

Round B - Coding Competitions Farewell Rounds

Intruder Outsmarting

PROBLEM

ANALYSIS

Problem

Amiria is a cautious internet user, and as such, she is setting up two-factor authentication for her accounts. She is using a special type of security key as an extra precaution to outsmart any intruders that may want to take it. Amiria's security key requires a code to activate. To enter the code, one must place it on wheels with numbers, similar to code padlocks.

Amiria's security key has a sequence of W wheels. Each wheel has the numbers 1 through N printed in order. When rotating the wheel, the user can move the currently shown integer either to the next or the previous integer. Numbers on the wheel wrap around. This means the number after N is 1 and the number before 1 is N .

There is no hidden password. To activate the security key Amiria needs to move the wheels such that the sequence of numbers shown is palindromic. That is, the sequence of numbers is the same when read from left to right and from right to left. To slow down intruders, Amiria rigged the security key such that the wheels only rotate in increments of D . That is, on a move, a wheel that is currently showing x can be made to show $x - D$ or $x + D$, applying the proper wraparound. That is, if $x - D < 1$ the actual number shown after the operation is $x - D + N$, and if $x + D > N$ the actual number shown is $x + D - N$.

Amiria wants to check how much this system would slow down an intruder trying to use her security key. Given the number of wheels and the number currently shown on each wheel, find the minimum number of operations needed to make the sequence of shown numbers palindromic, or report that it is impossible to do so.

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case consists of two lines. The first line of a test case contains 3 integers W , N , and D : the number of wheels in Amiria's security key, the number of integers shown in each of those wheels, and the fixed increment that Amiria rigged for every wheel. The second line of a test case contains W integers X_1, X_2, \dots, X_W , where X_i is the number currently shown in the i -th wheel from left to right.

Output

For each test case, output one line containing `Case # x : y` , where x is the test case number (starting from 1) and y is the minimum number of operations needed to make the sequence of shown numbers palindromic. If there is no way to make the sequence of shown numbers palindromic through the allowed operations, y must be `IMPOSSIBLE` instead.

Limits

Time limit: 5 seconds.

Memory limit: 2 GB.

$$1 \leq \mathbf{T} \leq 100.$$

$$1 \leq \mathbf{D} \leq \mathbf{N} - 1.$$

$$1 \leq \mathbf{X}_i \leq \mathbf{N}, \text{ for all } i.$$

Test Set 1 (Visible Verdict)

$$2 \leq \mathbf{W} \leq 5.$$

$$2 \leq \mathbf{N} \leq 5.$$

Test Set 2 (Visible Verdict)

$$2 \leq \mathbf{W} \leq 1000.$$

$$2 \leq \mathbf{N} \leq 10^9.$$

Sample

Sample Input	Sample Output
<pre> 3 5 5 4 1 4 5 5 4 3 4 2 3 4 3 2 4 2 1 4 </pre>	<pre> Case #1: 3 Case #2: 0 Case #3: IMPOSSIBLE </pre>

In Sample Case #1, the sequence can be made `5 4 5 4 5`, which is palindromic, with 3 operations by using one addition operation on the first and fourth wheels, and one subtraction operation on the fifth wheel. There is no way to make the sequence palindromic with fewer moves.

In Sample Case #2 the sequence is already palindromic, so we do not need any operations.

In Sample Case #3, both numbers would need to be equal for the sequence to be palindromic. Since wheel values can only move by 2 and both current numbers have different parity, that cannot be done.