B. Medians

time limit per test: 1 second
memory limit per test: 256 megabytes

You are given an array $a = [1, 2, \ldots, n]$, where n is **odd**, and an integer k.

Your task is to choose an **odd** positive integer m and to split a into m subarrays[†] $b_1, b_2, \ldots, b_m$ such that:

- Each element of the array a belongs to exactly one subarray.
- For all $1 \le i \le m$, $|b_i|$ is **odd**, i.e., the length of each subarray is odd.
- $\text{median}([\text{median}(b_1), \text{median}(b_2), \ldots, \text{median}(b_m)]) = k$, i.e., the median[‡] of the array of medians of all subarrays must equal k. $\text{median}(c)$ denotes the median of the array c.

[†] A subarray of the array a of length n is the array $[a_l, a_{l+1}, \ldots, a_r]$ for some integers $1 \le l \le r \le n$.

[‡] A median of the array of odd length is the middle element after the array is sorted in non-decreasing order. For example: $\text{median}([1, 2, 5, 4, 3]) = 3$, $\text{median}([3, 2, 1]) = 2$, $\text{median}([2, 1, 2, 1, 2, 2, 2]) = 2$.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 5000$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n and k ($1 \le k \le n < 2 \cdot 10^5$, n is **odd**) — the length of array a and the desired median of the array of medians of all subarrays.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case:

- If there is no suitable partition, output -1 in a single line.
- Otherwise, in the first line, output an **odd** integer m ($1 \le m \le n$), and in the second line, output m distinct integers $p_1, p_2, p_3, \ldots, p_m$ ($1 = p_1 < p_2 < p_3 < \ldots < p_m \le n$) — denoting the left borders of each subarray.

In detail, for a valid answer $[p_1, p_2, \ldots, p_m]$:

- $b_1 = [a_{p_1}, a_{p_1+1}, \ldots, a_{p_2-1}]$
- $b_2 = [a_{p_2}, a_{p_2+1}, \ldots, a_{p_3-1}]$
- $\ldots$
- $b_m = [a_{p_m}, a_{p_m+1}, \ldots, a_n]$.

If there are multiple solutions, you can output any of them.

Example

input Copy

```
4
1 1
3 2
3 3
15 8
```

output Copy

```
1
1
3
```

```
1 2 3
-1
5
1 4 7 10 13
```

Note

In the first test case, the given partition has $m = 1$ and $b_1 = [1]$. It is obvious that $\text{median}([\text{median}([1])]) = \text{median}([1]) = 1$.

In the second test case, the given partition has $m = 3$ and:

- $b_1 = [1]$
- $b_2 = [2]$
- $b_3 = [3]$

Therefore, $\text{median}([\text{median}([1]), \text{median}([2]), \text{median}([3])]) = \text{median}([1, 2, 3]) = 2$.

In the third test case, there is no valid partition for $k = 3$.

In the fourth test case, the given partition has $m = 5$ and:

- $b_1 = [1, 2, 3]$
- $b_2 = [4, 5, 6]$
- $b_3 = [7, 8, 9]$
- $b_4 = [10, 11, 12]$
- $b_5 = [13, 14, 15]$

Therefore, $\text{median}([\text{median}([1, 2, 3]), \text{median}([4, 5, 6]), \text{median}([7, 8, 9]), \text{median}([10, 11, 12]), \text{median}([13, 14, 15])]) = \text{median}([2, 5, 8, 11, 14]) = 8$.