

E. Two Rooted Trees

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You have two rooted undirected trees, each contains n vertices. Let's number the vertices of each tree with integers from 1 to n . The root of each tree is at vertex 1. The edges of the first tree are painted blue, the edges of the second one are painted red. For simplicity, let's say that the first tree is blue and the second tree is red.

Edge $\{x, y\}$ is called bad for edge $\{p, q\}$ if two conditions are fulfilled:

1. The color of edge $\{x, y\}$ is different from the color of edge $\{p, q\}$.
2. Let's consider the tree of the same color that edge $\{p, q\}$ is. Exactly one of vertices x, y lies both in the subtree of vertex p and in the subtree of vertex q .

In this problem, your task is to simulate the process described below. The process consists of several stages:

1. On each stage edges of exactly one color are deleted.
2. On the first stage, exactly one blue edge is deleted.
3. Let's assume that at the stage i we've deleted edges $\{u_1, v_1\}, \{u_2, v_2\}, \dots, \{u_k, v_k\}$. At the stage $i + 1$ we will delete all undeleted bad edges for edge $\{u_1, v_1\}$, then we will delete all undeleted bad edges for edge $\{u_2, v_2\}$ and so on until we reach edge $\{u_k, v_k\}$.

For each stage of deleting edges determine what edges will be removed on the stage. Note that the definition of a bad edge always considers the initial tree before it had any edges removed.

Input

The first line contains integer n ($2 \leq n \leq 2 \cdot 10^5$) — the number of vertices in each tree.

The next line contains $n - 1$ positive integers a_2, a_3, \dots, a_n ($1 \leq a_i \leq n; a_i \neq i$) — the description of edges of the first tree. Number a_i means that the first tree has an edge connecting vertex a_i and vertex i .

The next line contains $n - 1$ positive integers b_2, b_3, \dots, b_n ($1 \leq b_i \leq n; b_i \neq i$) — the description of the edges of the second tree. Number b_i means that the second tree has an edge connecting vertex b_i and vertex i .

The next line contains integer idx ($1 \leq idx < n$) — the index of the blue edge that was removed on the first stage. Assume that the edges of each tree are numbered with numbers from 1 to $n - 1$ in the order in which they are given in the input.

Output

For each stage of removing edges print its description. Each description must consist of exactly two lines. If this is the stage when blue edges are deleted, then the first line of the description must contain word *Blue*, otherwise — word *Red*. In the second line print the indexes of the edges that will be deleted on this stage in the increasing order.

Examples

input
5 1 1 1 1 4 2 1 1 3
output
Blue 3 Red

1 3
Blue
1 2
Red
2

Note

For simplicity let's assume that all edges of the root tree received some direction, so that all vertices are reachable from vertex 1. Then a *subtree* of vertex v is a set of vertices reachable from vertex v in the resulting directed graph (vertex v is also included in the set).