B1. Books Exchange (easy version)

time limit per test: 1 second
memory limit per test: 256 megabytes

The only difference between easy and hard versions is constraints.

There are n kids, each of them is reading a unique book. At the end of any day, the i-th kid will give his book to the p_i-th kid (in case of $i = p_i$ the kid will give his book to himself). It is guaranteed that all values of p_i are distinct integers from 1 to n (i.e. p is a permutation). The sequence p doesn't change from day to day, it is fixed.

For example, if $n = 6$ and $p = [4, 6, 1, 3, 5, 2]$ then at the end of the first day the book of the 1-st kid will belong to the 4-th kid, the 2-nd kid will belong to the 6-th kid and so on. At the end of the second day the book of the 1-st kid will belong to the 3-th kid, the 2-nd kid will belong to the 2-th kid and so on.

Your task is to determine the number of the day the book of the i-th child is returned back to him for the first time for every i from 1 to n.

Consider the following example: $p = [5, 1, 2, 4, 3]$. The book of the 1-st kid will be passed to the following kids:

- after the 1-st day it will belong to the 5-th kid,
- after the 2-nd day it will belong to the 3-rd kid,
- after the 3-rd day it will belong to the 2-nd kid,
- after the 4-th day it will belong to the 1-st kid.

So after the fourth day, the book of the first kid will return to its owner. The book of the fourth kid will return to him for the first time after exactly one day.

You have to answer q independent queries.

Input

The first line of the input contains one integer q ($1 \le q \le 200$) — the number of queries. Then q queries follow.

The first line of the query contains one integer n ($1 \le n \le 200$) — the number of kids in the query. The second line of the query contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$, all p_i are distinct, i.e. p is a permutation), where p_i is the kid which will get the book of the i-th kid.

Output

For each query, print the answer on it: n integers $a_1, a_2, \ldots, a_n$, where a_i is the number of the day the book of the i-th child is returned back to him for the first time in this query.

Example

input Copy

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

output Copy

```
1 1 1 1 1
3 3 3
2 3 3 2 1 3
1
2 2 2 2
4 4 4 1 4
```