Problem G. Round Table

Source file name:	Round.c, Round.cpp, Round.java, Round.py
Input:	`Standard`
Output:	`Standard`

There are n people, numbered from 1 to n, sitting at a round table. Person $i+1$ is sitting to the right of person i (with person 1 sitting to the right of person n).

You have come up with a better seating arrangement, which is given as a permutation $p_1, p_2, \ldots, p_n$. More specifically, you want to change the seats of the people so that at the end person p_{i+1} is sitting to the right of person p_i (with person p_1 sitting to the right of person p_n). Notice that for each seating arrangement there are n permutations that describe it (which can be obtained by rotations).

In order to achieve that, you can swap two people sitting at adjacent places; but there is a catch: for all $1 \leq x \leq n-1$ you cannot swap person x and person $x+1$ (notice that you **can** swap person n and person 1). What is the minimum number of swaps necessary? It can be proven that any arrangement can be achieved.

Input

Each test contains multiple test cases. The first line contains an integer t $\left(1 \leq t \leq 10^4\right)$ – the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains a single integer n $\left(3 \leq n \leq 2 \cdot 10^5\right)$ – the number of people sitting at the table.

The second line contains n distinct integers $p_1, p_2, \ldots, p_n$ $(1 \leq p_i \leq n,\ p_i \neq p_j$ for $i \neq j)$ – the desired final order of the people around the table.

The sum of the values of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print the minimum number of swaps necessary to achieve the desired order.

Example

Input	Output
3 4 2 3 1 4 5 5 4 3 2 1 7 4 1 6 5 3 7 2	1 10 22

Explanation

In the **first test case**, we can swap person 4 and person 1 (who are adjacent) in the initial configuration and get the order [4, 2, 3, 1] which is equivalent to the desired one. Hence in this case a single swap is sufficient.