

DMPG '16 G6 - SAO Queries

Time limit: 3.5s **Memory limit:** 768M

Kirito has an array of size N , indexed from 0 to $N - 1$. There are Q queries he wants you to perform on this array. Each query is a Squared Appearances Online (SAO) query on a subarray of the array. The SAO query is computed as follows: for each element x in the subarray, let $count(x)$ be the number of occurrences of x in the subarray. The result of the SAO query is the sum of $count(x)$ for each x in the subarray when taken modulo 2^{18} (if an element x appears more than once in the array, you should add $count(x)$ for each time it appears in the array).

You may think that this problem is pretty easy so far, but remember the \oplus in SAO query: you should perform these queries in an online manner. To enforce this condition, the input will be encrypted (see the input specification for more details).

Input Specification

The first line of input will contain N and Q .

The second line of input will contain N space separated nonnegative integers, the elements of the array.

The next Q lines of input will each contain an encrypted query pair $x y$: you should decode them by finding

$l = x \oplus lastAns$ and $r = y \oplus lastAns$, where $lastAns$ is the answer to the previous query (or 0 if it's the first query).

Then you should output the result of the SAO query on the subarray from l to r inclusive (0-based index). \oplus denotes the bitwise XOR operation. It is guaranteed that l and r will represent a valid subarray after decryption.

Output Specification

For each SAO query, output the result on a new line. Remember that the result should be taken modulo 2^{18} and that $lastAns$ will be equal to exactly the value you printed last (so, it will also be between 0 and $2^{18} - 1$).

Constraints

For all subtasks:

$$1 \leq N, Q \leq 262\,144$$

All other integers in the input will be between 0 and 262 143, inclusive.

Subtask 1 [5%]

$$1 \leq N, Q \leq 1\,000$$

Subtask 2 [55%]

$$1 \leq N, Q \leq 65\,536$$

Subtask 3 [40%]

No additional constraints.

Sample Input

```
5 5
0 1 2 1 2
0 4
9 10
7 5
7 1
6 1
```

Sample Input (not encrypted)

For your convenience, we provide a version of the sample input that is **NOT** encrypted. Remember, all of the real test files will be encrypted (like the input above).

```
5 5
0 1 2 1 2
0 4
0 3
1 3
2 4
3 4
```

Sample Output

```
9
6
5
5
2
```