#include <algorithm>

#include <cassert>

#include <cstdio>

#include <iostream>

#include <vector>

using namespace std;

double W, H, S;

double cost(double a, double b, double c) //Cost Function

{

double alen = ((a+b)\*S + (b-a))/2;

double blen = ((b+c)\*S + (c-b))/2;

return (alen\*alen + blen\*blen - b\*b\*S\*S)/2;

}

int main() {

int N;

while (cin >> W >> H >> N) { //Inputs

S = H/W;

vector<double> v(N+2);

v[0] = 0;

v[1] = 1;

for (int i = 2; i <= N+1; i++) v[i] = 2/(1-S\*S) \* v[i-1] - v[i-2];

for (int i = 0; i <= N+1; i++) v[i] \*= W/v.back();

assert(v[1] > 1.01e-3);

assert(v[v.size()-2] < W-1.01e-3);

double ret = cost(0, 0, v[1]) + cost(v[v.size()-2], W, W);

for (int i = 1; i+1 < v.size(); i++) ret += cost(v[i-1], v[i], v[i+1]);

printf("%.6lf\n", ret); //Output : Optimum Overall Cost

for (int i = 1; i+1 < v.size() && i <= 10; i++) {

if (i > 1) putchar(' ');

printf("%.6lf", v[i]); //Output : Placements of Shafts

}

putchar('\n');

}

return (0);

}