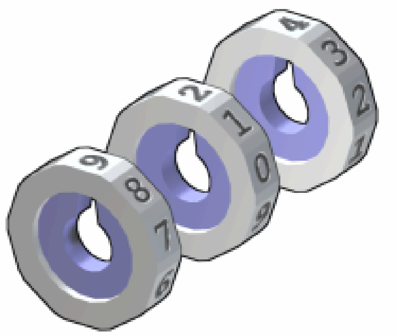
Combination Lock

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Ahmed is a regular traveler and an aviation passionate. He has visited a lot of countries all over the world, and plans on becoming a pilot one day to keep on discovering it.

Ahmed is a bit of a scatterbrain however. He has trouble keeping his luggage safe and he already happened to forget one of his suitcases in an airport. But after his last trip to Brazil, where he discovered his suitcase was opened and stolen from, he finally decided to start using combination locks to secure his items.



A combination lock with the current configuration being $[9, 2, 4]$

A combination lock consists of n rotating discs $d_1, \cdots, d_n$ with inscribed numbers $a_1, \cdots, a_n$, ranging from 1 to k, which directly interact with the locking mechanism. It not only offers more security, but also considerably more peace-of-mind. It wasn't the case for Ahmed the first time he used it though, as he forgot to personnalize the lock and left the default secret unlocking combination consisting of all 1s.

On the plane, worrying, he kept wondering what was the minimum number of operations an ill-intentionned person needed to do to go from the initial lock configuration (that is the one he left on the suitcase) to the unlocking one consisting of 1s only.

We define the operation $O(i, j)$ that one can perform on the combination lock as follows: choose any two integers $i, j \in \{1, \cdots, n\}$ and rotate the disks $d_i, \cdots, d_j$ once **to the right**, *ie.* in the direction that decreases the values on the disks (See the figure above). In particular, performing such an operation on a disk pointing to the value 1 changes its value to k.

Input

The first list contains an integer T – the number of test cases. Each test case consists of two lines. The first line contains two space-separated integers $1 \leq n \leq 10^5$ and $1 \leq k \leq 10^9$. The next line contains n space-separated integers $1 \leq a_i \leq k$ representing the initial lock configuration.

Output

For each test case, print the minimum number of operations needed to reach the configuration where all the lock's disks point to the value 1.

Example

standard input	standard output

4
3 3
3 3 3
5 10
1 2 3 4 5
9 5
3 1 5 4 3 2 1 5 2
6 5
3 1 2 1 4 1
	2
	4
	10
	6

Note

In the first test case, the optimal choice is to perform $O(1, 3)$ twice.

In the second test case, the optimal way is to perform $O(2, 5)$, $O(3, 5)$, $O(4, 5)$ and $O(5, 5)$.