

멋진 구간 2

영우는 정수로 이루어진 길이 N 의 배열 A, B 를 가지고 있다. 모든 정수 $0 \leq i \leq N - 1$ 에 대해 $A[i] \leq B[i]$ 를 만족한다.

다음 조건을 모두 만족하는 $[l, r]$ 을 멋진 구간이라고 정의한다:

- l, r 은 정수
- $0 \leq l \leq r \leq N - 1$
- 배열 $[A[l], \dots, A[r]]$ 에 다음과 같은 연산을 반복하여 $[B[l], \dots, B[r]]$ 이 되도록 할 수 있다.
 - 현재 배열을 $X = [X[0], X[1], \dots, X[r - l]]$ 이라 하자.
 - $X[i] = X[j]$ 인 서로 다른 두 정수 $0 \leq i, j \leq r - l$ 를 고르고 $X[i]$ 를 1 증가시킨다.

영우는 어떤 구간들이 멋진 구간인지 궁금해졌다.

구체적으로, 영우의 궁금증은 0부터 $Q - 1$ 까지의 번호가 붙은 Q 개의 질문으로 구성되어 있으며, 이는 정수로 이루어진 길이 Q 의 배열 L 과 R 로 표현된다.

j 번 질문은 구간 $[L[j], R[j]]$ 가 멋진 구간인지의 여부이다. ($0 \leq j \leq Q - 1$)

여러분은 영우의 질문들에 답하는 프로그램을 작성해야 한다.

함수 목록 및 정의

다음 함수를 구현해야 한다.

```
vector<int> array_operation(vector<int> A, vector<int> B, vector<int> L,
vector<int> R)
```

- A, B : 크기가 N 인 정수 배열.
- L, R : 크기가 Q 인 정수 배열.
- 이 함수는 크기가 Q 인 정수 배열 S 를 반환해야 한다. $S[j]$ 는 $[L[j], R[j]]$ 가 멋진 구간이라면 1, 그렇지 않다면 0이어야 한다. ($0 \leq j \leq Q - 1$)
- 이 함수는 단 한 번만 호출된다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $1 \leq N, Q \leq 250\,000$
- 모든 i 에 대해 $1 \leq A[i] \leq B[i] \leq 10^9$ ($0 \leq i \leq N - 1$)
- 모든 j 에 대해 $0 \leq L[j] \leq R[j] \leq N - 1$ ($0 \leq j \leq Q - 1$)

부분문제

번호	배점	제한
1	9	$N, Q \leq 100; B[i] \leq 100 (0 \leq i \leq N - 1)$
2	7	$N, Q \leq 2000; A[i] = 1 (0 \leq i \leq N - 1)$
3	16	$A[i] = 1 (0 \leq i \leq N - 1)$
4	10	$N, Q \leq 2000$
5	4	$B[i] \leq 2 (0 \leq i \leq N - 1)$
6	13	$B[i] \leq 100 (0 \leq i \leq N - 1)$
7	31	$B[i] \leq 250\,000 (0 \leq i \leq N - 1)$
8	10	추가적인 제약조건이 없다.

예제

예제 1

다음 호출을 생각해보자:

```
array_operation([2, 1, 1, 2], [2, 1, 3, 3], [0, 0, 1], [1, 3, 3])
```

- $[0, 1]$ 은 멱진 구간이다. 두 배열 $[A[0], A[1]]$ 과 $[B[0], B[1]]$ 이 같기 때문이다.
- $[0, 3]$ 은 멱진 구간이다. $[2, 1, 1, 2]$ 에서 다음과 같은 연산을 시행해 $[2, 1, 3, 3]$ 으로 만들 수 있기 때문이다.
 - $i = 3, j = 0$ 을 선택하여 연산을 시행한다. 연산이 시행된 이후의 배열은 $[2, 1, 1, 3]$ 이 된다.
 - $i = 2, j = 1$ 을 선택하여 연산을 시행한다. 연산이 시행된 이후의 배열은 $[2, 1, 2, 3]$ 이 된다.
 - $i = 2, j = 0$ 을 선택하여 연산을 시행한다. 연산이 시행된 이후의 배열은 $[2, 1, 3, 3]$ 이 된다.
- $[1, 3]$ 은 멱진 구간이 아니다. $[1, 1, 2]$ 에서 어떻게 연산을 시행하여도 $[1, 3, 3]$ 으로 만들 수 없음을 증명할 수 있다.

따라서 함수는 $[1, 1, 0]$ 을 반환해야 한다.

예제 2

다음 호출을 생각해보자:

```
array_operation([1, 2, 1, 2, 1], [2, 3, 1, 4, 2], [0, 0, 1, 1, 2], [2, 4, 3, 4, 3])
```

모든 구간들 중 멱진 구간은 $[0, 2], [0, 3], [0, 4], [1, 4], [2, 2]$ 가 있다. 따라서 함수는 $[1, 1, 0, 1, 0]$ 을 반환해야 한다.

샘플 그레이더

샘플 그레이더의 입력 형식은 다음과 같다.

- line 1: $N Q$

- 모든 $0 \leq i \leq N - 1$ 에 대해:
 - line $2 + i$: $A[i] B[i]$
- 모든 $0 \leq i \leq Q - 1$ 에 대해:
 - line $2 + N + i$: $L[i] R[i]$

샘플 그레이더는 다음 형식으로 답을 출력한다:

- line 1: `array_operation` 의 반환값