Problem H. Kuroni and the Sharing of the VNOI Cup Problem Setting Process

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Appearing in today's live stream session of the `VNOI` Cup are coordinator *Kuroni* and *RR* – former competitive programmer, who have reached the ICPC World Finals twice and is one of the two Vietnamese contestants who made it to the finals of the Facebook Hacker Cup. Still passionate about competitive programming, *RR* is also the owner of the Vietnamese fan page *Code cùng RR*, which has nearly 10 000 followers and features discussions on competitive programming.

Apart from monitoring and commenting on the solutions of the contestants, *RR* and *Kuroni* also discuss and share their experiences in the problem-setting process. *Kuroni* has many interesting stories about the contest. For example, one member of the problem setters team accidentally reinvented the "Slope trick" technique at some point. Or the story of *Kuroni* having to sadly reject some very good problems simply because they had similar ideas to problems from competitions many years ago. To ensure high-quality and consistent problem statements, all the problems are authored by a single member of the problem setters team...

After a conversation, *RR* has a question:

– With the investment in problem quality to achieve such high standards, can you share with me and the audience the process of problem selection?

– Sure! Since the problem setters team consists of many members, there are a large number of proposed problems. So the challenge for us is not creating problems but rather selecting which ones to include. For example, for the first problem in the set, we proposed m problems, numbered from 1 to m. It is worth noting that the number of proposed problems m is an **odd** number. Each problem i has a difficulty level of a_i. Since we can only choose one problem, we need to eliminate some of the problems. To do this, we perform the following steps until only one problem remains:

- Choose an index i $(1 < i < m)$.
- Consider the three consecutive problems with difficulties a_{i-1}, a_i, and a_{i+1}.
- Remove the **easiest** and the **hardest** problems among the three. If there is more than one problem with the same minimum or maximum difficulty, remove any one of them.
- Re-number the problems from 1 to $m - 2$ according to their order. The number of problems m decreases by 2.

With this approach, we ensure that the problems that are too easy or too hard are eliminated, and ultimately only one problem remains. Since this is the finals, to ensure a challenging competition for the contestants, we try to perform the above steps to find the **maximum possible** difficulty level for the problem.

– Oh, it sounds very interesting! But it seems quite complicated. Now, for example, if you have some problems with difficulty levels $a = [7, 8, 3, 1]$, how would you proceed?

– Oh, the number of problems m needs to be **odd**!

– Ah, right! So let's take $a = [7, 8, 3, 1, 5]$. How would you find the suitable problem using the elimination process?

– Well. First, I would choose $i = 4$ and consider the three problems with difficulties $[3, 1, 5]$. The problems with difficulties 1 and 5 would be removed, so we would have the list $a = [7, 8, 3]$. I would then perform this operation with the remaining 3 elements, resulting in $a = [7]$. Therefore, the problem we choose would have a difficulty level of 7. It can be seen that this is the **maximum** difficulty level that can be found, as there is certainly no way to filter out a problem with a difficulty level of 8.

– Wow, that's amazing!

Fascinated by the problem selection process of this year's VNOI Cup, *RR* wants to continue asking similar questions. *RR* has taken n problems from his personal computer with difficulty levels $b_1, b_2, \ldots, b_n$. Instead of asking about the difficulty level of all the problems, *RR* will ask q questions. The i-th question consists of two numbers (l_i, r_i) $(1 \le l_i \le r_i \le n,\ l_i \equiv r_i \pmod 2)$. *RR* wants to know the difficulty level of the problem that *Kuroni* will choose in the case of $a = [b_{l_i}, b_{l_i+1}, \ldots, b_{r_i}]$.

Knowing that *RR* wants to test the speed and accuracy of the problem setters team, *Kuroni* is ready to answer the questions. However, due to the large number of questions and *Kuroni* being in the live stream session, the use of a computer is limited. Given the list of difficulty levels of *RR*'s problems and the list of questions, help *Kuroni* find the problem with the *maximum* difficulty level using the elimination process described above.

Input

The first line contains two integers n and q $(1 \le n, q \le 100\,000)$ — the number of problems and questions of *RR*.

The next line contains n integers $b_1, b_2, \ldots, b_n$ $(0 \le b_i \le 10^9)$ — the difficulty levels of RR's problems.

The i-th line among the q following lines contains two integers l_i and r_i $(1 \le l_i \le r_i \le n,\ r_i \equiv l_i \pmod 2)$ — the parameters for the i-th question of RR.

Output

Print q lines. The i-th line is the *maximum* difficulty level of the problem that *Kuroni* will choose when using the elimination operation described in the problem statement.

Scoring

Subtask	Score	Constraints
1	500	$n \le 20,\ q \le 1000$
2	1500	$n, q \le 1000$
3	1500	No additional constraints
Total	3500	

Example

standard input	standard output

7 5
0 7 8 3 1 5 9
1 7
2 6
1 3
3 7
3 3
	7
	7
	7
	8
	8

Note

In the first query, we need to find the answer for the problem sequence $[0, 7, 8, 3, 1, 5, 9]$. We can obtain the problem with difficulty level 7 as follows:

- Choose index $i = 5$ and remove the problems with difficulty levels 1 and 5. The sequence becomes

$$[0, 7, 8, \underline{3, 1, 5}, 9] \rightarrow [0, 7, 8, \underline{3}, 9]$$

- Choose index $i = 2$ and remove the problems with difficulty levels 0 and 8. The sequence becomes

$$[\underline{0, 7, 8}, 3, 9] \to [\underline{7}, 3, 9]$$

- Choose index $i = 2$ and remove the problems with difficulty levels 3 and 9. The sequence becomes

$$[\underline{7, 3, 9}] \to [7]$$

Since there is no sequence of operations that can result in a problem with difficulty level greater than 7, 7 is the maximum difficulty level of the problem that *Kuroni* will choose.

In the second query, we need to find the answer for the problem sequence $[7, 8, 3, 1, 5]$. The appropriate operations to find the problem have been explained by *Kuroni* in the problem statement.