Problem G

Sum of Xor Sums

Song Zha Zha has a sequence *A* of length *N*. Li Zha Zha has *Q* queries. Each query provides two integers *L* and *R* indicating a closed interval *[L,R]*. Ran Zha Zha lists all subintervals of *[L,R]*. For each subinterval *[s,t]*, he calculates the *xor* sums of the *s*-th element to the *t*-th element in A. Then he adds all the xor sums up and the the whole sum is what he needs. Here '*xor*' means the 'Exclusive Or', and it corresponds to the operation '^' in C++ and Java.

Consider the sequence *A=[1,2,3]*, for instance, *L=1* and *R=3* indicating the interval *[1,3]*. Here we list all subintervals of *[1,3]*:

[1,1], [2,2], [3,3], [1,2], [2,3] and [3,3].

Their xor sums are *1, 2, 3, 1 xor 2, 2 xor 3, and 1 xor 2 xor 3* respectively. The sum of all *xor* sums above is *1+2+3+1 xor 2+2 xor 3+1 xor 2 xor 3*.

In this problem, you need to do what Ran Zha Zha do for several queries, calculating the sum of the xor sums for all subintervals.

Input

The input contains several test cases. The first line contains an integer *T (1<=T<=10)*, which is the number of test cases.

In each test case, the first line contains two integers *N* and *Q (1<=N, Q<=100000)*, where *N* is the length of the array *A* and *Q* is the number of queries. The second line contains *N* integers *A[1]* to *A[N]* indicating the sequence (*0<=A[i]<=1000000* for each *i* in *[1,N]*). Then *Q* lines follow, each of which describes a query with two integer *L* and *R* indicating the interval *[L,R]* where *1<=L<=R<=N*. The sum of *N* and *Q* in all test cases is less than *400000*.

Output

For each query, print the answer mod *1000000007*.

Sample Input	Sample Output
1 3 1 1 2 3 1 3	10