Invisible Edges

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

This is a multi-pass problem.

VegiCat and VegiDog play a game with a forest of n vertices, numbered from 1 to n, and m edges. A forest is an undirected graph with no cycles. It is **not necessarily connected**.

In the first stage, VegiCat assigns an integer a_u ($1 \le a_u \le 2n$) to each vertex u. The values may repeat.

After VegiCat finishes assigning the values, the vertices are renumbered. Let $p_1, p_2, \ldots, p_n$ be a permutation of $1, 2, \ldots, n$, in which each integer from 1 to n appears exactly once. For each u, the vertex originally numbered u is now numbered p_u.

Each edge (u, v) becomes (p_u, p_v). Each vertex keeps its assigned value. Let b_i be the value at vertex i after renumbering. Then $b_{p_u} = a_u$.

In the second stage, VegiDog receives only n and $b_1, b_2, \ldots, b_n$, without the original forest or p. He must recover all edges using the new vertex numbers.

Design strategies for both players that work for every valid forest and every permutation p.

Input

Your program will run twice: first as VegiCat, then as VegiDog. The test cases appear in the same order in both runs.

First Run. The first line contains the string `cat`, indicating that you play VegiCat.

The second line contains an integer T ($1 \le T \le 10^4$), the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \le n \le 2 \cdot 10^5$, $0 \le m < n$), the number of vertices and edges.

Each of the next m lines contains two integers u and v ($1 \le u, v \le n$, $u \ne v$), describing an undirected edge between vertices u and v.

It is guaranteed that the graph has no cycles or multiple edges.

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

Second Run. The first line contains the string `dog`, indicating that you play VegiDog.

The second line contains an integer T ($1 \le T \le 10^4$), the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($1 \le n \le 2 \cdot 10^5$), the number of vertices.

The second line contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 2n$), the assigned values in the new vertex order.

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

Output

First Run. For each test case, output n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2n$) on one line. The integer a_i is the value assigned to vertex i.

Second Run. For each test case, output an integer m on the first line, the number of edges. Each of the next m lines must contain two integers u and v ($1 \le u, v \le n$, $u \ne v$), describing an edge using the new vertex numbers.

Output each edge once, in any order. Either endpoint may come first. If $m = 0$, output only 0.

Examples

standard input	standard output
cat 4 1 0 4 3 1 2 4 3 2 3 7 2 6 4 7 1 10 7 10 9 7 9 10 5 4 8 3 7 4 10 1 2	1 2 5 8 5 11 11 4 11 3 7 2 10 13 12 5 3 2 16 6 19 3
dog 4 1 1 4 5 2 5 8 7 11 7 2 11 11 3 4 10 16 3 13 5 10 12 19 6 2 3	0 3 1 4 2 1 4 3 2 4 3 5 2 7 1 7 4 8 5 3 6 1 7 10 10 2 10 4

Note

In the first test case, the forest has one vertex and no edges. VegiCat assigns the value 1 to the vertex, and VegiDog outputs 0.

In the last test case, the permutation is $p = [5, 3, 6, 4, 2, 9, 1, 8, 7, 10]$.

VegiCat assigns $a_7 = 16$. Since $p_7 = 1$, VegiDog receives $b_1 = 16$.

The original edge $(3, 7)$ becomes $(p_3, p_7) = (6, 1)$.

The original edge $(4, 10)$ becomes $(p_4, p_{10}) = (4, 10)$. The sample outputs this edge as $(10, 4)$, which represents the same undirected edge.

The sample illustrates the input and output formats. Its outputs are correct for the graphs and permutations shown. However, the sample assignments are not guaranteed to be part of a strategy that works for every valid forest and every permutation.