You might have heard the game of 24: given 4 integers, you're to make an expression to get the number 24. For example, given 4, 4, 10, 10, you can write $(10 * 10 - 4)/4 = 24$, given 1, 5, 5, 5, you can write $(5 - 1/5) * 5 = 24$.

In this problem, your task is a little bit harder: count the number of numbers that can be made. Don't forget to count negative numbers and non-integers. You can use binary additions, subtractions, multiplications and divisions with parenthesis (unary operations are not allowed). Numbers cannot be concatenated to form a larger number (e.g. you cannot concatenate 1 and 2 to get 12).

For example, given two 1's, exactly 3 numbers can be made: 1+1=2, 1-1=0, $1 * 1 = 1$. You cannot get 11 or -1.

Input

The input consists of at most 30 test cases. Each case begins with a line containing a single integer n ($1 < n < 7$), the number of integers given. The next line contains n non-negative integers not greater than 10. The last case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and the number of numbers that can be made.

Sample Input

```
2
1 1
3
1 4 7
4
1 2 3 5
0
```

Sample Output

```
Case 1: 3
Case 2: 47
Case 3: 255
```