You've found a solution to an implementation-heavy geometry problem that requires typing out N lines of code. Annoyingly, you only have a $P\%$ chance of typing out any given line without a mistake, and your code will only be accepted if all N lines are correct. The chance of making a mistake in one line is independent of the chance of making a mistake in any other line.

You realize there might be a solution which only requires $N - 1$ lines (each also having a $P\%$ chance of being typed correctly). However, instead of thinking about that, you could also just type out the N-line solution more carefully to increase P. How much would P have to increase to yield the same chance of success as needing to type one fewer line of code?

Constraints

$1 \leq T \leq 100$
$2 \leq N \leq 1{,}000$
$1 \leq P \leq 99$

Input Format

Input begins with an integer T, the number of test cases. Each case is a single line containing the integers N and P.

Output Format

For the ith test case, print "`Case #i: `" followed by how much higher P would need to be to make spending your time typing carefully be as successful as typing one line fewer with your original P.

Your answer will be accepted if it is within an absolute or relative error of 10^{-6}.

Sample Explanation

In the first case, you initially need to type 2 lines. You can either type just 1 line with a 50% success rate, or you could improve your typing accuracy to $\sqrt{50\%} \approx 70.710678\%$, at which point you'd have a $\sqrt{50\%}^2 = 50\%$ chance of successfully typing the original 2 lines. So you would need to increase P by $70.710678 - 50 = 20.710678$ for both approaches to have an equal chance of success.

Sample Input

```
4
2 50
3 10
13 37
950 95
```

Sample Output

```
Case #1: 20.710678118654748
Case #2: 11.544346900318839
Case #3: 2.940819927087601
Case #4: 0.005129467915043762
```