

## Problem E. Rumpf

Input file: `rumpf.in`  
Output file: `rumpf.out`  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

You have developed a brand new method of fishing which you call *Monte-Carlo fishing*. For simplification, let's assume you're trying to catch a fish in a quadratic lake, with a coordinate system coinciding with the lake shores in such a way that the lake is the unit square: points  $(x, y)$  such that  $0 < x, y < 1$ .

In order to Monte-Carlo fish, you drop  $n$  stones at independent uniformly random locations inside the lake. A net is connected to all stones, and ends up catching all fish that lie within or on the boundary of the *convex hull* of the locations of the stones: the smallest convex polygon containing all locations of the stones.

A gold fish is located at point  $(x_0, y_0)$  of the lake. What is the probability of catching this fish with the Monte-Carlo method described above?

### Input

The first line of the input file contains the number  $n$  of randomly placed stones,  $3 \leq n \leq 20$ . The second line of the input file contains the location of the gold fish as two floating-point numbers  $x_0$  and  $y_0$  with at most 2 digits after the decimal point,  $0 < x_0, y_0 < 1$ .

### Output

Output one floating-point number: the probability that the convex hull of the locations of the  $n$  stones contains the gold fish at  $(x_0, y_0)$ . Your output will be considered correct if it differs from the answer by at most  $10^{-7}$ .

### Example

<code>rumpf.in</code>	<code>rumpf.out</code>
3 0.5 0.5	0.25