

# K-Coverage

Input file:            **standard input**  
Output file:         **standard output**  
Time limit:          1 second  
Memory limit:       1024 megabytes

Panda has a one-dimensional axis with  $n$  segments placed on its non-negative side. The left endpoint of the  $i$ -th segment is  $l_i$ . Every segment has the same length,  $L$ . This means the  $i$ -th segment covers the interval  $[l_i, l_i + L - 1]$  (inclusive).

Now, Panda wants you to find the maximum number of integer points on the non-negative side of the axis that can be covered **exactly**  $k$  times after moving **at most one** segment. Specifically, you are allowed to select at most one segment and change its left endpoint to **any non-negative integer** (with no upper bound). The segment's length,  $L$ , remains the same.

## Input

The first line contains an integer  $T$  ( $1 \leq T \leq 2 \times 10^5$ ), denoting the number of test cases.

For each test case, the first line contains three positive integers  $n, L, k$  ( $1 \leq n \leq 2 \times 10^5$ ,  $1 \leq L \leq n$ ,  $1 \leq k \leq n$ ).  $n$  is the total number of segments,  $L$  is the fixed length of every segment, and  $k$  is the required coverage count.

The second line contains  $n$  non-negative integers. The  $i$ -th integer is  $l_i$  ( $0 \leq l_i \leq 2 \times n$ ) representing the left endpoint of the  $i$ -th segment.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $2 \times 10^5$ .

## Output

For each test case, output one line with a single integer, denoting the maximum number of integer points that can be covered exactly  $k$  times after moving at most one segment.

## Example

standard input	standard output
3	6
3 2 1	3
2 6 2	0
3 3 2	
6 2 0	
5 1 3	
5 6 7 8 9	

## Note

For the first test case in the example, initially, the three segments are  $[2, 3]$ ,  $[6, 7]$ , and  $[2, 3]$ . You can change the third segment's left endpoint to 114514. After this move, the segments are  $[2, 3]$ ,  $[6, 7]$ , and  $[114514, 114515]$ . This results in 6 integer points covered exactly once, which is the maximum achievable.

For the second test case, you can move the first segment's left endpoint to 3 to maximize the number of integer points that are covered exactly twice.