

Snacks Scheduling

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

Busy Beaver has N favorite snack spots along the Charles River, labeled $1, 2, \dots, N$. Each day, he wishes to enjoy a snack at spot i during time i . What a perfectly optimized routine!

Unfortunately, the geese have other plans. For each index i , a goose occupies spot a_i at time i . Busy Beaver absolutely refuses to visit the same spot as a goose, so his final schedule p_1, p_2, \dots, p_N must be a permutation of $1, 2, \dots, N$ that satisfies $p_i \neq a_i$ for every $1 \leq i \leq N$.

Busy Beaver also wants to keep his life predictable. Among all valid schedules p , determine the minimum possible number of inversions* p can have. If no valid schedule exists, output -1 .

Input

The first line contains a single integer T ($1 \leq T \leq 3 \cdot 10^5$) representing the number of test cases.

The first line of each test case contains a single integer N ($1 \leq N \leq 3 \cdot 10^5$).

The second line of each test case contains N integers a_1, a_2, \dots, a_N ($1 \leq a_i \leq N$).

The sum of N over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, print a single integer: the minimum possible inversion count among all permutations p satisfying $p_i \neq a_i$ for all i . If no valid schedule exists, print -1 .

Example

standard input	standard output
4	0
3	2
2 3 1	-1
4	9
1 2 3 4	
5	
1 1 1 1 1	
10	
1 1 1 1 1 10 10 10 10 10	

Note

In the first test case, no position satisfies $a_i = i$, so Beaver can use his original routine $p = [1, 2, 3]$, which has 0 inversions.

In the second test case, a valid schedule is $p = [2, 1, 4, 3]$, which has exactly 2 inversions.

In the third test case, spot 1 is occupied throughout the day, so that poor Busy Beaver fails to schedule. What a pity!

*Recall that the number of inversions in p is defined as the number of pairs (i, j) with $1 \leq i < j \leq N$ and $p_i > p_j$.