E. Cannon

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Bertown is under siege! The attackers have blocked all the ways out and their cannon is bombarding the city. Fortunately, Berland intelligence managed to intercept the enemies' shooting plan. Let's introduce the Cartesian system of coordinates, the origin of which coincides with the cannon's position, the Ox axis is directed rightwards in the city's direction, the Oy axis is directed upwards (to the sky). The cannon will make n more shots. The cannon balls' initial speeds are the same in all the shots and are equal to V, so that every shot is characterized by only one number $alpha_i$ which represents the angle at which the cannon fires. Due to the cannon's technical peculiarities this angle does not exceed 45 angles ($\pi / 4$). We disregard the cannon sizes and consider the firing made from the point$(0, 0)$.

The balls fly according to the known physical laws of a body thrown towards the horizon at an angle:

$$v_x(t) = V \cdot cos(alpha)$$
$$v_y(t) = V \cdot sin(alpha) - g \cdot t$$
$$x(t) = V \cdot cos(alpha) \cdot t$$
$$y(t) = V \cdot sin(alpha) \cdot t - g \cdot t^2 / 2$$

Think of the acceleration of gravity g as equal to 9.8.

Bertown defends m walls. The i-th wall is represented as a vertical segment $(x_i, 0)$ - (x_i, y_i). When a ball hits a wall, it gets stuck in it and doesn't fly on. If a ball doesn't hit any wall it falls on the ground ($y = 0$) and stops. If the ball exactly hits the point (x_i, y_i), it is considered stuck.

Your task is to find for each ball the coordinates of the point where it will be located in the end.

Input

The first line contains integers n and V ($1 \le n \le 10^4, 1 \le V \le 1000$) which represent the number of shots and the initial speed of every ball. The second line contains n space-separated real numbers $alpha_i$ ($0 < alpha_i < \pi / 4$) which represent the angles in radians at which the cannon will fire. The third line contains integer m ($1 \le m \le 10^5$) which represents the number of walls. Then follow m lines, each containing two real numbers x_i and y_i ($1 \le x_i \le 1000, 0 \le y_i \le 1000$) which represent the wall’s coordinates. All the real numbers have no more than 4 decimal digits. The walls may partially overlap or even coincide.

Output

Print n lines containing two real numbers each — calculate for every ball the coordinates of its landing point. Your answer should have the relative or absolute error less than 10^{-4}.

Examples

input

```
2 10
0.7853
0.3
3
5.0 5.0
4.0 2.4
6.0 1.9
```

output

```
5.000000000 2.549499369
4.000000000 0.378324889
```

input

```
2 10
0.7853
0.3
2
4.0 2.4
6.0 1.9
```

output

```
10.204081436 0.000000000
4.000000000 0.378324889
```