

## Bubble

[ Time limit : 1s ] [ Memory limit : 64 MB ]

Bubble sort algorithm ถือได้ว่าเป็น algorithm พื้นฐานในการเรียงข้อมูลโดยซึ่งสมมุติให้ข้อมูลเป็นอาร์เรย์ A ขนาด n (แต่เพื่อความเข้าใจที่ตรงกันอาร์เรย์นี้เป็นแบบ 1-index หมายความว่าตัวแรกจะเริ่มจากช่องที่ 1) โดยสิ่งที่ bubble sort ทำในแต่ละรอบคือ

```
// written in c++
for (int i = 1; i < n; ++i) {
    if (A[i] > A[i+1]) {
        swap(A[i], A[i+1]); // สลับค่าของ i กับของ i+1
    }
}
```

สังเกตว่าถ้าเกิดเราทำซ้ำโค้ดนี้  $n-1$  รอบตอนจบอาร์เรย์ A จะเรียงจากน้อยไปมากอย่างถูกต้อง

แต่โจทย์ข้อนี้จะมีความซับซ้อนอยู่ แทนที่เราจะทำซ้ำโค้ดนี้  $n-1$  รอบสมมุติว่าเราจะทำเพียงแค่ k รอบเท่านั้นแล้วถามว่าตัวเลข x นั้นจะอยู่ที่ตำแหน่งใด

### ข้อมูลนำเข้า

บรรทัดแรก ค่า n กับ q (ขนาดอาร์เรย์ A และจำนวนคำถามทั้งหมด)

บรรทัดที่สอง เป็นเลข n ตัว ค่าของ  $A[1], A[2], A[3], \dots, A[n]$  (เพื่อความง่ายเลขทุกตัวจะไม่เท่ากัน และมีค่าตั้งแต่ 1 ถึง n เท่านั้น หรืออีกนัยหนึ่ง A เป็น permutation ของเลข 1 ถึง n นั่นเอง)

บรรทัดที่ 3 ถึง  $q+2$ : แต่ละบรรทัดมีค่า k กับ x เป็นคำถามว่าถ้าเกิดเราทำโค้ดด้านบน k รอบแล้วตัวเลข x นั้นจะอยู่ที่ตำแหน่งใด ( $0 \leq k < n, 1 \leq x \leq n$ )

### ข้อมูลส่งออก

มีทั้งหมด q บรรทัด โดยแต่ละบรรทัดคือคำตอบของคำถามที่ได้ถามไป

### ตัวอย่าง

Input	Output
5 5	4
4 5 3 1 2	5
0 1	1
1 5	1
2 3	4
1 4	



2 4	
-----	--

### ข้อมูลเพิ่มเติมเกี่ยวกับชุดข้อมูลทดสอบ

40% ของชุดข้อมูลทดสอบทั้งหมดมี  $N, Q \leq 2,000$

100% ของชุดข้อมูลทดสอบทั้งหมด  $N, Q \leq 100,000$

### คำอธิบายตัวอย่าง

ตำแหน่ง	1	2	3	4	5
รอบที่ 0	4	5	3	1	2
รอบที่ 1	4	3	1	2	5
รอบที่ 2	3	1	2	4	5
รอบที่ 3	1	2	3	4	5
รอบที่ 4	1	2	3	4	5

\*ตัวสีเขียวคือคำถามตัวเลขที่ถูกถาม

