

Paimon Sorting

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Paimon just invents a new sorting algorithm which looks much like *bubble sort*, with a few differences. It accepts a 1-indexed sequence A of length n and sorts it. Its pseudo-code is shown below.

Algorithm 1 The Sorting Algorithm

```
1: function SORT( $A$ )  
2:   for  $i \leftarrow 1$  to  $n$  do             $\triangleright n$  is the number of elements in  $A$   
3:     for  $j \leftarrow 1$  to  $n$  do  
4:       if  $a_i < a_j$  then             $\triangleright a_i$  is the  $i$ -th element in  $A$   
5:         Swap  $a_i$  and  $a_j$   
6:       end if  
7:     end for  
8:   end for  
9: end function
```

If you don't believe this piece of algorithm can sort a sequence it will also be your task to prove it. Anyway here comes the question:

Given an integer sequence $A = a_1, a_2, \dots, a_n$ of length n , for each of its prefix A_k of length k (that is, for each $1 \leq k \leq n$, consider the subsequence $A_k = a_1, a_2, \dots, a_k$), count the number of swaps performed if we call $\text{SORT}(A_k)$.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \leq n \leq 10^5$) indicating the length of the sequence.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) indicating the given sequence.

It's guaranteed that the sum of n of all test cases will not exceed 10^6 .

Output

For each test case output one line containing n integers s_1, s_2, \dots, s_n separated by a space, where s_i is the number of swaps performed if we call $\text{SORT}(A_i)$.

Please, DO NOT output extra spaces at the end of each line or your solution may be considered incorrect!

Example

standard input	standard output
3	0 2 3 5 7
5	0 2 4
2 3 2 1 5	0
3	
1 2 3	
1	
1	