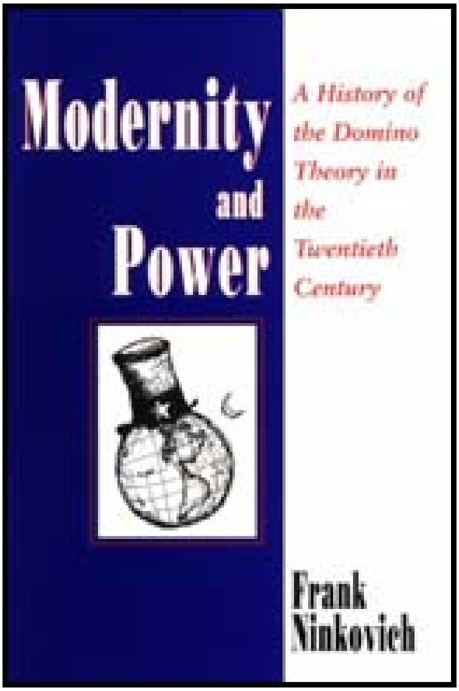

You are trying to set up a straight line of dominos, standing on end, to be pushed over later for your entertainment. (Sure, it seems pointless to set something up only to knock it down again, but you have some strange hobbies) The tricky thing about setting dominos, however, is that if you make a mistake and knock one over as you place it, it will knock down any adjacent line of consecutive dominos on one side of it, partially ruining your work.

For instance, if you've already placed dominos in the pattern ‘`DD__DxDDD_D`, and you try placing a domino at position ‘`x`’, there is a chance it will fall and knock over the domino to the left or the three dominos to its right, forcing you to place them again.

This human error is somewhat unavoidable, but you *can* make the odds somewhat more favourable by using a domino-placing technique that leads to dominos falling in one direction more often than in the other.

Given the number of dominos you are trying to set up, and the probability that you'll knock over any individual domino either to the left or to the right while placing it, determine the average number of dominos you'll need to place before you finish. Assume that you're using an optimal placement strategy.

Input

Input will consist of up to 100 cases. Each case consists of one line of input. It will contain the number of dominos to place, n, $1 \leq n \leq 1000$, followed by nonnegative values P_l and P_r, indicating the probability of any domino falling to the left ot to the right when placed. You may assume $0 < P_l + P_r \leq 0.5$.

The last test case is followed by a line containing a single ‘`0`’.

Output

For each case, output the expected number of dominos that will need to be placed before you finish, accurate to two digits after the decimal.

Sample Input

```
10 0.25 0.25
10 0.1 0.4
10 0.0 0.5
0
```

Sample Output

```
46.25
37.28
20.00
```