

8518 Sum of xor sum

Song Zha Zha has an 1-indexed array A . Li Zha Zha has Q queries. Each query has two integers L, R , asking Ran Zha Zha to do the following thing.

First, find all subintervals of $[L, R]$. Then calculate their **xor** sum.

For example. $A = \{1, 2, 3\}$, $L = 1, R = 3$.

All subintervals of $[1, 3]$ are $[1, 1], [2, 2], [3, 3], [1, 2], [2, 3], [3, 3]$.

Their **xor** sum = $1 + 2 + 3 + 1 \text{ xor } 2 + 2 \text{ xor } 3 + 1 \text{ xor } 2 \text{ xor } 3$.

Xor means exclusive or. (\wedge in C++ or Java)

Input

The input contains multiple test cases.

First line contains an integer T ($1 \leq T \leq 10$), which is the number of test cases.

In each test case:

- The first line contains two integers N, Q . ($1 \leq N, Q \leq 100000$). N is the length of the array A .
- Then one line contains N integer indicating $A[i]$. ($1 \leq i \leq N, 0 \leq A[i] \leq 1000000$)
- Then Q lines follow. Each line two integer L, R , means that there's a query $[L, R]$. ($1 \leq L \leq R \leq N$)

Output

For each query, print the answer mod 1000000007.

Sample Input

```
1
3 1
1 2 3
1 3
```

Sample Output

```
10
```