Команди

Є клас з N учнів, пронумерованих від 0 до $N-1$. Кожного дня вчителька цього класу пропонує учням проекти. Кожен проект має бути виконаний командою учнів того-ж дня. Проекти мають різну складність. Для кожного проекта вчителька знає точний розмір команди, що має над ним працювати.

Кожному з учнів подобається працювати у команді певного розміру. А саме, учня i можна включити до команди, розмір якої лежить між $A[i]$ і $B[i]$ включно. Кожного дня учня можна включити не більш ніж в одну команду. Деякі учні можуть не потрапити до жодної з команд. Кожна команда буде працювати над одним з проектів.

Вчителька вже обрала проекти на кожен з наступних Q днів. Для кожного з цих днів ви маєте визначити, чи можна розподілити учнів по командам так, щоб над кожним проектом працювала окрема команда.

Приклад

Нехай є $N = 4$ учні та $Q = 2$ дні. Обмеження учнів по розмірам команди наведено в наступній таблиці.

учень	0	1	2	3
A	1	2	2	2
B	2	3	3	4

На перший день є $M = 2$ проекти. Необхідні розміри команд $K[0] = 1$ і $K[1] = 3$. Ці дві команди можна сформувати, призначивши учня 0 до комади розміру 1, а решту учнів до — команди розміром 3.

На другий день знову є $M = 2$ проекти, але цього разу необхідні розміри $K[0] = 1$ і $K[1] = 1$. В цьому випадку неможливо сформувати команди, оскільки є лише один учень, що може працювати у команді розміру 1.

Задача

Дано опис усіх учнів: N, A та B, а також послідовність Q запитань — одне на кожен день. Кожне запитання складається з кількості проектів M на цей день та послідовності K довжини M що містить розміри потрібних команд. Для кожного запитання, ваша програма повинна повернути відповідь, чи можна сформувати всі команди.

Ви повинні реалізувати функції `init` та `can`:

- `init(N, A, B)` — Модуль перевірки викличе цю функцію першою і тільки один раз.

- N: кількість учнів.
- A: масив довжини N: A[i] мінімальний розмір команди для учня i.
- B: масив довжини N: B[i] максимальний розмір команди для учня i.
- Функція не повертає значення.

Передбачається, що $1 \leq A[i] \leq B[i] \leq N$ для кожного $i = 0, \ldots, N-1$.

- can(M, K) — після одиничного виклику init, модуль перевірки буде викликати цю функцію Q разів поспіль, по разу для кожного дня.
  - M: кількість проектів у цей день.
  - K: масив довжини M, що містить розміри команд, потрібні для кожного з проектів.
  - Функція повинна повернути 1, якщо є можливість сформувати всі потрібні команди, та 0 в іншому випадку.
  - Передбачається, що $1 \leq$ M $\leq N$, і що для кожного $i = 0, \ldots,$ M -1 виконується $1 \leq$ K[i] $\leq N$. Зауважте, що сума всіх K[i] може перевищувати N.

Підзадачі

Позначимо через S суму значень M у всіх викликах can(M, K).

Підзадача	Бали	N	Q	Додаткові обмеження
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	немає
2	13	$1 \leq N \leq 100,000$	$Q = 1$	немає
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

Приклад модуля перевірки

Отриманий вами модуль перевірки читає вхідні дані в наступному форматі:

- рядок 1: N
- рядки 2, …, N + 1: A[i] B[i]
- рядок N + 2: Q
- рядки N + 3, …, N + Q + 2: M K[0] K[1] … K[M - 1]

На кожне питання він виводить значення, що повертає функція can.