```
/*  SPOJ - SUBXOR - SubXor
Given an array of N positive integers you have to print the number of subarrays
whose XOR is less than K. Subarrays are defined as a sequence of continuous
elements Ai,Ai+1,...,Aj. XOR of a subarray is defined as Ai^Ai+1^...^Aj.

First line contains T, the number of test cases. Each of the test case consists
of N and K in one line, followed by N space separated integers in next line.

1 ≤ T ≤ 10; 1 ≤ N ≤ 10^5; 1 ≤ A[i] ≤ 10^5; 1 ≤ K ≤ 10^6
*/
#include <bits/stdc++.h>
using namespace std;
#define ll long long
const int MAXN = 100005;
int a[MAXN], cum[MAXN];
struct Node{
    int cnt;
    Node *next[2];
    Node(){
        cnt = 0;
        next[0] = next[1] = NULL;
    }
}*root;

string DecToBin(int n,int msz){
    string s;
    while(n!=0){
        s += (n&1) + '0';
        n >>= 1;
    }
    while((int)s.size()<msz) s += '0';
    reverse(s.begin(), s.end());
    return s;
}

void Insert(string s){
    Node *cur = root;
    for(int i=0; i<s.size(); i++){
        int id = s[i] - '0';
        if(cur->next[id] == NULL) cur->next[id] = new Node();
        cur = cur->next[id];
        cur->cnt++;
    }
}

int query(string s, string k){
    Node *cur = root;
    int cnt = 0;

    for(int i=0; i<s.size(); i++){
        int ids = s[i] - '0';
        int idk = k[i] - '0';

        if(ids==0 && idk==0){
            if(cur->next[0]==NULL)return cnt;
            cur = cur->next[0];
        }
        else if(ids==0 && idk==1){
            if(cur->next[0]!=NULL) cnt += cur->next[0]->cnt;
            if(cur->next[1]==NULL)return cnt;
            cur = cur->next[1];
        }
        else if(ids==1 && idk==0){
            if(cur->next[1]==NULL)return cnt;
            cur = cur->next[1];
        }
        else{
            if(cur->next[1]!=NULL) cnt += cur->next[1]->cnt;
            if(cur->next[0]==NULL)return cnt;
            cur = cur->next[0];
        }
    }

    return cnt;
}

void del(Node *cur){
    for(int i=0; i<2; i++){
        if(cur->next[i]) del(cur->next[i]);
    }
    delete(cur);
}

int main() {
    int tt; scanf("%d",&tt);
    for(int ks=1; ks<=tt; ks++){
        int n,k; scanf("%d%d",&n,&k);
        int maxx = 0;
        for(int i=1; i<=n; i++){
            scanf("%d",&a[i]);
            maxx = max(maxx, a[i]);
        }

        int msz;
        if(maxx == 0) msz = 1;
        else msz = floor(log2(maxx)) + 1;
        int msk =  floor(log2(k)) + 1;
        msz = max(msz,msk);

        string s, kk;
        s  = DecToBin(0, msz);
        kk = DecToBin(k, msz);

        root = new Node();
        Insert(s);

        ll ans = 0;
        int p = 0;
        for(int i=1; i<=n; i++){
            int q = p^a[i];
            s = DecToBin(q, msz);

            ans += query(s, kk);
            Insert(s);
            p = q;
        }
        printf("%lld\n",ans);

        del(root);
    }
    return 0;
}

INPUT        OUTPUT
1            3
5 2
4 1 3 2 7
Only subarrays satisfying the conditions are [1], [1,3,2] and [3,2].
```