

Problem L. Permutations and Cycles (Maximum Version)

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

For a given x , a permutation of size n is called *good* if for each $1 \leq i < n$ the condition $p_i + p_{i+1} \leq x$ holds. Find any good permutation with the **maximum** number of cycles.

A permutation of size n is a sequence of n distinct integers from 1 to n .

A cycle of a permutation p is a sequence of indices i_1, i_2, \dots, i_k such that $p_{i_1} = i_2, p_{i_2} = i_3, \dots, p_{i_k} = i_1$. The cycles obtained by a cyclic shifting of the sequence are considered to be the same.

Input

The first line contains an integer t ($1 \leq t \leq 2 \cdot 10^5$), the number of test cases. The test cases follow.

Each test case is given on a line with two integers n ($2 \leq n \leq 2 \cdot 10^5$) and x ($n + 1 \leq x \leq 2 \cdot n - 1$). These constraints guarantee that at least one *good* permutation exists.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print two lines. The first one should contain the maximum number of cycles in a good permutation of length n . The second line should consist of n integers: the permutation itself. If multiple such permutations exist, print any one of them.

Example

<i>standard input</i>	<i>standard output</i>
3	2
2 3	1 2
3 4	2
3 5	2 1 3
	3
	1 2 3