

Game Show: XOR Edition

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

Famous game show returns with yet another season — now it is XOR edition! The show works like this: the contestant has a bag of numbers, initially empty. Then n rounds will happen in order. In each round, the contestant will be offered an integer, uniformly randomly chosen from $[0; 2^k - 1]$. Then the contestant can choose to either add it to their bag or ultimately skip it. After all n rounds, the contestant goes away with the prize equal to the XOR of the numbers in their bag. Note that the XOR of an empty bag is considered to be equal to zero.

The contestant knows n and k , and the fact that numbers are generated uniformly randomly from $[0; 2^k - 1]$, but does not know exactly what the numbers are equal to. Given that, the contestant optimizes their strategy to maximize the expected value of the prize, and also, for the great show, if in the optimal strategy taking or not taking the next number leads to the same expected value, the contestant will always choose to pick a number.

You are the producer of the show. You were given a list of integers that will be used for tomorrow's show, that are indeed uniformly randomly generated. You want to minimize the prize for the contestant, to cut the costs of a show. But changing the numbers themselves may raise too much suspicion; therefore, you only want to change the order of the numbers in which they will be presented to the contestant. Of course, you are doing this secretly, and no one, including the contestant, knows about such shenanigans; therefore, the contestant's strategy will remain the same. Please find any possible order of the numbers such that the contestant will make as little money as possible!

Input

First line of the input contains two integers k, n ($4 \leq k \leq 18, 10k \leq n \leq 2 \cdot 10^5$).

Next line contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i < 2^k$). It is guaranteed that these integers are uniformly randomly generated.

This problem has 90 tests, with 6 tests for each value of $4 \leq k \leq 18$, where n values are distributed in arithmetic progression between $10k$ and $2 \cdot 10^5$, including both ends.

Output

Output n integers: rearranged array $a_1, a_2, \dots, a_n$ that achieves a minimal value. If there are multiple answers, output any of them.

Example

standard input	standard output
4 40 4 12 3 5 9 7 0 6 13 10 12 14 14 7 2 8 3 4 2 3 4 9 13 4 13 6 10 2 0 10 4 9 15 13 10 8 6 7 14 3	***

Note

In order to fit on the page, several line breaks were added to sample input. The real inputs follow the input description.

We deliberately don't show you the answer for the first test. Follow output format.