

## แม่น้ำไนล์

คุณต้องการส่งสิ่งของ  $N$  ชิ้นผ่านทางแม่น้ำไนล์ สิ่งของแต่ละชิ้นมีหมายเลขตั้งแต่ 0 ถึง  $N - 1$  น้ำหนักของสิ่งของชิ้นที่  $i$  ( $0 \leq i < N$ ) คือ  $W[i]$

คุณใช้เรือพิเศษในการขนส่งสิ่งของ เรือแต่ละลำสามารถบรรทุกสิ่งของได้ **ไม่เกินสองชิ้น**

- ถ้าคุณตัดสินใจส่งสิ่งของหนึ่งชิ้น สิ่งของชิ้นนั้นจะมีน้ำหนักเท่าไรก็ได้
- ถ้าคุณต้องการส่งสิ่งของสองชิ้นในเรือลำเดียวกัน คุณจะต้องมั่นใจว่าเรือนั้นสมดุล กล่าวคือคุณสามารถส่งสิ่งของชิ้นที่  $p$  และ  $q$  ( $0 \leq p < q < N$ ) ในเรือลำเดียวกันก็ต่อเมื่อผลต่างสัมบูรณ์ของน้ำหนักของสิ่งของทั้งสองชิ้นมีค่าไม่เกิน  $D$  นั่นคือ  $|W[p] - W[q]| \leq D$

ในการส่งสิ่งของ คุณจะต้องเสียค่าใช้จ่ายซึ่งขึ้นกับจำนวนสิ่งของที่ถูกบรรทุกในเรือลำเดียวกัน ค่าใช้จ่ายในการขนส่งสิ่งของชิ้นที่  $i$  ( $0 \leq i < N$ ) คือ:

- $A[i]$  ถ้าคุณใส่สิ่งของในเรือของมันเอง หรือ
- $B[i]$  ถ้าคุณใส่สิ่งของในเรือไปพร้อมกับสิ่งของชิ้นอื่น

สังเกตว่าในกรณีหลัง คุณจะต้องเสียค่าใช้จ่ายสำหรับสิ่งของทั้งสองชิ้นบนเรือ กล่าวคือ ถ้าคุณตัดสินใจส่งสิ่งของชิ้นที่  $p$  และ  $q$  ( $0 \leq p < q < N$ ) ในเรือลำเดียวกัน คุณจะเสียค่าใช้จ่าย  $B[p] + B[q]$

การส่งสิ่งของในเรือของมันเองมีค่าใช้จ่ายแพงกว่าการส่งสิ่งของไปพร้อมกับสิ่งของชิ้นอื่นในเรือลำเดียวกันเสมอ นั่นคือ  $B[i] < A[i]$  สำหรับทุก ๆ  $i$  ที่  $0 \leq i < N$

อโนจา! แม่น้ำมีความแปรปรวนและค่าของ  $D$  เปลี่ยนแปลงได้บ่อยครั้ง งานของคุณคือตอบคำถาม  $Q$  คำถามที่มีหมายเลข 0 ถึง  $Q - 1$  คำถามอยู่ในรูปของอาร์เรย์  $E$  ที่มีความยาว  $Q$  คำตอบของคำถามที่  $j$  ( $0 \leq j < Q$ ) คือ ค่าใช้จ่ายรวมน้อยที่สุดสำหรับการขนส่งสิ่งของทั้ง  $N$  ชิ้น เมื่อค่าของ  $D$  เท่ากับ  $E[j]$

