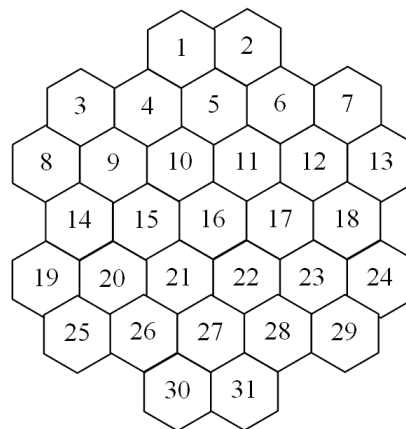


Problem E. Hexodoku

Input file: `hexodoku.in`
Output file: `hexodoku.out`
Time limit: 10 seconds
Memory limit: 256 megabytes

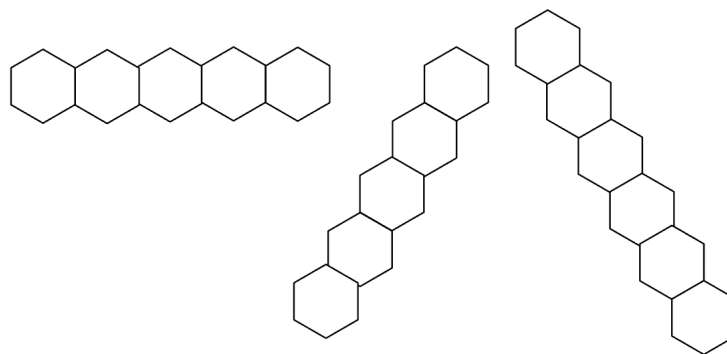
Sudoku is an amazing game. Many people have fun solving it. They say that a legendary programmer had spent just 7 minutes on writing a program that solves standard sudoku. That's über cool, don't you think? And now he has solved another problem. Can you do the same?

Consider a non-standard hexagonal Sudoku board:

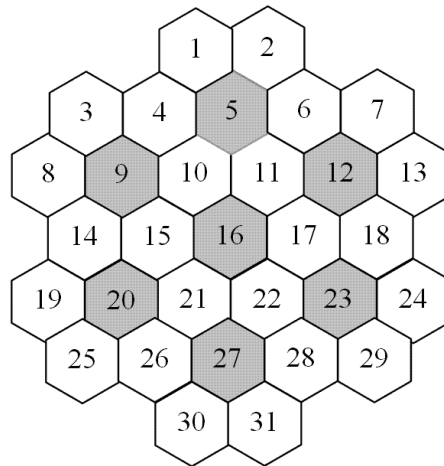


The cells are numbered from 1 to 31.

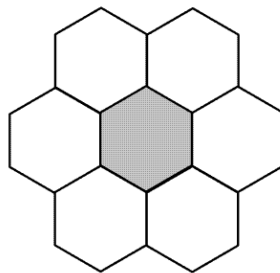
According to the rules, numbers (from 1 to K) can be placed in the cells with the condition that all numbers in the same row (rows are located in three directions) must be different.



Additionally, for each of the marked cells, the number in marked cell and all numbers in adjacent cells must also differ from each other.



Must be different



Some numbers may already be placed in the cells according to the rules. You are to find N -th solution in lexicographical order, if it exists.

Let A_i be the number in the cell i in solution A , and B_i — the number in the cell i in solution B . Solution A is lexicographically smaller than solution B , if such j exists that for each i where $i < j$: $A_i = B_i$ and $A_j < B_j$.

Input

First line of input contains two integers K and N .

- $7 \leq K \leq 31$
- $1 \leq N \leq 100\,000$

Second line contains 31 integer numbers: A_i ($1 \leq i \leq 31$) is the number standing in the cell i .

- $1 \leq A_i \leq K$, or 0, if there is no number in this cell.

Output

First line of output should contain an answer:

- “Found” — if the solution has been found.
- “No way” — if there is no N -th solution.

If the solution has been found, the second line of output should contain the N -th solution in the same format as in input file.

This is the first example above:

