# Monk and Lucky Minimum

Attempted by: **2774** / Accuracy: **71%** / Maximum Score: **20** / 24 Votes

Tag(s): Data Structures, Easy, approved



Monk just purchased an array $A$ having $N$ integers. Monk is very superstitious. He calls the array $A$ Lucky if the frequency of the minimum element is odd, otherwise he considers it Unlucky. Help Monk in finding out if the array is Lucky or not.

**Input:**
First line consists of a single integer $T$ denoting the number of test cases.
First line of each test case consists of a single integer $N$ denoting the size of array $A$.
Second line of each test case consists of $N$ space separated integers denoting the array $A$.

**Output:**
For each test case, print "Lucky" (without quotes) if frequency of minimum element is odd, otherwise print "Unlucky"(without quotes). Print a new line after each test case.

**Constraints:**
$1 \leq T \leq 10$
$1 \leq N \leq 10^5$
$1 \leq A[i] \leq 10^9$

SAMPLE INPUT

```
2
5
8 8 9 5 9
5
3 3 3 5 3
```

SAMPLE OUTPUT

```
Lucky
Unlucky
```

**Explanation**

In first case, value of minimum element is $5$ and it's frequency is $1$ which is odd, so the array is Lucky.
In second case, value of minimum element is $3$ and it's frequency is $4$ which is even, so the array is Unlucky.

| | |
|---|---|
| **Time Limit:** | 1.0 sec(s) for each input file. |
| **Memory Limit:** | 256 MB |
| **Source Limit:** | 1024 KB |
| **Marking Scheme:** | Marks are awarded when all the testcases pass. |
| **Allowed Languages:** | C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic |



## CODE EDITOR

Enter your code or Upload your code as file. Save C (gcc 4.8.2)

```c
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}

```

1:1

Provide custom input

COMPILE & TEST SUBMIT

*Press Ctrl-space for autocomplete suggestions.* POWERED BY code table



## PROGRAMMERS WHO SOLVED THIS PROBLEM ALSO SOLVED

### Biased Chandan
Attempted By: **4667** / Accuracy: **79**

### Speed
Attempted By: **3599** / Accuracy: **91**

### Modify Sequence
Attempted By: **4876** / Accuracy: **78**