

A **Array Annihilation**

Time limit: 2s

Already another algorithms assignment about arrays – Amy has absolute aversion to them all! To avenge the amountful assignments, she activates an Array Annihilator Pledging Justice Executable, so all annoying arrays are abolished.

As the arrays are astronomically large, she agrees to only use one type of operation repeatedly. For each operation, she chooses two or more consecutive values in the array, and subtracts 1 from each of them. Given an array, is it possible to make all values equal to 0 in some finite number of these operations?

As an example, consider the second sample input. The array can be fully annihilated in four operations: first, decrement the first two values (yielding [1, 3, 3]), then decrement the last two values twice (yielding [1, 1, 1]), and finally decrement the entire array once (yielding [0, 0, 0]).



Amy, actively annihilating an array.
Image generated using DALL·E

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 5 \cdot 10^5$), the length of the array.
- One line with n integers a ($1 \leq a \leq 10^9$), the values in the array.

Output

If it is possible to make all values of the array equal to 0 in some finite number of operations, output “possible”. If this is not possible, output “impossible”.

Sample Input 1	Sample Output 1
2 1 1	possible
Sample Input 2	Sample Output 2
3 2 4 3	possible
Sample Input 3	Sample Output 3
1 5	impossible

Sample Input 4

```
4
2 1 1 2
```

Sample Output 4

```
impossible
```

Sample Input 5

```
9
3 4 5 5 1 2 3 2 1
```

Sample Output 5

```
possible
```