| problem | špilja | tetris |
|---|---|---|
| source file | `spilja.pas`<br>`spilja.c`<br>`spilja.cpp` | `tetris.pas`<br>`tetris.c`<br>`tetris.cpp` |
| input data | stdin | |
| output data | stdout | |
| memory limit (heap) | 32 MB | 96 MB |
| memory limit (stack) | 1 MB | |
| time limit (pentium4 1.6 ghz) | 1 second | 3 seconds |
| points | **80** | **100** |
| | **180** | |

In a cave near Mirko's village lived his ancestors thousands of years ago.

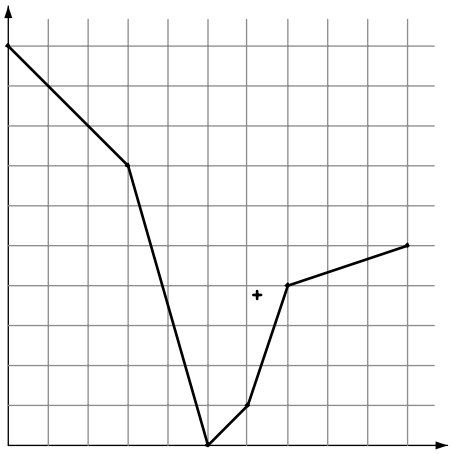
Mirko came to a conclusion that he is the only one who can find the remains of this ancient civilization so he started with preparations to explore. Therefore, he bought new pants, boots, a shovel and a hammer. Just before research he realized that there was no power supply in the cave and that he has to buy some lamps. But as he spent all the money for buying an up-to-date hammer, he realized that he had money only for one lamp.

As Mirko wants to choose a spot from where he can see the entire floor, help him to determine a **minimum height** to which he must raise a lamp in order **to illuminate the entire floor** of the cave.

We will imagine the cave's floor as **a broken line** in a coordinate system consisting of N peaks $t_1$, $t_2$ $t_N$ and lengths which connect consecutive peaks.

The floor always goes **from left to right**, i.e. for every i=1, ..., N-1 x-coordinate of $t_i$ is **smaller** than x-coordinate of $t_{i+1}$.

Example (possible solution for third test example)



The lamp should be placed somewhere in a spot "**above**" cave's floor so that it **illuminates the entire floor**. To put it more precisely, x coordinate of the lamp must be placed between x coordinate of the first and last point of the floor (inclusive), and y coordinate of the lamp must be bigger than or equal to y coordinate of the floor point with the same x coordinate.

We can say that the lamp illuminates the entire floor, if **for every point on the floor** we can say that the length which connects this point with a lamp **does not perforate** the broken line which represents the floor. However, it is allowed for the length and the broken line to connect at some points or along certain segments.

Write a program that will determine **the smallest possible height** on which we can place a lamp so that it illuminates the entire floor.

You can presume that the result will always be less than or equal to 1,000,000 (one million).

## input data

In the first line,there is an integer N, $2 \leq N \leq 5000$, number of peaks of the floor.

In each of the next N lines, there are integers $X_i$ i $Y_i$, $0 \leq X_i, Y_i \leq 100{,}000$, in $i^{th}$ line numbers $X_i$ i $Y_i$ i.e. coordinates of $i^{th}$ peak. Numbers $X_i$ will be in **ascending** order.

## output data

In first and only line you should write y coordinate where you will put a lamp, real decimal number **rounded to 2 decimal places**.

Your output value must be within **0.01** absolute error of the correct value.

## examples

**input**

```
6
0 0
10 0
11 1
15 1
16 0
25 0
```

**output**

```
3.00
```

**input**

```
6
1 1
4 2
5 0
9 2
12 3
16 4
```

**output**

```
2.00
```

**input**

```
6
0 10
3 7
5 0
6 1
7 4
10 5
```

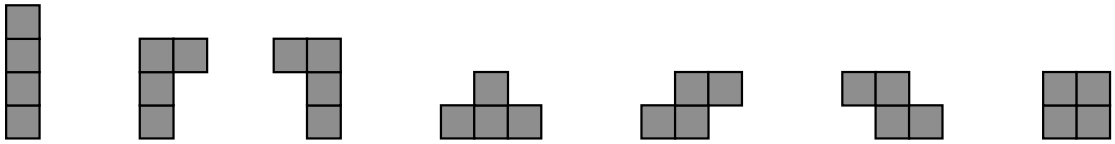
**output**

```
3.75
```

Johnny plays a very interesting game, very similar to the popular Tetris.

Game field is rectangular, 3 columns wide and 1000 rows high. Tiles **fall from the top** and **have to be arranged** keeping **the total height of all the tiles minimal**.

There are **7** possible tiles, marked with numbers 1 through 7 (see picture below).




Johnny **can rotate** each tile that appears. Rotation are made by angles that are **multiples of 90 degrees**. The tile then **falls downwards until stopped by some tile inserted previously or by the bottom of the game field**.

Johnny can choose **any horizontal poisition** of the tile, keeping in mind that **the whole tile remains within the game filed**. I.e. tile no. 1 cannot be rotated (because it is to wide for the game field), but can be inserted in any of the three columns. If Johnny decides to rotate tile no. 5 then it can be inserted in two ways: into columns 1 and 2 or into columns 2 and 3.

After the tile is inserted into the game field and starts to decend, **additional moves or rotations are not allowed**. Unlike the original tetris, **filled rows are not removed from the field**.

**A sequence** of tiles to be inserted into the field **is given**.

Write a program that will calculate **the minimal possible height of all the tiles inserted**. In other words, find the least integer K such that, by arranging the tiles in a specific manner, they can all fit in the bottom K rows of the game field.

## input data

In first line, there is an integer N, $1 \leq N \leq 100$, number of tiles in the game.

In each of the next N lines, there is a number which denote mark label of that tail, from the first to the last tail.

## output data

In first and only line you should write the minimal possible height as described in the text.

## examples

**input**

```
3
4
5
1
```

**output**

```
5
```

**input**

```
4
5
3
1
2
```

**output**

```
6
```

**input**

```
5
1
6
3
4
2
```

**output**

```
8
```