Problem F. Classic: Classical Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

Deja vu!
I've just been in this place before.

— Dave Rodgers, *Deja Vu*

This is a classical problem, so let's make it quick.

Given a set S of n numbers and a prime number p, find all integer c ($0 \le c < p$) such that $\text{mex}(S_c)$ is maximized where $S_c = \{(c \cdot x) \bmod p | x \in S\}$. Here $\text{mex}(S)$ is the smallest non-negative integer x such that $x \notin S$.

Input

The input contains multiple test cases.

The first line contains an integer T, denoting the number of test cases.

For each test case, the first line contains two integers n, p ($1 \le n \le p \le 2 \times 10^5$), denoting the size of the set and the prime. It is guaranteed that p is a prime.

The following line contains n integers, the i-th integer is a_i ($0 \le a_i < p$), denoting one element in the set. It is guaranteed that $a_i \ne a_j$ if $i \ne j$.

It is guaranteed that the sum of p over all test cases will not exceed 2×10^5.

Output

For each test case, output two integers k, m on the first line, denoting the number of c and the mex.

Output k integers on the following line, denoting the possible c in increasing order.

Example

standard input	standard output
3 2 3 0 2 3 5 2 3 4 3 5 0 2 3	1 2 2 1 1 0 2 2 2 3