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
std::vector<long long> calculate_costs(  
    std::vector<int> W, std::vector<int> A,  
    std::vector<int> B, std::vector<int> E)
```

- $W, A, B$ : อาร์เรย์ของจำนวนเต็มที่มีความยาว  $N$  ระบุน้ำหนักของสิ่งของและค่าใช้จ่ายในการขนส่งสิ่งของ
- $E$ : อาร์เรย์ของจำนวนเต็มที่มีความยาว  $Q$  ระบุค่าของ  $D$  สำหรับแต่ละคำถาม

- ฟังก์ชันนี้ต้องคืนค่าอาร์เรย์ของจำนวนเต็ม  $R$  ความยาว  $Q$  ที่ระบุค่าใช้จ่ายรวมน้อยที่สุดสำหรับการขนส่งสิ่งของทั้งหมด โดยที่สำหรับแต่ละ  $j$  ที่  $0 \leq j < Q$ ,  $R[j]$  คือค่าใช้จ่ายเมื่อค่าของ  $D$  เท่ากับ  $E[j]$
- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้งเท่านั้นสำหรับแต่ละกรณีทดสอบ

## เงื่อนไข

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$  สำหรับแต่ละ  $i$  ที่  $0 \leq i < N$
- $1 \leq B[i] < A[i] \leq 10^9$  สำหรับแต่ละ  $i$  ที่  $0 \leq i < N$
- $1 \leq E[j] \leq 10^9$  สำหรับแต่ละ  $j$  ที่  $0 \leq j < Q$

## ปัญหาย่อย

ปัญหาย่อย	คะแนน	เงื่อนไขเพิ่มเติม
1	6	$Q \leq 5$ ; $N \leq 2000$ ; $W[i] = 1$ สำหรับแต่ละ $i$ ที่ $0 \leq i < N$
2	13	$Q \leq 5$ ; $W[i] = i + 1$ สำหรับแต่ละ $i$ ที่ $0 \leq i < N$
3	17	$Q \leq 5$ ; $A[i] = 2$ และ $B[i] = 1$ สำหรับแต่ละ $i$ ที่ $0 \leq i < N$
4	11	$Q \leq 5$ ; $N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ และ $B[i] = 1$ สำหรับแต่ละ $i$ ที่ $0 \leq i < N$
7	18	ไม่มีเงื่อนไขเพิ่มเติม

## ตัวอย่าง

พิจารณาการเรียกฟังก์ชันดังนี้

```
calculate_costs([15, 12, 2, 10, 21],
                [5, 4, 5, 6, 3],
                [1, 2, 2, 3, 2],
                [5, 9, 1])
```

ในตัวอย่างนี้ เรามีสิ่งของ  $N = 5$  ชิ้นและมี  $Q = 3$  คำถาม

ในคำถามแรก  $D = 5$  คุณสามารถส่งสิ่งของชิ้นที่ 0 และชิ้นที่ 3 ในเรือลำเดียวกัน (เนื่องจาก  $|15 - 10| \leq 5$ ) และส่งสิ่งของชิ้นที่เหลือในเรือคนละลำกัน ผลรวมค่าใช้จ่ายในการส่งสิ่งของทั้งหมดคือ  $1 + 4 + 5 + 3 + 3 = 16$

ในคำถามที่สอง  $D = 9$  คุณสามารถส่งสิ่งของชิ้นที่ 0 และชิ้นที่ 1 ในเรือลำเดียวกัน (เนื่องจาก  $|15 - 12| \leq 9$ ) และส่งสิ่งของชิ้นที่ 2 และชิ้นที่ 3 ในเรือลำเดียวกัน (เนื่องจาก  $|2 - 10| \leq 9$ ) ส่งสิ่งของชิ้นที่เหลือในเรืออีกลำ ผลรวมค่าใช้จ่ายในการส่งสิ่งของทั้งหมดคือ  $1 + 2 + 2 + 3 + 3 = 11$

ในคำถามสุดท้าย  $D = 1$  คุณต้องส่งสิ่งของแต่ละชิ้นในเรือคนละลำกัน ผลรวมค่าใช้จ่ายในการส่งสิ่งของทั้งหมดคือ  $5 + 4 + 5 + 6 + 3 = 23$

ดังนั้นฟังก์ชันนี้จะต้องคืนค่า  $[16, 11, 23]$

## เกรตเตอร์ตัวอย่าง

รูปแบบข้อมูลนำเข้า:

```
N
W[0] A[0] B[0]
W[1] A[1] B[1]
...
W[N-1] A[N-1] B[N-1]
Q
E[0]
E[1]
...
E[Q-1]
```

รูปแบบข้อมูลส่งออก:

```
R[0]
R[1]
...
R[S-1]
```

ในที่นี้  $S$  คือความยาวของอาร์เรย์  $R$  ที่ถูกคืนโดยฟังก์ชัน `calculate_costs`