เกมในห้องเรียน

โรงเรียนมัธยมชื่อดังแห่งหนึ่งในทาชเคนต์เฉลิมฉลองวันครบรอบการก่อตั้งด้วยการจัดเกมการแข่งขันระหว่างนักเรียนและครู มีนักเรียน N คนนั่งอยู่ในห้องเรียน โดยแต่ละคนถูกกำหนดหมายเลขตั้งแต่ 0 ถึง $N-1$ นักเรียนทุกคนถือกระดาษคนละหนึ่งแผ่นสำหรับบันทึกอาร์เรย์ของจำนวนเต็ม ซึ่งในตอนเริ่มต้น กระดาษทุกแผ่นยังคงว่างเปล่า ดังนั้นจึงถือว่าบรรจุอาร์เรย์ว่างเอาไว้

นักเรียนกำลังเล่นเกมกับครู M คน ซึ่งเรียงลำดับหมายเลขตั้งแต่ 0 ถึง $M-1$ และผู้อำนวยการ (ผอ.) เกมนี้เล่นทั้งหมด M ขั้นตอน โดยแต่ละขั้นตอนมีหมายเลขตั้งแต่ 0 ถึง $M-1$ เช่นกัน ในขั้นตอนที่ j ครูคนที่ j เดินเข้ามาในห้องเรียน แล้วเกิดเหตุการณ์ตามลำดับดังนี้

- นักเรียนบางคน (หรืออาจไม่มีเลย) ยกมือขึ้น รับประกันได้ว่าในทุกขั้นตอนจะมีนักเรียนอย่างน้อยหนึ่งคนที่**ไม่**ยกมือขึ้น และนักเรียนแต่ละคนสามารถยกมือขึ้นได้**อย่างมากหนึ่งครั้ง**ตลอดทั้งเกม
- ครูจะดูเอกสารที่นักเรียนถืออยู่และอาจ **แก้ไข** เอกสารของนักเรียนที่ **ไม่ได้** ยกมือในขั้นตอนนี้ สำหรับนักเรียนแต่ละคน ครูสามารถแทนที่เนื้อหาในเอกสารของพวกเขาด้วยอาร์เรย์ของจำนวนเต็มชุดใหม่ (หรืออาร์เรย์ว่าง) โดยจำนวนเต็มแต่ละตัวต้องมีค่าตั้งแต่ 0 ถึง 63 รวมหัวท้าย และอาร์เรย์นั้นสามารถมีสมาชิกได้สูงสุดไม่เกิน 63 ตัว
- หลังจากแก้ไขเสร็จแล้ว ครูคนที่ j จะเดินออกจากห้องไป จากนั้นนักเรียนจะแลกเปลี่ยนกระดาษกันตามลำดับการเรียงสับเปลี่ยนลับ P กล่าวคือ นักเรียนคนที่ i ($0 \le i < N$) จะส่งกระดาษของตนเองให้กับนักเรียนคนที่ $P[i]$ โดยที่ $P[0], P[1], \ldots, P[N-1]$ เป็นจำนวนเต็มที่**ไม่ซ้ำกัน** N จำนวน ซึ่งมีค่าตั้งแต่ 0 และ $N-1$ รวมหัวท้าย ค่าของ P นั้น **คงที่** ตลอดทุกขั้นตอนของเกม แต่ครูและ ผอ. จะไม่ทราบค่านี้เลย

เมื่อสิ้นสุดการดำเนินเกมทั้ง M ขั้นตอน และการแลกเปลี่ยนกระดาษตาม P ในรอบสุดท้ายเสร็จสิ้นลง ผอ. จะเดินเข้ามาในห้องเรียน โดย ผอ.จะต้องระบุให้ได้ว่านักเรียน i แต่ละคน ($0 \le i < N$) ยกมือในขั้นตอนที่ j ใด หรือสรุปว่านักเรียน i คนนั้นไม่เคยยกมือเลยตลอดทั้งเกม โดยดูจากเนื้อหาบนกระดาษที่นักเรียนแต่ละคนถืออยู่เท่านั้น

