Problem E: Critical Hits

In a traditional RPG game, each time a character attacks it has a chance of performing a *Critical Hit*. Typically this value is expressed as a percentage - for example, the character has a 5% chance to perform a critical hit with each swing. Seems straightforward, right? :)

However, some complain that this approach can lead to long periods of time without getting a critical hit, and at other times hitting several in a short span of time. To alleviate this, a new algorithm is proposed:

- The first attack shall have a critical hit chance of X
- The second attack, if the first is not a critical, shall have a critical hit chance of $2X$
- The third attack, if the first and second attacks are not criticals, shall have a critical hit chance of $3X$
- This pattern continues until a critical hit occurs, at which point the critical hit chance is reset to X and the cycle begins again. If the critical hit chance ever exceeds 100%, then a critical hit is guaranteed.

This new algorithm helps eliminate long strings of attacks without a critical hit. Your job is to help implement the algorithm by finding the right value for X. In particular, suppose that a player currently has a critical hit chance of P%. This means that, if he were to attack forever, P% of his attacks would be critical hits. You want to find the value X such that the new algorithm also has P% of the attacks as critical hits.

Input Specification:

You will be given a series of test cases, one per line. Each line contains a double value $0 < P < 1$, with exactly 3 decimal places. Input ends on EOF.

Output Specification:

For each value of P in the input, output the corresponding value of X, rounded to 3 decimal places.

Sample Input:

```
0.900
```

Sample Output:

```
0.889
```