Problem J
Jackpot

The thunderingly fun new game show of ’Jackpot’ has bounded onto televisio screens up and down the land and is the next big thing. It runs along a simple premise: there are a number of doors, behind one of which is hidden a monetary prize and behind all of the others of which are hidden goats. Each contestant chooses how many doors will be opened before they pick a door to look at. The prize decreases as each door is opened. If they manage to open the door and find the money they get to keep it!

Not content with guessing, you’ve decided to try and work out the best number of doors to open to maximise your expected profit, where expected profit is the probability of winning multiplied by the remaining prize fund.

You’ve managed to find out that the remaining prize fund P_R is a function of the initial prize fund m, the number of doors opened d and a scaling factor f, ie.

$$P_R = m \qquad (d \qquad f)^2$$

Given the number of doors, initial prize fund and the scaling factor can you work out how many doors should be opened before you try to pick a door to maximise your expected profit?

Input

One line of input containing:

- One integer n (1   n   20; 000), the number of doors.
- One integer m (1 m 10; 000; 000) the initial prize fund.
- One floating point f (0 < f 1), the scaling factor.

There will always only be one door with the best value.

Output

One line containing

- One floating point number, the value of the expected payout.

The output must be accurate to an absolute or relative error of at most 10^{6}.

Sample Input 1

```
4 100    1
```

Sample Output 1

```
91.0
```

Sample Input 2

```
20000   100200300       0.8
```

Sample Output 2

```
5628.83998462
```