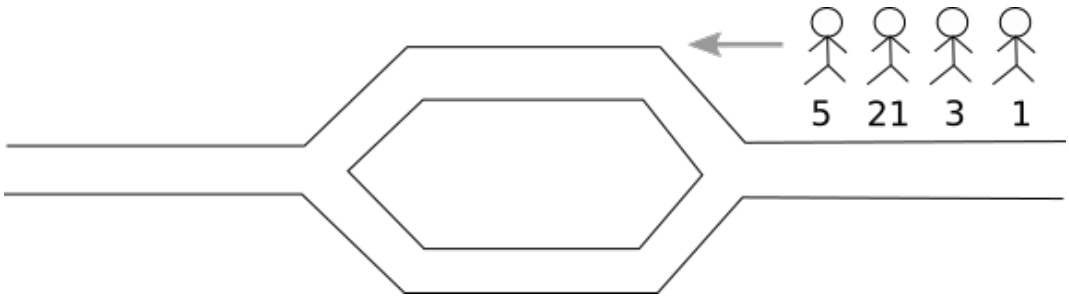
แซงคิว

มีคน N คน ($N \leq 2\,500\,000$) ต้องการเข้าชมการแสดงครั้งยิ่งใหญ่ที่สนามกีฬากลางแจ้ง ในการเข้าชมผู้ชมจะต้องเข้าคิวกันแต่ละคนเริ่มมาเข้าคิวไม่พร้อมกัน อย่างไรก็ตาม เพื่อให้การเข้านั่งชมเป็นไปอย่างสะดวกได้มีการจัดลำดับว่าใครควรจะได้เข้าในสนามกีฬาก่อนโดยพิจารณาจากตำแหน่งที่นั่งและข้อมูลประกอบอื่น ๆ ผู้จัดงานได้ระบุหมายเลขที่ไม่ซ้ำกันให้กับผู้ชมทุกคน คนที่ได้หมายเลขมากจะต้องเข้าอาคารก่อน

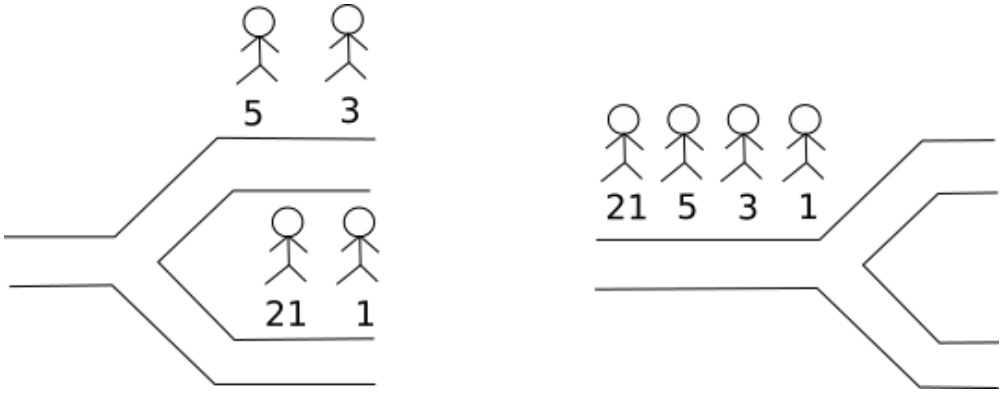
ในยุคที่ไม่มีใครอยากจะดราม่า การจัดแถวคนที่มารอคิวแล้วเป็นคิวใหม่อาจทำให้เกิดเรื่องราวใหญ่โต ผู้จัดการแสดงจึงได้ออกแบบระบบการจัดแถวใหม่ โดยทำเป็นการให้ผู้เข้าชมเดินวน ๆ ผ่านทางที่ต้องมีการแยกแถวเป็นแถวย่อย ๆ จำนวน K แถว แล้วค่อยพาไปรวมกันเหลือแถวเดียว โดยการควบคุมความเร็วการเดินของแต่ละแถวย่อย ผู้จัดงานสามารถเรียงลำดับคนในแถวได้ใหม่

สำหรับการแยกผู้เข้าชมออกเป็น K แถวนั้น จะทำได้โดยให้ผู้เข้าชมเดินเข้ามาตามลำดับจากหัวแถวไปท้ายแถว แล้วเลือกให้คนด้านหน้าสุดแยกไปทางแถวใดแถวหนึ่งจากทั้ง K แถว หลังจากนั้นเลือกแถวให้กับคนถัดไป ไปเรื่อย ๆ จนครบทั้ง N คนในแถว เมื่อจัดแถวแยก K แถวให้กับคนทั้ง N คนแล้ว สามารถรวมแถวกลับเข้ามาได้ โดยการเลือกคนใดคนหนึ่งจากหัวแถวทั้ง K แถว (หากแถวไหนไม่มีคนจะไม่สามารถเลือกหัวแถวนั้นได้) เมื่อเลือกหัวแถวไปเรื่อย ๆ จนครบทั้ง N คนแล้วจะได้การรวมแถวเป็นแถวใหม่แถวเดียว

