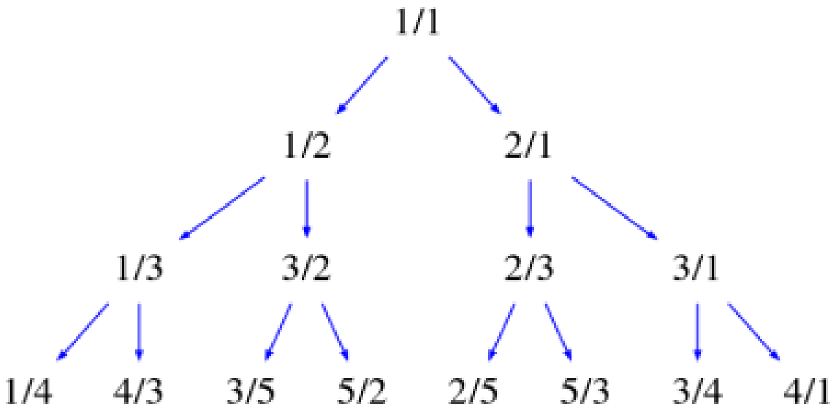
Fractions on Tree

A fraction tree is an infinite binary tree defined as follows:

1. Every node of tree contains a fraction.
2. Root of tree contains the fraction 1/1.
3. Any node with fraction i/j has two children : left child with fraction i/(i+j) and right child with fraction (i+j)/j.

For example, fraction tree up to 3 levels is as shown:



We number the nodes according to increasing levels (root is at level 1) and at any same level, nodes are numbered from left to right. So first node holds the fraction 1/1, second one holds 1/2, third one holds 2/1 fourth one holds 1/3 and so on.

Your task is simple. Given a number n, you are to find the fraction at the nth node.

Input

Every line of the input contains a single number n. You are to find the fraction at nth node of fraction tree. Input file terminates with a 0 which is not to be processed.

Output

For each input, print numerator and denominator of the lowest form of the fraction separated by a /. Output of each case to be done in separate lines.

Example

Input:

```
1
2
3
7
0
```

Output:

```
1/1
1/2
```

```
2/1
3/1
```

Constraints

1 <= number of test cases <= 30000, 1 <= N <= 10^10