Bonus Problem: Rick is crazy

locked

Rick is planning on a crazy adventure where he needs to quickly pour n number of acids in a row. The acid m is at the position m and has power equal to x_{m}.

Each acid can be poured into either some acid boxes to the left or some boxes to the right. If the m-th acid box is turned to the left, it pours the acid in all such boxes k that k ∈ [m−x_{m},m−1]. Similarly, if it is turned to the right, it pours acid in all such boxes k that k ∈ [m+1, m+x_{m}].

Your goal is to choose a direction for each acid box so each acid box is filled by at least one other box, or report it to Rick if that is impossible.

Input Format

The first line contains one integer y (1 ≤ y ≤ 10000) — the number of test cases.

Each test case consists of two lines. The first line contains one integer n (2 ≤ n ≤ 3*10000) — the number of acids.

The second line contains n integers m_{1}, m_{2},... ,m_{n} (0 ≤ m_{i} ≤ n) — the power of the *i*-th acid.

The sum of n over all test cases does not exceed 3*10000.

Constraints

- (1 ≤ y ≤ 10000)
- (2 ≤ n ≤ 3*10000)

Output Format

For each test case, print the answer as follows:

If it is possible to direct all acids so that each acid box is filled, print YES in the first line and a string of n characters L or (and) R (the i-th character is L if the i-th acid is turned to the left, otherwise this character is R) in the second line. If there are multiple answers, you may print any of them.

If there is no answer, simply print NO for that test case.

Sample Input 0

```
4
8
0 0 3 1 1 1 1 2
2
1 1
2
2 2
2
0 1
```

Sample Output 0

```
YES
RRLLLLRL
YES
RL
YES
RL
NO
```

Sample Input 1

```
16
4
1 0 3 3
4
3 0 3 0
4
1 4 4 0
4
2 2 2 0
4
3 1 2 2
6
3 6 2 0 3 0
5
0 2 1 1 0
14
12 0 5 4 0 5 0 0 0 0 0 14 0 5
7
0 0 0 0 0 0 0
14
0 0 9 0 0 12 0 0 0 4 0 0 0 5
18
0 0 0 0 0 13 0 0 0 0 0 0 0 0 0 0 0 10
6
0 0 0 5 0 0
2
1 1
7
3 1 0 6 0 6 2
10
0 0 0 0 0 3 1 0 0 0
11
4 4 1 0 11 6 3 4 0 0 0
```

Sample Output 1

```
YES
RRRL
YES
RRLR
YES
RRLR
YES
RLRR
YES
RLRR
YES
RLRRRR
NO
YES
RRLRRRRRRRRRR
NO
YES
RRRRRLRRRRRRR
NO
NO
YES
RL
YES
RLRRRRR
NO
YES
RLRRRRRRRRR
```

Sample Input 2

```
6
11
1 0 0 1 0 0 0 1 0 1 0
11
1 1 0 0 1 0 1 1 1 1 0
13
0 0 0 1 1 0 1 0 1 1 1 1 1
12
0 1 1 0 0 0 1 1 1 1 0 1
13
0 0 0 0 1 1 0 0 0 1 0 0 1
```

```
12
0 0 1 0 0 0 0 1 1 1 0 1
```

Sample Output 2

```
NO
NO
NO
NO
NO
NO
```

Sample Input 3

```
8
8
0 0 2 2 1 0 0 2
7
0 0 2 2 0 0 0
8
0 2 2 1 1 2 1 2
8
0 0 0 0 2 1 1 2
6
2 0 1 1 2 2
7
1 2 1 2 2 1 1
5
1 1 1 0 2
8
2 0 2 2 1 0 2 1
```

Sample Output 3

```
NO
NO
YES
RRLRRRRR
NO
NO
YES
RLRRLRR
NO
YES
RRLRLRRL
```



```cpp
//A CPP Solution by Ayush Chaudhari
//Time Complexity=O(n^2)

#include <bits/stdc++.h>

using namespace std;

```