหมดกี่พื้นที่ย่อย

ขอบเขต

- $2 \leq N, M \leq 60\,000$
- $1 \leq A[i] \leq 1\,000\,000$ สำหรับทุกจำนวนเต็ม $0 \leq i < N$
- $1 \leq B[i] \leq 1\,000\,000$ สำหรับทุกจำนวนเต็ม $0 \leq i < M$
- $1 \leq P \leq 100\,000$

ปัญหาย่อย

1. (2 คะแนน) ขุนนางแต่ละคนได้รับพื้นที่ขนาด 1 หน่วยเสมอ
2. (8 คะแนน) $N, M, P \leq 100$
3. (22 คะแนน) $N, M \leq 1000$
4. (17 คะแนน) $P = 1$ ขุนนางเพียงคนเดียวได้รับพื้นที่ทั้งอาณาจักร $A[i] \leq 100\,000$ สำหรับทุกจำนวนเต็ม $0 \leq i < N$ และ $B[i] \leq 100\,000$ สำหรับทุกจำนวนเต็ม $0 \leq i < M$
5. (15 คะแนน) $A[i] \leq 20$ สำหรับทุกจำนวนเต็ม $0 \leq i < N$ และ $B[i] \leq 20$ สำหรับทุกจำนวนเต็ม $0 \leq i < M$
6. (36 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

จากตัวอย่างข้างต้น การเรียกฟังก์ชัน `init_land` เริ่มต้นจะเป็นดังนี้

```
init_land(8, 8, 10, [5, 4, 1, 2, 8, 9, 3, 7], [2, 1, 4, 3, 8, 6, 5, 2])
```

ต่อมาจะมีการเรียก `answer_query` ทั้งหมด $P = 10$ ครั้ง ดังนี้

```
answer_query(0, 1, 1, 3)
```

จะต้องคืนค่า 2

```
answer_query(0, 4, 3, 4)
```

จะต้องคืนค่า 1

```
answer_query(0, 7, 0, 7)
```

จะต้องคืนค่า 0

```
answer_query(1, 6, 1, 7)
```

จะต้องคืนค่า 0

```
answer_query(2, 6, 4, 7)
```

จะต้องคืนค่า 2

```
answer_query(6, 6, 7, 7)
```

จะต้องคืนค่า 0

```
answer_query(7, 0, 7, 4)
```

จะต้องคืนค่า 0

```
answer_query(4, 2, 5, 4)
```

จะต้องคืนค่า 2

```
answer_query(4, 0, 5, 0)
```

จะต้องคืนค่า 1

```
answer_query(2, 0, 3, 2)
```

จะต้องคืนค่า 3

เกรตเดอร์ตัวอย่าง

เกรตเดอร์ตัวอย่างอ่านข้อมูลดังนี้

- บรรทัดที่ 1: $N \ M \ P$
- บรรทัดที่ 2: $A[0] \ A[1] \ A[2] \ \dots \ A[N-1]$
- บรรทัดที่ 3: $B[0] \ B[1] \ B[2] \ \dots \ B[M-1]$
- บรรทัดที่ $3 + i$ ถึง $3 + P$: $r_1 \ c_1 \ r_2 \ c_2$

เกรตเดอร์จะพิมพ์ค่าที่คืนจากฟังก์ชัน `answer_query`

ข้อจำกัด

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