When we need to find the arithmetic average of some numbers we always tend to keep the result in a floating-point number as in many cases the average is a fraction. But if all the numbers are integers then we should not use floating-point numbers to store the average, as floating-point numbers can lead to all sorts of precision errors. In this problem your job is to find the average of some integers and express it in the form a *or* $\frac{b}{c}$ *or* $a\frac{b}{c}$, where a, b and c are all integers. In this problem we will denote this average as avg.

a) If $|avg| > 1$ and a fractional number then we should print it as $a\frac{b}{c}$

b) If $|avg| < 1$ and a fractional number then we should print it as $\frac{b}{c}$

c) If avg is an integer we should print it as a

d) If avg is negative then the whole result should be preceded by a minus ('−') sign and a space.

e) In the printed fractional part b and c should be relative prime. In other words $\gcd(b, c)$ should be 1.

f) The horizontal bar in the fraction should be formed using only the hyphen ('-'). Its length should be equal to the number of digits in c. If the number of digits in b is less than c then b should be right justified on the horizontal bar.

Input

The input file contains less than 101 sets of input. Each set of input is given in a single line. The description of each line is given below:

The first integer n ($1 \le n \le 100$) of a line denotes how many numbers are to be averaged. It is followed by n numbers, all of which have absolute values less than 10000.

Input is terminated by a case where $n = 0$. This case should not be processed.

Output

For each line of input produce two or four lines of outputs. The first line contains the serial number of the output. The next one or three lines contain the value of the average following the rules specified above. The output lines should not contain any trailing spaces. And there must not be any leading or trailing spaces other than the ones that are required for formatting.

Sample Input

```
3 1 2 3
3 -1 -2 -3
3 1 2 4
4 2 4 6 10
3 -1 -2 -4
10 1 1 1 1 1 1 1 1 1 4
10 1 -1 1 -1 1 -1 -1 1 1 1
10 1 -1 1 -1 1 -1 -1 1 1 -3
0
```

Sample Output

```
Case 1:
2
Case 2:
- 2
Case 3:
 1
2-
 3
Case 4:
 1
5-
 2
Case 5:
  1
- 2-
  3
Case 6:
 3
1--
 10
Case 7:
1
-
5
Case 8:
 1
- -
 5
```