



Zero-Move Nim

by nikasvanidze

Problem

Submissions

Leaderboard

Discussions

Editorial

Nim is a famous game in which two players take turns removing items from n distinct piles. During each turn, a player must remove one or more items from a single, non-empty pile. The winner of the game is whichever player *removes the last item from the last non-empty pile*.

John and Kate modified Nim by adding the following rule, which they call a *Zero-Move*:

For each *non-empty* pile, either player can remove **0** items from that pile and have it count as their move; however, this move can only be performed *once per pile by either player*. For example, let's say pile i initially has $p_i = 2$ items in it. If John decides to use a Zero-Move on pile i , then neither John nor Kate can perform another Zero-Move on pile i ; that said, either player is free to perform a Zero-Move on any other non-empty pile that hasn't had a Zero-Move performed on it yet.

John and Kate play g games of *Zero-Move Nim*. Given the number of items in each pile for each game, determine whether or not John can win the game if he always moves first and each player always moves optimally (i.e., never makes a move that causes them to lose if some better, winning move exists). For each game, print **W** on a new line if John can win; otherwise, print **L** instead.

Input Format

The first line contains an integer, g , denoting the number of games. The $2 \cdot g$ subsequent lines describe each game over two lines:

1. The first line contains an integer, n , denoting the number of heaps.
2. The second line contains n space-separated integers describing p_0, p_1, \dots, p_{n-1} .

Constraints

- $1 \leq g \leq 10$
- $1 \leq n \leq 10$
- $1 \leq p_i < 2^{30}$

Subtasks

- For **40%** of the test cases, $1 \leq p_i < 2^7$
- For **60%** of the test cases, $1 \leq p_i < 2^{17}$

Output Format

For each game, print **W** on a new line if John will win; otherwise, print **L** instead.

Sample Input 0

```
2
2
1 2
2
2 2
```

Sample Output 0

W
L**Explanation 0**

John and Kate play the following $g = 2$ games:

1. We have two piles, $p_0 = 1$ and $p_1 = 2$. John removes 1 item from p_1 , so $p_1 = 2 - 1 = 1$. Now that there is only 1 item in each pile, gameplay can proceed in either of the following ways:

- Kate removes the last object from one pile, then John removes the last object from the other pile.
- Kate uses a Zero-Move on one of the piles, and John uses a Zero-Move on the other pile. Next, Kate must take the last object from one pile, at which point John removes the last object from the other pile.

Because John always wins in either scenario, we print W on a new line.

2. John cannot win this game because the two piles are of equal size and Kate has an opportunity to counter any move he makes by performing the same action. Consider the following scenarios:

- If John uses a Zero-Move on one pile, Kate can use a Zero-Move on the other pile (meaning the piles still have the same configuration after both players move).
- If John removes one element from a pile, Kate can remove one element from the other pile so that both remaining piles contain one element when John takes his next turn. He would then be forced to empty one of the piles, leaving Kate to make the winning move by emptying the last pile.
- If John removes both elements from one of the piles, Kate can remove both elements from the other pile and win the game.

Because John always loses this game, we print L on a new line.

[f](#) [t](#) [in](#)

 Submissions: [305](#)

Max Score: 50

Difficulty: Medium

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

[More](#)

Current Buffer (saved locally, editable)

Java 7



```

1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8
9     public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11         int g = in.nextInt();
12         for(int a0 = 0; a0 < g; a0++){
13             int n = in.nextInt();
14             int[] p = new int[n];
15             for(int p_i=0; p_i < n; p_i++){
16                 p[p_i] = in.nextInt();
17             }
18             // your code goes here
19         }
20     }
21 }
22

```

Line: 1 Col: 1

 [Upload Code as File](#)☐ Test against custom input

Run Code

Submit Code

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)