



**Croatian National Competition 2007**  
**Zagreb, April 16-21**

**Senior category**  
**Competition day 1**

<b>TASK</b>	<b>KOLICA</b>	<b>POTPUNI</b>	<b>OGRADA</b>
<b>input</b>	standard input		
<b>output</b>	standard output		
<b>time limit</b>	1 seconds	0.7 seconds	1 second
<b>memory limit</b>	64 MB		
<b>points</b>	<b>40</b>	<b>50</b>	<b>60</b>
	<b>150</b>		



A number of shopping carts filled with explosives are floating in a coordinate system, in one of the four main directions (up, down, left or right). All carts are moving at a speed of one unit per second. Movement is continuous; for example, in one third of a second, a cart travels one third of a unit.

When two or more carts collide (are at the same place at the same time), there is an explosion and all carts taking part in the collision explode and cease to exist.

Write a program that, given the starting points and directions of all carts, determines which carts never explode.

### INPUT

The first line of input contains an integer  $N$  ( $2 \leq N \leq 500$ ), the number of carts.

Each of the following  $N$  lines contains two integers and a string. Each pair of integers describes the starting coordinates of one cart (between 0 and 100 000 000, inclusive), and the string describes the direction in which the cart is moving ("gore" for up, "dolje" for down, "lijevo" for left, or "desno" for right).

No two carts will start at the same coordinates.

### OUTPUT

Output the indices of all carts which never explode, sorted in ascending order, one index per line. The first cart in the input is labeled 1, the second is labeled 2 etc. If no carts survive, output "nema".

### EXAMPLE TEST CASES

#### input

```
4
5 5 dolje
5 6 lijevo
5 7 desno
5 8 gore
```

#### output

```
1
2
3
4
```

#### input

```
5
3 3 dolje
1 1 desno
5 1 lijevo
100000 500000 desno
900000 500000 lijevo
```

#### output

```
nema
```

#### input

```
3
10 0 gore
0 10 desno
15 5 lijevo
```

#### output

```
2
```



To make everyday life better, it has been decided that a most unusual device be placed in a public location. The device interacts with users with a keyboard and a screen, and **stores an integer internally**, which is initially 1.

The device is used as follows:

- The user **enters an integer** using the keyboard.
- The device **multiplies** the internal number with the entered number and stores the result/
- The device outputs whether the result is a **perfect square** (a perfect square is an integer which is the square of some integer) on the screen. If it is, the user cheers, jumps, kicks his heels in the air before landing and, happy on the inside but careful not to show off, discretely moves away from the device.

Write a program that, given the numbers users enter in order, determine which users the device makes happy.

### INPUT

The first line of input contains an integer  $N$  ( $1 \leq N \leq 500\,000$ ), the number of users of the device.

The following  $N$  lines contain integers entered by the users, in order in which they use the device, one per line. The integers will be between 1 and 1 000 000 (a million).

Note that the stored integer may not fit in a 64-bit integer type.

### OUTPUT

Output the answer each of the users gets, in order in which they use the device. The device outputs "DA" if the stored integer is a complete square, and "NE" if it is not.

### EXAMPLE TEST CASES

input	input	input
5	7	7
2	2	999999
2	3	111111
2	6	333333
2	15	111111
2	35	3
	21	27
	64	243
output	output	output
NE	NE	NE
DA	NE	DA
NE	DA	NE
DA	NE	NE
NE	NE	DA
	DA	NE
	DA	DA



Mirko and Slavko lived next door to each other in a small village and they were inseparable friends. One day they had a fight (Slavko went crazy because he found out Mirko was stealing rakija from his basement and eggs from the henhouse).

When Mirko woke up the next morning, he was surprised to see that, during the night, Slavko had built a fence between their houses. Slavko's fence consists of  $N$  boards of varying heights placed in one row.

Mirko, furious, decided to build an even bigger fence on his side, in front of Slavko's, so he doesn't have to look at Slavko's fence. In front of each of Slavko's  $N$  boards he wants to put one of his own which will cover Slavko's board (in other words, he wants each of his boards to be at least as high as Slavko's board it is in front of).

But Mirko doesn't have any boards at home (he forgot to steal them from Slavko) so he called his friend Lea who (conveniently) works in a sawmill. Lea brought  $N$  boards; each of the boards has a **height and price**. Unfortunately, when they started raising the fence, they noticed that Lea's boards may not be high enough to cover entire Slavko's fence. So they agreed they will build the fence anyway, but Mirko will **pay only those boards which will cover Slavko's boards**.

Write a program which will help Lea arrange the boards and make the largest possible profit.

### INPUT

The first line contains an integer  $N$  ( $1 \leq N \leq 100\,000$ ), the number of boards in Slavko's fence.

The second line contains  $N$  integers, the heights of the boards in Slavko's fence. The height of each board is between 1 and 10 000, inclusive.

Each of the following  $N$  lines contains two integers, the height and price of one board which Lea brought. Both numbers will be between 1 and 10 000, inclusive.

Lea's boards are numbered 1 to  $N$ , in the order in which they are given.

### OUTPUT

The first line of output should contain the largest profit Lea can achieve.

The second line should contain the indices of Lea's boards in order in which they put them in front of Slavko's fence – the first number is the index of Lea's board which they will put in front of Slavko's first board, and so on.

In the second line write the indexes of Lea's boards in order in which they put them in front of Slavko's fence. So the first number is the index of Lea's board they will put in front of Slavko's first board, and so on.

**Note:** the optimal arrangement of the boards may not be unique.



## EXAMPLE TEST CASES

**input**

5

400 200 500 600 400

200 400

300 600

400 200

500 800

600 100

**output**

1700

4 2 1 5 3

**input**

8

70 40 80 70 50 60 20 30

30 10

30 20

30 25

60 15

60 5

50 30

40 5

40 5

**output**

95

8 6 1 7 5 4 2 3

**Clarification for first example:** the heights of the boards which Mirko puts in front of Slavko's fence are in order 500, 300, 200, 600 i 400. Lea gets 800, 600, 0, 100 and 200 for them.