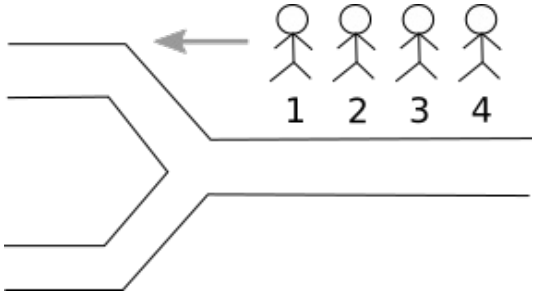
พิจารณาตัวอย่างดังรูปด้านล่างที่ $K = 2$ และมีคนเข้าแถว N คน มีหมายเลขเข้าชมเรียงลำดับจากหัวแถวไปท้ายแถวคือ $5, 21, 3, 1$



สังเกตว่า ถ้าเราแบ่งแถวเป็น 2 แถว ดังรูปด้านล่าง (ซ้าย) จากนั้นเรียกหัวแถวแต่ละแถวย่อยทีละคนให้เดินไปรวมแถวใหญ่ ผู้จัดงานสามารถเรียงลำดับคนใหม่ให้มีลำดับเป็น $21, 5, 3, 1$ ได้ตามต้องการ (รูปขวา)



ในบางกรณีผู้จัดไม่สามารถเรียงคิวให้ได้ตามต้องการทั้งหมดในการแยกและรวมแถวในครั้งเดียว พิจารณาตัวอย่างในภาพต่อไปนี้



งานของคุณคือคำนวณว่าจะต้องเรียงคิวโดยกระบวนการแยกและรวมอย่างน้อยที่สุดกี่รอบ จึงจะทำให้คิวเรียงลำดับตามหมายเลขที่ผู้จัดต้องการ จากมากไปหาน้อย

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
int requeue(vector<int> Q, int K)
```

- ฟังก์ชันจะถูกเรียกเพียงครั้งเดียว
- เวกเตอร์ Q จะมีขนาด N จะเก็บหมายเลขของคนในแถวเรียงตามลำดับในแถว $Q[0]$ คือหมายเลขของคนหัวแถว $Q[N-1]$ คือหมายเลขของคนท้ายแถว
- ฟังก์ชันจะต้องคืนค่าเป็นจำนวนเต็ม แทนจำนวนรอบที่น้อยที่สุดที่ต้องเรียงคิวโดยกระบวนการแยกและรวม โดยมีจำนวนแถวย่อย K แถว

ขอบเขต

- $1 \leq N \leq 2\,500\,000$
- $2 \leq K \leq 10$
- $1 \leq Q[i] \leq 1\,000\,000\,000$ สำหรับทุก $0 \leq i < N$
- Q ไม่มีจำนวนซ้ำกันเลย (pairwise distinct)

ปัญหาย่อย

1. (5 คะแนน) $K = 2$, $N \leq 4$
2. (17 คะแนน) $K = 2$, $N \leq 7$
3. (8 คะแนน) $K = 2$, $N \leq 1\,000$ และรับประกันว่าคำตอบมีค่าไม่เกิน 2
4. (11 คะแนน) $K = 2$, $N \leq 100\,000$ และรับประกันว่าคำตอบมีค่าไม่เกิน 2
5. (27 คะแนน) $K = 2$, $N \leq 1\,000$
6. (8 คะแนน) $K = 2$, $N \leq 100\,000$
7. (18 คะแนน) $N \leq 100\,000$
8. (6 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

จากตัวอย่างข้างต้น ตัวอย่างแรกจะเรียกฟังก์ชันเป็น

```
requeue([5, 21, 3, 1], 2)
```

ฟังก์ชันจะต้องคืนค่า 1 ตัวอย่างนี้เป็นไปตามคำอธิบายในโจทย์

สำหรับตัวอย่างที่สอง จะเรียกฟังก์ชันดังนี้

```
requeue([1, 2, 3, 4], 2)
```

ฟังก์ชันจะต้องคืนค่า 2 สำหรับกรณีนี้ จะสามารถแยกแถว `[1, 2, 3, 4]` ออกเป็น `[1, 3]` และ `[2, 4]` หลังจากนั้นรวมเป็น `[2, 1, 4, 3]` ถือว่าเป็นการเรียงคิวหนึ่งครั้ง ต่อมาแยกแถว `[2, 1, 4, 3]` ออกเป็น `[2, 1]` และ `[4, 3]` หลังจากนั้นรวมเป็น `[4, 3, 2, 1]` จึงนับเป็นการเรียงคิวสองครั้ง

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างอ่านข้อมูลดังนี้

- บรรทัดที่ 1: N, K
- บรรทัดที่ $2, \ldots, N+1$: $Q[i]$ (หมายเลขของคนที่ i)

เกรดเดอร์จะพิมพ์ค่าที่คืนจากฟังก์ชัน `requeue`

ข้อจำกัด

- Time limit: 1 second
- Memory limit: 64 MB