G. Queue

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

On a cold winter evening our hero Vasya stood in a railway queue to buy a ticket for Codeforces championship final. As it usually happens, the cashier said he was going to be away for 5 minutes and left for an hour. Then Vasya, not to get bored, started to analyze such a mechanism as a queue. The findings astonished Vasya.

Every man is characterized by two numbers: a_i, which is the importance of his current task (the greater the number is, the more important the task is) and number c_i, which is a picture of his conscience. Numbers a_i form the permutation of numbers from 1 to n.

Let the queue consist of $n - 1$ people at the moment. Let's look at the way the person who came number n behaves. First, he stands at the end of the queue and the does the following: if importance of the task a_i of the man in front of him is less than a_n, they swap their places (it looks like this: the man number n asks the one before him: "Erm... Excuse me please but it's very important for me... could you please let me move up the queue?"), then he again poses the question to the man in front of him and so on. But in case when a_i is greater than a_n, moving up the queue stops. However, the man number n can perform the operation no more than c_n times.

In our task let us suppose that by the moment when the man number n joins the queue, the process of swaps between $n - 1$ will have stopped. If the swap is possible it necessarily takes place.

Your task is to help Vasya model the described process and find the order in which the people will stand in queue when all the swaps stops.

Input

The first input line contains an integer n which is the number of people who has joined the queue ($1 \le n \le 10^5$). In the next n lines descriptions of the people are given in order of their coming — space-separated integers a_i and c_i ($1 \le a_i \le n$, $0 \le c_i \le n$). Every description is located on s single line. All the a_i's are different.

Output

Output the permutation of numbers from 1 to n, which signifies the queue formed according to the above described rules, starting from the beginning to the end. In this succession the i-th number stands for the number of a person who will stand in line on the place number i after the swaps ends. People are numbered starting with 1 in the order in which they were given in the input. Separate numbers by a space.

Examples

input

```
2
1 0
2 1
```

output

```
2 1
```

input

```
3
1 3
2 3
3 3
```

output

```
3 2 1
```

input

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

output

```
3 1 5 4 2
```