



# International Olympiad in Informatics 2015

26th July - 2nd August 2015

Almaty, Kazakhstan

Day 1

teams

Language: th-TH

## ทีม

ห้องเรียนหนึ่งมีนักเรียนทั้งหมด  $N$  คน นักเรียนแต่ละคนมีหมายเลขตั้งแต่ 0 ถึง  $N-1$  ในแต่ละวัน คุณครูประจำชั้นจะมอบหมายโครงการจำนวนหนึ่งให้นักเรียนในห้อง แต่ละโครงการจะต้องทำให้เสร็จภายในวันที่มอบหมายด้วยทีมหนึ่งทีมเท่านั้น โครงการอาจมีความยากง่ายแตกต่างกัน ในแต่ละโครงการคุณครูจะทราบจำนวนของสมาชิกในทีมที่จะทำโครงการนั้น

นักเรียนแต่ละคนอาจมีความชอบที่จะทำงานในทีมที่จำนวนสมาชิกแตกต่างกัน กล่าวคือนักเรียนคนที่  $i$  สามารถทำงานในทีมที่มีจำนวนสมาชิกอย่างน้อย  $A[i]$  คนแต่ไม่เกิน  $B[i]$  คน ในแต่ละวัน นักเรียนคนหนึ่งสามารถถูกมอบหมายให้อยู่ได้เพียงทีมเดียวเท่านั้น แต่นักเรียนบางคนอาจไม่ถูกมอบหมายให้อยู่ทีมใดเลยก็ได้ และแต่ละทีมจะทำเพียงแคeworkshopเดียวในแต่ละวัน

คุณครูได้เลือกโครงการทั้งหมดที่จะใช้ใน  $Q$  วันถัดไปเรียบร้อยแล้ว สำหรับแต่ละวันให้หาว่าเป็นไปได้หรือไม่ที่จะมอบหมายนักเรียนให้อยู่ในทีมโดยที่ทุก ๆ โครงการมีทีมหนึ่งทีมที่ทำโครงการนั้น

## ตัวอย่าง

กำหนดให้มีนักเรียนทั้งหมด  $N = 4$  คน และมีทั้งหมด  $Q = 2$  วัน เงื่อนไขจำนวนสมาชิกของทีมของนักเรียนแต่ละคนแสดงดังตารางด้านล่าง

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

วันที่หนึ่งมีทั้งสิ้น  $M = 2$  โครงการ จำนวนสมาชิกที่ต้องการสำหรับโครงการทั้งสองคือ  $K[0] = 1$  คน และ  $K[1] = 3$  คนตามลำดับ ทีมทั้งสองนี้สามารถจัดขึ้นได้โดยให้นักเรียนคนที่ 0 อยู่ในทีมที่มีจำนวนสมาชิก 1 คน และนักเรียนที่เหลือให้อยู่ในทีมที่มีสมาชิก 3 คน

วันที่สองมีทั้งสิ้น  $M = 2$  โครงการ จำนวนสมาชิกที่ต้องการสำหรับโครงการทั้งสองคือ  $K[0] = 1$  คน และ  $K[1] = 1$  คนตามลำดับ แต่ในกรณีนี้ไม่สามารถจัดทีมตามต้องการได้เนื่องด้วยมีนักเรียนเพียงคนเดียวเท่านั้นที่สามารถทำงานในทีมที่มีสมาชิก 1 คนได้

## งานของคุณ

เมื่อกำหนดให้คุณทราบรายละเอียดของนักเรียนทั้งหมด:  $N$ ,  $A$ , และ  $B$  และลำดับของคำถามอีก  $Q$  คำถามซึ่งเป็นคำถามของแต่ละวัน แต่ละคำถามประกอบด้วยตัวเลข  $M$  แสดงจำนวนโครงการที่มีในวันนั้น และลำดับ  $K$  ที่มีความยาว  $M$  ที่แสดงจำนวนสมาชิกที่เหมาะสมของแต่ละโครงการ โปรแกรมของคุณต้องตอบว่าเป็นไปได้หรือไม่ที่จะจัดทีมตามที่ต้องการสำหรับแต่ละคำถาม

คุณต้องเขียนฟังก์ชัน `init` และ `can`:

- `init(N, A, B)` – เกรดเดอร์จะเรียกเรียกฟังก์ชันนี้เพียงครั้งแรกและครั้งเดียว
  - $N$ : จำนวนนักเรียนทั้งหมด
  - $A$ : อาร์เรย์ความยาว  $N$  เมื่อ  $A[i]$  แทนจำนวนสมาชิกที่น้อยที่สุดของทีมที่นักเรียนคนที่  $i$  สามารถทำงานในทีมนั้นได้
  - $B$ : อาร์เรย์ความยาว  $N$  เมื่อ  $B[i]$  แทนจำนวนสมาชิกที่มากที่สุดของทีมที่นักเรียนคนที่  $i$  สามารถทำงานในทีมนั้นได้
  - ฟังก์ชันนี้ไม่คืนค่า (ไม่มี return value)

คุณสามารถสมมติว่า  $A[i] \leq B[i] \leq N$  สำหรับทุกค่า  $i = 0, \dots, N-1$

- `can(M, K)` – หลังจากการเรียก `init` แล้ว เกรดเดอร์จะเรียกฟังก์ชันนี้ทั้งสิ้น  $Q$  ครั้งต่อเนื่องกัน หนึ่งครั้งสำหรับแต่ละวัน
  - $M$ : จำนวนโครงการทั้งหมดในวันนี้
  - $K$ : อาร์เรย์ความยาว  $M$  ที่ระบุขนาดของทีมที่ต้องการของแต่ละโครงการ
  - ฟังก์ชันนี้จะคืนค่า 1 ถ้าเป็นไปได้ที่จะจัดทีมทั้งหมดตามต้องการได้ และคืนค่า 0 สำหรับกรณีอื่น
  - คุณสามารถสมมติว่า  $1 \leq M \leq N$  และสำหรับแต่ละ  $i = 0, \dots, M-1$  เรามี  $1 \leq K[i] \leq N$  หมายถึงผลรวมของค่าทั้งหมดใน  $K[i]$  อาจมีค่าเกิน  $N$

## ปัญหาย่อย

ให้  $S$  แทนผลรวมของค่า  $M$  ทั้งหมดที่เกิดขึ้นจากทุกครั้งในการเรียกใช้ฟังก์ชัน `can(M, K)`

ปัญหาย่อย	คะแนน	$N$	$Q$	ข้อกำหนดเพิ่มเติม
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	ไม่มี
2	13	$1 \leq N \leq 100,000$	$Q = 1$	ไม่มี
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านค่าข้อมูลนำเข้าตามรูปแบบด้านล่าง:

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

แต่ละคำถามข้างต้น เกรตเตอร์ตัวอย่างจะพิมพ์ค่าที่ฟังก์ชัน `can` คืนกลับมา