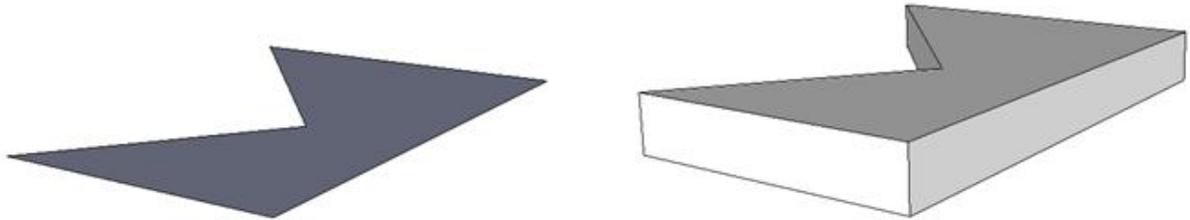


## 12305 Polishing a Extruded Polygon

You have a simple polygon on  $z = 0$  plane, then you make an extrusion so that its height becomes  $h$  unit (i.e. the top surface is now on  $z = h$ ). We call the extruded polygon *the model*. Note that the model is a solid.



After that, you apply  $m$  “cutting” operations on the model. Each operation is described by four integers  $a, b, c, d$ , that is to remove the set of points  $(x_p, y_p, z_p)$  satisfying  $ax_p + by_p + cz_p + d > 0$ . Finally, you need to calculate the volume and surface area of the polished model.

### Input

There will be at most 10 test cases. Each case begins with three integers  $n, h, m$  ( $1 \leq n \leq 1000$ ,  $1 \leq h \leq 10$ ,  $1 \leq m \leq 100$ ), the number of vertices of the polygon, the height after the extrusion, and the number of operations. The next  $n$  lines contain the vertices of the polygon in counter-clockwise order (note that they’re in  $z = 0$  plane, so the  $z$  coordinates are not given. Coordinates are real numbers whose absolute values are not greater than 1000). Each of the next  $m$  lines contains four real numbers  $a, b, c, d$ , describing one operation. At least one of  $a, b, c$  is non-zero. The last test case is followed by a line with  $n = h = m = 0$ , which should not be processed.

### Output

For each test case, print the volume and surface area of the original model and polished model *after every operation* (i.e. output  $m + 1$  lines in total). Each number should be rounded to three digits after the decimal point. It is guaranteed that the volume and surface area is always strictly positive. To reduce the impact of floating-point errors, each number you print could differ from the standard output by up to 0.005.

### Sample Input

```
4 5 2
0 0
10 0
10 10
0 10
1 0 0 -9
0 0 1 -3
5 5 1
0 0
4 0
4 4
2 2
```

```
0 4
0 -1 0 3
5 5 1
0 0
4 0
4 4
2 2
0 4
0 1 0 -3
0 0 0
```

### Sample Output

```
500.000 400.000
450.000 370.000
270.000 294.000
60.000 112.284
5.000 36.142
60.000 112.284
55.000 96.142
```