

Game

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 1024 mebibytes

There are n piles of stones; the i -th pile contains a_i stones.

Two players take turns making moves. In one move, the current player chooses a pile and removes a positive number of stones k from it. The move is allowed only if the binary representation of k contains at most m ones.

The player who cannot make a move loses. Determine who wins if both players play optimally.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$): the number of test cases.

Each test case is described by two lines. The first line contains two integers n and m ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq 60$): the number of piles and the maximum number of ones allowed in the binary representation of a removed amount.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^{18}$): the sizes of the piles. The sizes are not necessarily distinct.

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

Output

For each test case, print **First** if the player who moves first wins, and **Second** otherwise.

Example

<i>standard input</i>	<i>standard output</i>
3	Second
1 2	First
7	Second
2 1	
5 6	
3 2	
7 14 21	

Note

In the first test case, $m = 2$, and the only pile contains 7 stones. The first player cannot take all 7 stones, because $7 = 111_2$ contains three ones. It can be checked that every allowed move leads to a position in which the second player wins.

In the second test case, the first player can take 2 stones from the first pile, leaving piles of sizes 3 and 6.