The organizers of CEOI 2011 are planning to hold a party with lots of balloons. There will be $n$ balloons, all sphere-shaped and lying in a line on the floor.

The balloons are yet to be inflated, and each of them initially has zero radius. Additionally, the $i$-th balloon is permanently attached to the floor at coordinate $x_i$. They are going to be inflated sequentially, from left to right. When a balloon is inflated, its radius is increased continuously until it reaches the upper bound for the balloon, $r_i$, or the balloon touches one of the previously inflated balloons.



Figure 1: The balloons from the example test, after being fully inflated.

The organizers would like to estimate how much air will be needed to inflate all the balloons. You are to find the final radius for each balloon.

## Input

The first line of the standard input contains one integer $n$ ($1 \leqslant n \leqslant 200\,000$) — the number of balloons. The next $n$ lines describe the balloons. The $i$-th of these lines contains two integers $x_i$ and $r_i$ ($0 \leqslant x_i \leqslant 10^9$, $1 \leqslant r_i \leqslant 10^9$). You may assume that the balloons are given in a strictly increasing order of the $x$ coordinate.

In test data worth 40 points an additional inequality $n \leqslant 2\,000$ holds.

## Output

Your program should output exactly $n$ lines, with the $i$-th line containing exactly one number — the radius of the $i$-th balloon after inflating. Your answer will be accepted if it differs from the correct one by no more than 0.001 for each number in the output.

## Example

For the input data:                     the correct result is:

```
3                                       9.000
0 9                                     1.000
8 1                                     4.694
13 7
```

**Hint:** To output a `long double` with three decimal places in C/C++ you may use `printf("%.3Lf\n", a);` where `a` is the `long double` to be printed. In C++ with streams, you may use `cout << fixed << setprecision(3);` before printing with `cout << a << "\n";` (and please remember to include the `iomanip` header file). In Pascal, you may use `writeln(a:0:3);`. You are advised to use the `long double` type in C/C++ or the `extended` type in Pascal, this is due to the greater precision of these types. In particular, in every considered correct algorithm no rounding errors occur when using these types.