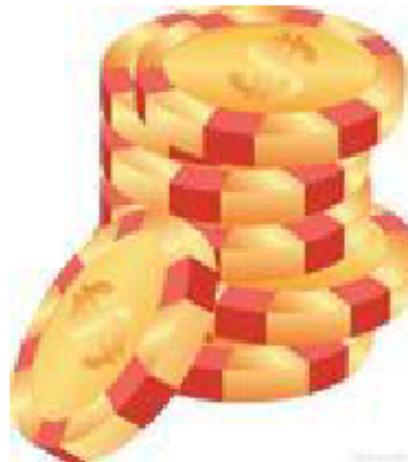


4032 Jewel Trading

You want to sell a diamond to a merchant for a good price. You know so much about how merchant likes the diamond that you have even built a mathematical model for it: He will definitely accept the price p if it's not greater than a certain threshold a , but for a price p higher than it, he must have a think. The higher the price, the lower probability he will accept. Precisely, the probability that he accept price $p > a$ is $1/(1 + (p - a)^b)$, where $b > 1$ is a positive constant in your model.

The exact trading process is as follows: you first propose a price (a non-negative *integer*), then the merchant decides whether to accept. If he accepts, the trade is over and you have no chance to regret. If he does not accept, you propose another price, and so on. You know that the merchant would get angry if you always propose unacceptable high prices, so you promised that the n -th proposal (if there is) is always not greater than a (which he can accept for sure).

Write a program to find an optimal way to propose prices to maximize your expected earning (i.e. the final price).



Input

The input consists of several test cases. Each case is described by two positive integers n , a , and a real number b ($1 \leq n \leq 100$, $1 \leq a \leq 1000$, $1 < b < 10$), b is given to up to three decimal places. The last test case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and the expected earning, to two decimal places. It is guaranteed that the maximal earning exists.

Sample Input

```
1 10 2
10 33 3.14
0
```

Sample Output

```
Case 1: 10.00
Case 2: 34.41
```