

The university canteen does not want any student to leave the canteen hungry. Therefore, as long as a student is hungry, he can get another portion of food for free. The canteen uses a fixed food portion size, because it would take too much time to first ask a student how much food he wanted. It can happen that a student doesn't finish his last portion of food and the remainder has to be thrown away.

To minimize costs, the manager of the canteen wants to determine a food portion size S such that the amount of food that is wasted is small, but also the number of times the students have to fetch another portion of food is not too big. Note that these two goals can be conflicting:

- By choosing a very small food portion size, one does not waste food, but simultaneously the number of times the students have to fetch food is big.
- By choosing a large food portion size, one can make sure each student has to fetch only one portion, but at the same time it may happen that a large quantity of food is wasted.

The manager of the canteen has collected data about how many units of food each student eats. The problem to be solved can now be formulated mathematically as follows: Let x be the amount of food that is wasted, and y the number of times the students go to fetch food. Then, the goal is to minimize $a \times x + b \times y$, where a, b are weights that represent the relative importance of the two opposing goals. Note that x and y depend on the food portion size S and the quantities of food each student eats. We impose the additional constraint that no student should have to go more than 3 times to fetch food.

Input

The input file contains several test cases. Each test case starts with a line containing an integer n , ($1 \leq n \leq 1000$), the number of students eating in the canteen. The next line contains the values a and b ($1 \leq a, b \leq 10$). The third line of each test case consists of n integers y_1, \dots, y_n ($1 \leq y_i \leq 100$), where y_i is the amount of food student i eats. Input is terminated by $n = 0$.

Output

For each test case print one line containing the costs resulting from an optimal choice of the food portion size. Print each value as a reduced fraction. If the result is an integer, do not print the denominator 1. See the sample output for details.

Note: In the first sample input, the optimal food portion size is 4.5. Note that a food portion size of 3 would yield a smaller cost of 16, the 5th student, however, would have to fetch food 4 times.

Sample Input

```
5
1 1
3 7 1 9 12
3
10 1
11 13 17
2
2 3
6 3
0
```

Sample Output

```
35 / 2
154 / 3
9
```