ครูแต่ละคนไม่สามารถติดต่อสื่อสารกับครูคนอื่นหรือ ผอ. ได้ เว้นแต่จะสื่อสารผ่านอาร์เรย์ของจำนวนเต็มที่เขียนลงบนกระดาษซึ่งนักเรียนส่งต่อให้กันเท่านั้น ครูแต่ละคนรู้หมายเลขขั้นตอนที่ตัวเองจะต้องเข้าไปในห้องเรียน

งานของคุณคือ การวางแผนและดำเนินการกลยุทธ์สำหรับครูและ ผอ. เพื่อให้สามารถระบุได้อย่างถูกต้องว่านักเรียนแต่ละคนได้ยกมือขึ้นหรือไม่ และยกมือในขั้นตอนใด คะแนนของคุณจะขึ้นอยู่กับ**ความยาวสูงสุด**ของอาร์เรย์ใด ๆ ที่ถูกเขียนลงบนกระดาษ ยิ่งอาร์เรย์มีความยาวสูงสุดลดลง คะแนนของคุณก็ยิ่งสูงขึ้น

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนสองฟังก์ชัน ฟังก์ชันหนึ่งสำหรับครูและอีกฟังก์ชันหนึ่งสำหรับ ผอ.

ฟังก์ชันสำหรับ**ครู**ที่คุณต้องเขียนเป็นดังนี้

```
std::vector<std::vector<int>> process_step(
    int N, int M, int R,
    std::vector<int> T,
    std::vector<std::vector<int>> A)
```

- N: จำนวนของนักเรียน
- M: จำนวนขั้นตอน ซึ่งคือจำนวนของครูด้วยเช่นกัน
- R: หมายเลขขั้นตอนปัจจุบัน (ตั้งแต่ 0 ถึง $M-1$)
- T: อาร์เรย์ (ที่อาจจะเป็นอาร์เรย์ว่าง) ของหมายเลขของนักเรียนที่ยกมือในขั้นตอนนี้ เรียงตามหมายเลขจากน้อยไปมาก
- A: อาร์เรย์ความยาว N ที่อธิบายถึงกระดาษทุกใบ โดยที่ $A[i]$ $(0 \le i < N)$ คืออาร์เรย์ของจำนวนเต็มในกระดาษที่ถือโดยนักเรียนหมายเลข i ณ ตอนเริ่มต้นของขั้นตอนนี้
- ฟังก์ชันนี้จะถูกเรียกทั้งหมด M ครั้งต่อเกม ในลำดับ $R = 0, 1, \ldots, M-1$

ฟังก์ชันนี้จะต้องคืนอาร์เรย์ B ที่มีความยาว N ซึ่งแสดงถึงกระดาษทั้งหมดหลังจากที่ครูทำการแก้ไขแล้ว โดยแสดงในรูปแบบเดียวกับ A

- กระดาษของสำหรับนักเรียนหมายเลข i ที่ยกมือ (มีตัวเลข i ปรากฏใน T) นั้นจะต้องไม่มีการเปลี่ยนแปลง กล่าวคือ $B[i] = A[i]$
- สำหรับแต่ละ i ที่ $0 \le i < N$ นั้น ความยาวของ $B[i]$ จะต้องไม่เกิน 63 และแต่ละตัวเลขใน $B[i]$ จะต้องมีค่าตั้งแต่ 0 ถึง 63 (รวมหัวท้าย)

ฟังก์ชันสำหรับ **ผอ.** ที่คุณต้องเขียนเป็นดังนี้

```
std::vector<int> determine_steps(
    int N, int M, std::vector<std::vector<int>> A)
```

- N, M: เป็นเช่นเดียวกับของครู
- A: อาร์เรย์ความยาว N โดยที่ $A[i]$ คืออาร์เรย์ของจำนวนเต็มในกระดาษที่อยู่ในมือของนักเรียนหมายเลข i หลังจากจบทั้ง M ขั้นตอนแล้ว
- ฟังก์ชันนี้จะถูกเรียกเพียงหนึ่งครั้งต่อเกม หลังจากการเรียก `process_step` ในครั้งสุดท้าย

ฟังก์ชันนี้จะต้องคืนค่า D ที่มีความยาว N โดยที่เงื่อนไขต่อไปนี้ต้องเป็นจริงสำหรับแต่ละ i ที่ $0 \le i < N$

- $D[i] = j$ ถ้านักเรียนหมายเลข i ยกมือในขั้นตอน j หรือ
- $D[i] = -1$ ถ้านักเรียนหมายเลข i ไม่เคยยกมือเลยตลอดเกม

