

## One...Two...Five!



The set of integers has rarely been a domain of error in everyday conversation. The king, however, is “three blind” and cannot visualize any number containing the digit ‘3’ in its base 10 representation. He does intuitively sense the number between 2 and 4 and compensates for his blindness in the following manner: whenever he wants to state any number containing the digit ‘3’, he will speak a series of numbers until they can all be combined (in the order given) via addition, subtraction, multiplication and division to produce the desired value which contains the digit ‘3’. Mathematical operators work from left to right without any other regard for order of precedence (i.e.  $6 + 7 * 11 = 143 \rightarrow$  a number with a ‘3’).

For example, if the king says “1 2 5”, then a knight will say “3” using the following logic:

$$1 \ 2 \ 5 = 1 * 2 - 5 = 2 - 5 = |2 - 5| = 3$$

Note that there are no negative numbers in optimistic Camelot. Every subtraction will produce a nonnegative result by what is called, in these enlightened times, the absolute value. All division is integer division, i.e.  $7/5 = 1$ . Obviously, if the number zero appears as a divisor, then division will not be attempted.

The court, however, has a problem. Some of the computations produce more than 1 number containing the digit ‘3’! You have been appointed to write a program which computes and displays the most frequently appearing number containing the digit ‘3’. In the event of a tie, use the largest number.

### Input

The input will consist of an unspecified number of lines. Each line will contain at least 1 and at most 9 integers. Every number will be nonnegative and less than 100. A line with a single ‘#’ character will be the end of input.

### Output

For every line of input, print a line showing the number most probably meant by the king as described above. If there is no such number, print “No result”.

#### Sample Input

```
1 2 5
1 1
6 5 1
#
```

#### Sample Output

```
3
No result
30
```