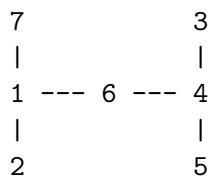


5021 Tree Representation

Given a labeled tree T with n vertices, repeat the following step: *delete the leaf with the smallest label and record the label of its parent until only a single edge remains*. The resulting sequence of $n - 2$ labels can be used to represent the tree.

To make life easier, let the labels of the tree be $1, \dots, n$. Observe that every tree has at least two nodes of degree 1 (i.e., tree leaves). More specifically, the node v which is incident to the lowest labeled leaf is uniquely determined, and v is then taken as the first symbol in the sequence. This lowest labeled leaf is then deleted and the procedure is repeated until a single edge is left, giving a total of $n - 2$ integers between 1 and n . Note the important fact that the number of occurrences of the label i in the sequence is one less than the degree of the vertex i in the original tree.

For example, the sequence 1, 4, 4, 6, 1 represents the following labeled spanning tree. Vertex 4 has degree 3 and appears twice in the above sequence.



Now given a sequence of $n - 2$ numbers from the set of labels $\{1, 2, \dots, n\}$, your task is to write a program to recover the tree. Note that any sequence is associated with a unique labeled tree.

Input

The input file contains a set of test data. The first line of the input file contains a positive integer m (≤ 5) indicating the number of test cases to follow. Each test case starts with a positive integers n ($5 \leq n \leq 50$) indicating the tree size and then $(n - 2)$ integers from $\{1, 2, \dots, n\}$ follow.

Output

For each test case, output the adjacent list of the tree, where the vertices are listed in increasing order. Likewise the neighbor(s) of all vertices are ordered in increasing order. Answers for adjacent cases are separated by a blank line.

Sample Input

```

2
7 1 4 4 6 1
5 1 1 1
  
```

Sample Output

```

1 2 6 7
2 1
3 4
4 3 5 6
  
```

5 4

6 1 4

7 1

1 2 3 4 5

2 1

3 1

4 1

5 1