โปรแกรมของคุณไม่ได้รับอนุญาตให้เก็บหรือส่งข้อมูลใด ๆ ระหว่างการเรียก `process_step` แต่ละครั้ง นอกเหนือไปจากค่าที่คืนมาของการเรียก `process_step` ถ้าหากโปรแกรมของคุณพยายามกระทำการดังกล่าว คะแนนของคุณในระบบ CMS อาจจะไม่ถูกต้องและอาจจะถูกลดลงหลังจากจบการแข่งขัน โปรดทราบว่าเกรดเดอร์อาจจะทำการเรียกโปรแกรมของคุณหลาย ๆ ครั้ง พร้อม ๆ กันในหนึ่งช่วงเวลาก็เป็นได้ ซึ่งหมายความว่าการเรียก `process_step` และ `determine_steps` จากเกมเดียวกันอาจจะถูกรันในโปรแกรมที่แตกต่างกัน และการเรียก `process_step` และ `determine_steps` จากเกมที่แตกต่างกันอาจจะถูกรันในโปรแกรมเดียวกันก็เป็นได้ ข้อมูลทดสอบแต่ละชุดมีเกมได้มากถึง 5 เกมการแข่งขัน

ข้อกำหนด

- $2 \le N \le 63$
- $1 \le M \le 63$
- นักเรียนแต่ละคนยกมือ **ไม่เกินหนึ่งครั้ง** ในระหว่างเกมทั้งเกมหนึ่ง กล่าวคือ สำหรับนักเรียนหมายเลข i แต่ละคนนั้น จะมีอย่างมากเพียงหนึ่งขั้นตอน j ที่นักเรียน i ยกมือ
- ในแต่ละขั้นตอนจะมีนักเรียนอย่างน้อยหนึ่งคนที่**ไม่**ยกมือ

ปัญหาย่อยและการให้คะแนน

ปัญหาย่อย	คะแนน	เงื่อนไขเพิ่มเติม
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ สำหรับ $0 \le i \le N - 1$.
4	25	มีนักเรียนยกมืออย่างมากแค่ 1 คนในแต่ละขั้นตอน
5	56	ไม่มีเงื่อนไขเพิ่มเติม

ในแต่ละชุดข้อมูลทดสอบ คะแนนของคุณจะเป็น 0 (ได้ผลลัพธ์เป็น `Output isn't correct` ในระบบ CMS) หากว่าค่าที่คืนมาจากการเรียก `process_step` ไม่เป็นไปตามข้อกำหนด หรือ ค่าที่คืนมาจากการเรียก `determine_steps` นั้นไม่ถูกต้อง

หากการคืนค่าทั้งหมดเป็นไปอย่างถูกต้องแล้ว การคิดคะแนนของแต่ละชุดข้อมูลทดสอบที่อยู่ในปัญหาย่อยที่มีคะแนนเป็น S จะเป็นดังนี้ ให้ C เป็นค่าความยาวมากสุดของอาร์เรย์ B **ใด ๆ** ที่คืนมาจาก `process_step` คะแนนของชุดข้อมูลทดสอบดังกล่าวคือ $S \cdot X$ โดยที่ X จะขึ้นอยู่กับค่า C ตามตารางข้างล่างนี้

เงื่อนไข	X
$C \le 2$	1.00
$C = 3$	0.75
$C = 4$	0.55
$5 \le C \le 13$	$0.50 - 0.03 \cdot (C - 5)$
$14 \le C \le 63$	$0.19 \cdot \frac{64-C}{64-14} + 0.04$

ตัวอย่าง

ให้พิจารณากรณีที่มีนักเรียน $N = 4$ คน มีครู $M = 2$ คน และการเรียงสับเปลี่ยน $P = [0, 3, 1, 2]$

