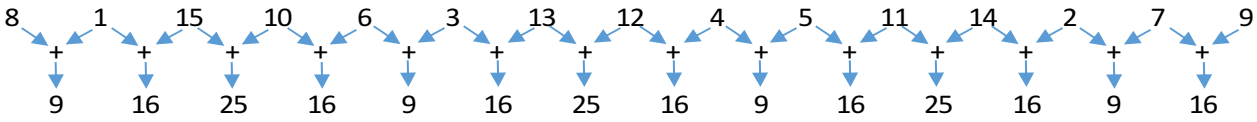
C. Squaring the Sequence

Program:	`square.(cpp|java)`
Input:	`square.in`
Balloon Color:	`Black`

Description

A square number is one that has an integer square root, e.g. 4, 9, 16, are square numbers. It was discovered that the numbers from 1 to 15 can be ordered in such a way that any two successive ones add-up to a square number:



Such an ordering is neither unique, nor necessary in general. For example, the numbers from 1 to 18 cannot form such a sequence.

Your task is to find such a sequence for an arbitrary set of integers provided to you. If multiple solutions exist, you are supposed to provide the one that starts with the smaller numbers possible, or output "No solution" is such an ordering is impossible.

Input Format

The input has a number of test cases (<100). Each test case starts with a line containing an integer N ($1 \le N \le 40$) that declares the integers making up our set. N numbers follow in the next line, separated by white space. The numbers are positive integers ≤40. The input ends with an N equal to 0.

Output Format

For each test case you should output in a single line a sequence where successive numbers form square sums, or

```
No solution
```

if such a sequence is impossible. If multiple sequences exist, you should output the one that starts with the smallest numbers.

Sample Input / Output

square.in

```
10
2 2 2 2 2 2 2 2 2 2
10
3 3 3 3 3 3 3 3 3 3
15
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0
```

OUTPUT

```
2 2 2 2 2 2 2 2 2 2
No solution
8 1 15 10 6 3 13 12 4 5 11
14 2 7 9
```