ABC313 D - Odd or Even

交互题，交互题不想flush就直接在每次输出的时候多输出单独一行cout<<endl;就行。

题意：

有一个由01组成的长n的序列。最多可以问n个问题。每个问题必须是问k个位置的异或值是奇数还是偶数。当可以唯一确定序列时输出序列。

题解：

首先，我们肯定是n个问题要问完的。现在就是怎么问的问题。每次必须都要问k个数，所以我们可以先问前k-1个数每次加后面的一个数作为一个整体。只要最后可以确定前k-1个数的奇偶性就可以确定后面的数了。

现在的问题是，如何确定前k-1个数的奇偶性？

按照我们最初的想法如果k个数全是同一个数就可以直接确定了，但题目限制了下标不能相同，如果我们能把k个数中的第k个数和前面的k-1个数中的数有一个重复，再让k-1个数中的每个数重复一次然后再把这k-1次的结果相加，就得到了相当于k次的前k-1个数的异或。题目中保证k是奇数相当于直接得到了前k-1个数的异或。将每个重复结果再异或上一次前k-1个数，就相当于得到每个数的值了。

前k个数

前k-1个数带一个重复

每个数重复一次

K次前k-1个数异或

再异或上一次前k-1个数

每个数重复一次

那么如何制造重复呢？将前k个数中其中第i个换成后面的第c个数，和前面已经求出的前k-1个数和第c个数的结果再做一次异或。就可以得到第的i个数和第k个数的异或值。在将前k个数的异或值异或上，就相当于将第k个数变成第i个数了。

前k个数

将前k-1个数中其中一个换成后面的数

相异或，相当于求

将前k个数异或上

相当于将第k个数变成第i个数了。

题解代码：

#include <bits/stdc++.h>

using namespace std;

#define int long long

#define rep(a, b, c) for (int a = b; a <= c; a++)

#define per(a, b, c) for (int a = b; a >= c; a--)

typedef long long ll;

typedef pair<int, int> pii;

const ll INF = (ll)9e18;

const int inf = 0x7fffffff;

const int maxn = 1e6 + 10;

const int mod = 1e9 + 7;

const int mod2 = 998244353;

int a[maxn];

int query(vector<int> a) {

    sort(a.begin(), a.end());

    cout << "?";

    for (auto x : a) {

        cout << " " << x + 1;

    }

    cout << endl;

    int x;

    cin >> x;

    return x;

};

void print(const vector<int>& a) {

    cout << "!";

    for (auto x : a) {

        cout << " " << x;

    }

    cout << endl;

    exit(0);

}

void solve() {

    //

    int n, k;

    cin >> n >> k;

    vector<int> ans(n);

    vector<int> ask(k);

    iota(ask.begin(), ask.end(), 0);

    for (int i = k - 1; i < n; i++) {

        ask[k - 1] = i;

        ans[i] = query(ask);

    }

    for (int i = 0; i < k - 1; i++) {

        ask[i] = n - 2;

        ans[i] = ans[n - 2] ^ (query(ask) != ans[n - 1]);

        ask[i] = i;

    }

    if (accumulate(ans.begin(), ans.begin() + k, 0) % 2 != ans[k - 1]) {

        for (auto& x : ans) {

            x ^= 1;

        }

    }

    print(ans);

}

signed main() {

    ios::sync\_with\_stdio(0);  // can't use printf & scanf & read()

    cin.tie(0);               // untie cin,cout

    cout.tie(0);              // untie cin,cout

    int T = 1;

    // cin >> T;

    while (T--) {

        solve();

    }

    return 0;

}