ในตอนเริ่มต้น กระดาษทุกใบว่างเปล่า

เกรดเดอร์เริ่มต้นโดยเรียก

```
process_step(4, 2, 0, [1], [[], [], [], []])
```

นักเรียนหมายเลข 1 ยกมือ ทำให้ครูแก้ไขกระดาษของนักเรียนคนนั้นไม่ได้ ครูหมายเลข 0 เลือกที่จะเขียน $[10]$ ลงในกระดาษของนักเรียนหมายเลข 0 เขียน $[1, 63, 4]$ ลงในกระดาษของนักเรียนหมายเลข 3 และปล่อยให้กระดาษของนักเรียนหมายเลข 2 ว่างเปล่า ฟังก์ชันนี้จะต้องคืนค่า $B = [[10], [], [], [1, 63, 4]]$ เพื่อระบุถึงการทำงานของครูตามข้างต้น

หลังจากที่ครูหมายเลข 0 ออกจากห้อง นักเรียนจะสลับกระดาษกันตามค่า P หลังจากการสลับกระดาษแล้ว $A = [[10], [], [1, 63, 4], []]$

หลังจากนั้นเกรดเดอร์เรียก

```
process_step(4, 2, 1, [0,3], [[10], [], [1,63,4], []])
```

นักเรียนหมายเลข 0 และ 3 ยกมือ ทำให้ครูแก้ไขกระดาษของนักเรียนทั้งสองคนนั้นไม่ได้ ครูหมายเลข 1 เลือกที่จะเขียน $[0, 40]$ ลงในกระดาษของนักเรียนหมายเลข 1 และเขียน $[1, 50]$ ลงในกระดาษของนักเรียนหมายเลข 2 ฟังก์ชันนี้จะต้องคืนค่า $B = [[10], [0, 40], [1, 50], []]$ เพื่อระบุถึงการทำงานของครูตามข้างต้น

หลังจากที่ครูหมายเลข 1 ออกจากห้อง นักเรียนจะสลับกระดาษกันตามค่า P หลังจากการสลับกระดาษแล้ว $A = [[10], [1, 50], [], [0, 40]]$

ในขั้นตอนสุดท้ายเกรดเดอร์จะเรียก

```
determine_steps(4, 2, [[10], [1,50], [], [0,40]])
```

ฟังก์ชันนี้จะต้องคืนค่าอาร์เรย์ $D = [1, 0, -1, 1]$ เนื่องจากนักเรียนหมายเลข 0 และ 3 ยกมือในขั้นตอน 1 และนักเรียนหมายเลข 1 ยกมือในขั้นตอนที่ 0 และนักเรียนหมายเลข 2 ไม่เคยยกมือเลย ในตัวอย่างนี้ $C = 3$

เกรดเดอร์ตัวอย่าง

รูปแบบข้อมูลนำเข้า:

```
N M
Q[0] Q[1] ... Q[N-1]
P[0] P[1] ... P[N-1]
```

Q คืออาร์เรย์ความยาว N โดยที่ $Q[i] = j$ ถ้าหากนักเรียนหมายเลข i ยกมือในขั้นตอน j และ $Q[i] = -1$ ถ้าหากนักเรียนหมายเลข i ไม่เคยยกมือเลยตลอดเกม

รูปแบบข้อมูลส่งออก:

หลังจากการเรียก `process_step` แต่ละครั้ง เกรดเดอร์ตัวอย่างจะแสดงข้อมูลในกระดาษทั้งหมดออกมา ให้ K คือความยาวของ B และให้ $L[i]$ คือความยาวของ $B[i]$ (สำหรับ $0 \leq i < K$) เกรดเดอร์ตัวอย่างจะพิมพ์ B ในรูปแบบต่อไปนี้ **(และตามด้วยบรรทัดว่างหนึ่งบรรทัด)**

```
K
L[0] B[0][0] B[0][1] ... B[0][L[0]-1]
L[1] B[1][0] B[1][1] ... B[1][L[1]-1]
:
L[K-1] B[K-1][0] B[K-1][1] ... B[K-1][L[K-1]-1]
```

หลังจากนั้นเกรดเดอร์ตัวอย่างจะสลับกระดาษตามการเรียงสับเปลี่ยน P ถ้าหาก $K \neq N$ แล้ว เกรดเดอร์ตัวอย่างจะแสดงข้อความระบุข้อผิดพลาดแล้วหยุดทำงาน

เกรดเดอร์ตัวอย่างจะแสดงผลต่อไปนี้หลังจากการเรียก `determine_steps`

```
C H
D[0] D[1] ... D[H-1]
```

ให้ H คือความยาวของอาร์เรย์ D ที่คืนมาจาก `determine_steps`