Teams

Er is een klas met N leerlingen, genummerd van 0 tot en met $N-1$. Elke dag geeft de leraar van die klas een aantal projecten aan de leerlingen. Elk project moet door een team leerlingen op dezelfde dag nog afgerond worden. De projecten kunnen een verschillende moeilijkheidsgraad hebben. Voor elk project weet de leraar de precieze grootte van een team dat er aan zou moeten gaan werken.

Verschillende leerlingen kunnen verschillende voorkeuren hebben voor de grootte van een team. Preciezer: student i kan alleen worden ingedeeld in een team met een grootte van $A[i]$ tot en met $B[i]$. Elke dag kan een leerling in maximaal één team werken. Er kunnen leerlingen zijn die niet aan een team zijn toegewezen. Elk team werkt aan één project.

De leraar heeft de projecten voor de komende Q dagen al bepaald. Ga voor elke dag na of het mogelijk is om de teams zo samen te stellen dat aan elk project een team werkt.

Voorbeeld

Stel dat er $N = 4$ leerlingen zijn en $Q = 2$ dagen. De voorwaarden die de leerlingen stellen aan de groottes van teams staan in de volgende tabel:

leerling	0	1	2	3
A	1	2	2	2
B	2	3	3	4

Op de eerste dag zijn er $M = 2$ projecten. De benodigde teamgroottes zijn $K[0] = 1$ en $K[1] = 3$. Deze twee teams kunnen worden gevormd door leerling 0 toe te kennen aan een team met grootte 1 en de overige drie leerlingen aan een team met grootte 3.

Op de tweede dag zijn er opnieuw $M = 2$ projecten, maar deze keer zijn de vereiste groottes van de teams $K[0] = 1$ en $K[1] = 1$. Het is in dit geval niet mogelijk om de teams samen te stellen, omdat er maar één student is die in een team met grootte 1 past.

Opdracht

Je krijgt de beschrijvingen van alle leerlingen: N, A en B, alsmede een rij met Q vragen — één per dag. Elke vraag bestaat uit het aantal projecten M op die dag en een rij K van lengte M met daarin de vereiste teamgroottes. Voor iedere vraag moet je programma aangeven of het al dan niet mogelijk is om alle teams samen te stellen.

Schrijf de functies `init` en `can`:

- `init(N, A, B)` — De grader zal deze functie als eerste aanroepen en slechts één keer.
  - `N`: het aantal leerlingen.

- A: een array van lengte N: A[i] is de minimale team grootte voor leerling i.
- B: een array met lengte N: B[i] is de maximale team grootte voor leerling i.
- De functie levert geen resultaat op.

Je mag aannemen dat $1 \leq A[i] \leq B[i] \leq N$ voor alle $i = 0, \ldots, N-1$.

- can(M, K) — Nadat init éénmaal is aangeroepen, zal de grader deze functie Q keer achter elkaar aanroepen; één keer voor elke dag.
  - M: het aantal projecten op deze dag.
  - K: een array van lengte M die de vereiste teamgrootte voor elk van deze projecten bevat.
  - De functie geeft als resultaat 1 waneer het mogelijk is om alle vereiste teams te vormen en 0 wanneer dat niet het geval is.
  - Je mag aannemen dat $1 \leq$ M $\leq N$, en dat voor elke $i = 0, \ldots,$ M-1 geldt $1 \leq$ K[i] $\leq N$. Let op dat de som van alle K[i] groter kan zijn dan N.

Subtasks

Noem S de som van alle waarden M in alle aanroepen can(M, K).

subtask	punten	N	Q	Aanvullende voorwaarden
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	geen
2	13	$1 \leq N \leq 100,000$	$Q = 1$	geen
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

Voorbeeldgrader

De voorbeeld grader leest de input in het volgende format:

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

Voor elke vraag drukt de voorbeeld grader de return waarde van can af.