



團隊

有一個班級有 N 個學生，編號從 0 到 $N - 1$ 。每天這班級的老師都有一些專題要給學生完成。每個專題都需要在當天內由一個學生團隊來完成。由於每個專題的困難程度可能不相同，老師知道完成各個專題所需要的團隊人數。

每個學生都會有團隊組成人數的偏好，學生 i 只能被分配到人數在 $A[i]$ 和 $B[i]$ 之間的團隊。在一天當中，一個學生最多被分到一個團隊，有些學生可能沒有被分到任何團隊，而每個團隊將負責一個專題。

老師已經針對未來的 Q 天中，每天指定一些專題。試判斷未來的每一天，是否能適當的將學生分到各團隊，以使每一個專題都有一個學生團隊來完成。

Example

假設有 $N = 4$ 個學生和 $Q = 2$ 天。所有學生能參與的團隊組成人數的限制都列在下表中。

student	0	1	2	3
A	1	2	2	2
B	2	3	3	4

在第一天有 $M = 2$ 個專題，需要人數分別為 $K[0] = 1$ 和 $K[1] = 3$ 團隊。這兩個團隊可以由下面的方式組成：由學生 0 組成人數為 1 的團隊，再由其他三位學生組成人數為 3 的團隊。

在第二天又有 $M = 2$ 個專題，需要的人數分別為 $K[0] = 1$ 和 $K[1] = 1$ 團隊。在這個情況下不可能組成所需要的團隊，因為只有一個學生可以被分到人數為 1 的團隊中。

Task

給定所有學生的相關描述: N, A, B , 以及一個包含 Q 個問題的序列 — 每個問題都對應一天。每個問題包含了在那一天有的專題個數 M ，和一個長度為 M 的序列 K 表示每個專題所需要的團隊人數。對於每一個問題，你的程式必須回傳是否有可能組成所有需要的團隊。

你需要實作以下兩個函數 `init` 和 `can`:

- `init(N, A, B)` — Grader 評分程式會最先呼叫這個函數，且只會呼叫一次。
 - N : 學生人數.
 - A : 是一長度為 N : 的陣列， $A[i]$ 是學生 i 能參與的最小團隊人數.
 - B : 是一長度為 N : 的陣列， $B[i]$ 是學生 i 能參與的最大團隊人數.
 - 這函數沒有回傳數字.

你可以假設 $1 \leq A[i] \leq B[i] \leq N$ for each $i = 0, \dots, N - 1$.

- `can(M, K)` — 呼叫一次 `init` 之後, Grader 評分程式將會連續呼叫這個函數 Q 次, 每一天呼叫一次.
 - M : 是這一天的專題個數.
 - K : 是一個長度為 M 的陣列, 包含了每一個專題所需要的團隊人數.
 - 如果有可能組成所有需要的團隊, 函數必須回傳 1, 否則回傳 0.
 - 你可以假設 $1 \leq M \leq N$, 而且對每一個 $i = 0, \dots, M - 1$, 我們都知 $1 \leq K[i] \leq N$. 請注意所有 $K[i]$ 的總和可能會超過 N .

Subtasks

我們用 S 來表示所有呼叫 `can(M, K)` 的 M 數字的總和.

subtask	points	N	Q	Additional Constraints
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	none
2	13	$1 \leq N \leq 100,000$	$Q = 1$	none
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

Sample grader

範例評分程式會依照下面格式讀取資料:

- line 1: N
- lines 2, ..., $N + 1$: $A[i] B[i]$
- line $N + 2$: Q
- lines $N + 3, \dots, N + Q + 2$: $M K[0] K[1] \dots K[M - 1]$

對每一個問題, 範例評分程式會印出呼叫 `can` 函數的回傳數字.