B. Marathon

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Valera takes part in the Berland Marathon. The marathon race starts at the stadium that can be represented on the plane as a square whose lower left corner is located at point with coordinates $(0, 0)$ and the length of the side equals a meters. The sides of the square are parallel to coordinate axes.

As the length of the marathon race is very long, Valera needs to have extra drink during the race. The coach gives Valera a bottle of drink each d meters of the path. We know that Valera starts at the point with coordinates $(0, 0)$ and runs counter-clockwise. That is, when Valera covers a meters, he reaches the point with coordinates $(a, 0)$. We also know that the length of the marathon race equals $nd + 0.5$ meters.

Help Valera's coach determine where he should be located to help Valera. Specifically, determine the coordinates of Valera's positions when he covers $d, 2 \cdot d, ..., n \cdot d$ meters.

Input

The first line contains two space-separated real numbers a and d $(1 \le a, d \le 10^5)$, given with precision till 4 decimal digits after the decimal point. Number a denotes the length of the square's side that describes the stadium. Number d shows that after each d meters Valera gets an extra drink.

The second line contains integer n $(1 \le n \le 10^5)$ showing that Valera needs an extra drink n times.

Output

Print n lines, each line should contain two real numbers x_i and y_i, separated by a space. Numbers x_i and y_i in the i-th line mean that Valera is at point with coordinates (x_i, y_i) after he covers $i \cdot d$ meters. Your solution will be considered correct if the absolute or relative error doesn't exceed 10^{-4}.

Note, that this problem have huge amount of output data. Please, do not use cout stream for output in this problem.

Examples

input
2 5 2

output
1.0000000000 2.0000000000 2.0000000000 0.0000000000

input
4.147 2.8819 6

output
2.8819000000 0.0000000000 4.1470000000 1.6168000000 3.7953000000 4.1470000000 0.9134000000 4.1470000000 0.0000000000 2.1785000000 0.7034000000 0.0000000000