Problem B: Puzzle About Magical Quadruples

Time limit: 1s; Memory limit: 512 MB

Quynh Anh is an adventurous young programmer who loves solving puzzles with numbers. One day, while wandering through a digital forest named ESC21, she stumbled upon an enchanted scroll that described a peculiar puzzle involving arrays and bitwise operations. The puzzle seemed so intriguing that she decided to challenge herself and her friends to solve it.

Here's how the puzzle works:

Quynh Anh has a collection of **N** non-negative integers, represented as an array **A** where **A[1]**, **A[2]**, ..., **A[N]** are the elements. She defines a *super beautiful quadruple* as a tuple of four indices **(i, j, p, q)** such that:

- **1 ≤ i**, **j**, **p**, **q ≤ N** (note that they are not necessarily distinct).
- **A[i] & A[j] & A[p] & A[q] = 0**, where **&** denotes the bitwise **AND** operation.

In other words, the bitwise **AND** of the four chosen elements at these indices must equal zero.

But that's not all! Quynh Anh wants to explore these quadruples in a very specific order. The quadruples should be sorted lexicographically, meaning that a quadruple **(b[1]**, **b[2]**, **b[3]**, **b[4])** is considered smaller than another quadruple **(c[1]**, **c[2]**, **c[3]**, **c[4])** if there exists an index **i** ($1 \le i \le 4$) such that:

- **b[1]** = **c[1]**, **b[2]** = **c[2]**, ..., **b**[**i-1]** = **c[i-1]**
- and **b[i]** < **c[i]**.

Your challenge is to help Quynh Anh find the **k**-th lexicographically smallest quadruple from all possible super beautiful quadruples that satisfy the above conditions.

Input

- The first line contains two integers **N** and **k** ($1 \le N \le 5000$, $1 \le k \le N^4$).
- The second line contains **N** integers, representing the array **A**. ($0 \le A[x] \le 10^6$ for all $1 \le x \le N$)

Output

- Output four integers, **i, j, p, q**, representing the **k**-th lexicographically smallest super beautiful quadruple. In the case of **k** is greater than the total number of beautiful quadruples, print **-1**.

Sample

Input	Output
6 1 1 4 7 10 13 16	1 1 1 2
6 2 1 4 7 10 13 16	1 1 1 4
6 1120 1 4 7 10 13 16	6 6 6 5
6 1121 1 4 7 10 13 16	-1