B. Odd Sum Segments

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array a consisting of n integers $a_1, a_2, \ldots, a_n$. You want to split it into exactly k **non-empty non-intersecting subsegments** such that each subsegment has odd sum (i. e. for each subsegment, the sum of all elements that belong to this subsegment is odd). It is impossible to rearrange (shuffle) the elements of a given array. Each of the n elements of the array a must belong to exactly one of the k subsegments.

Let's see some examples of dividing the array of length 5 into 3 subsegments (not necessarily with odd sums): $[1, 2, 3, 4, 5]$ is the initial array, then all possible ways to divide it into 3 non-empty non-intersecting subsegments are described below:

- $[1], [2], [3, 4, 5]$;
- $[1], [2, 3], [4, 5]$;
- $[1], [2, 3, 4], [5]$;
- $[1, 2], [3], [4, 5]$;
- $[1, 2], [3, 4], [5]$;
- $[1, 2, 3], [4], [5]$.

Of course, it can be impossible to divide the initial array into exactly k subsegments in such a way that each of them will have odd sum of elements. In this case print "NO". Otherwise, print "YES" and **any** possible division of the array. See the output format for the detailed explanation.

You have to answer q independent queries.

Input

The first line contains one integer q ($1 \le q \le 2 \cdot 10^5$) — the number of queries. Then q queries follow.

The first line of the query contains two integers n and k ($1 \le k \le n \le 2 \cdot 10^5$) — the number of elements in the array and the number of subsegments, respectively.

The second line of the query contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), where a_i is the i -th element of a .

It is guaranteed that the sum of n over all queries does not exceed $2 \cdot 10^5$ ($\sum n \le 2 \cdot 10^5$).

Output

For each query, print the answer to it. If it is impossible to divide the initial array into exactly k subsegments in such a way that each of them will have odd sum of elements, print "NO" in the first line. Otherwise, print "YES" in the first line and **any** possible division of the array in the second line. The division can be represented as k integers r_1 , r_2 , ..., r_k such that $1 \le r_1 < r_2 < \cdots < r_k = n$, where r_j is the right border of the j -th segment (the index of the last element that belongs to the j -th segment), so the array is divided into subsegments $[1; r_1], [r_1 + 1; r_2], [r_2 + 1, r_3], \ldots, [r_{k-1} + 1, n]$. **Note that r_k is always n but you should print it anyway**.

Example

input Copy

```
3
5 3
7 18 3 14 1
```

Codeforces Round #575 (Div. 3)
Finished
Practice

→ **Virtual participation**

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ **Practice**

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ **Clone Contest to Mashup**

You can clone this contest to a mashup.

Clone Contest

→ **Submit?**

Language: GNU G++11 5.1.0

Choose file: Browse...

Submit

→ **Problem tags**

constructive algorithms | math

No tag edit access

→ **Contest materials**

- Announcement #1 (en)
- Announcement #2 (ru)
- Tutorial #1 (en)
- Tutorial #2 (ru)

```
5 4
1 2 3 4 5
6 2
1 2 8 4 10 2
```

output Copy

```
YES
1 3 5
NO
NO
```

- Tutorial #3 (en)