An Unsure Catch

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Some prisoners have managed to escape from the prison! Your job is to get them back.

At time 0, there are n prisoners out in the city, each at a different position. We label these positions from 1 to n.

Your colleagues have made a plan to block as many road as possible so that there will be only one way out for each position. Once in an hour, the prisoners at position i will leave and arrive at position $a_i(1 \le a_i \le n)$ immediately.

You can arrange an assault at any integer time at any position. Every prisoner at the same location at that time will be arrested. However, you can only attack once, after that all the prisoners on the run will leave the city.

After consideration, you decide to contact your colleagues to change their plan. Specifically, you can change some a_i before time 0 ($1 \le a_i \le n$ must still hold after changing).

Now you want to know, what's the minimum number of a_i to change (let's denote the answer as K) in order to catch all the prisoners. You also want to know, if you can change a_i for at most j ($j \in \{0, 1, 2, \ldots, K\}$) times, how many prisoners at most can you arrest. Write a program to solve the problem.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$), indicating the number of prisoners and positions.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$), their meanings are described above.

For only about 5 cases will n larger than 50.

Output

For each test case output two lines.

For the first line, print an integer K, indicating the least number of a_i to change so that you can arrest all the prisoners.

For the second line, print $K + 1$ integers separated by one space, indicating the number of prisoners you can arrest at most if you can change a_i for at most $0, 1, 2, \ldots, K$ times.

Please, DO NOT print extra spaces at the end of each line.

Example

standard input	standard output
2 1 1 10 2 3 1 4 5 7 6 6 6 7	0 1 3 3 6 9 10