Problem F: Keep the Customer Satisfied

Simon and Garfunkel Corporation (SG Corp.) is a large steel-making company with thousand of customers. Keeping the customer satisfied is one of the major objective of Paul and Art, the managers.

Customers issue orders that are characterized by two integer values q, the amount of steel required (in tons) and d, the due date (a calender date converted in seconds). The due date has to be met if SG Corp. accepts the order. Stated another way, when an order is accepted, the corresponding amount of steel has to be produced before its due date. Of course, the factory can process no more than one order at a time.

Although the manufacturing process is rather complex, it can be seen as a single production line with a constant throughput. In the following, *we assume that producing q tons of steel takes exactly q seconds* (*i.e.*, throughput is 1). The factory runs on a monthly production plan. Before the beginning of the month, all customers' orders are collected and Paul and Art determine which of them are going to be *accepted* and which ones are to be *rejected* in the next production period. A production schedule is then designed. To keep customers satisfied, Paul and Art want to *minimize the total number of orders that are rejected*. In the following, we assume that the beginning of the next production plan (*i.e.*, the first day of the next month) corresponds to date 0.

Hogdson and Moore have been appointed as Chief Scientific Officers and you are requested to help them to compute an optimal solution and to build a schedule of all accepted orders (starting time and completion time).

Small Example

Consider the following data set made of 6 orders $J_1, ..., J_6$. For a given order, J_j, q_j denotes the amount of steel required and d_j is the associated due date.

Order	q_j	d_j
J_1	6	8
J_2	4	9
J_3	7	15
J_4	8	20
J_5	3	21
J_6	5	22

You can check by hand that all orders cannot be accepted and it's very unlikely you could find a solution with less than two rejected orders. Here is an optimal solution: Reject J_1 and J_4, accept all other orders and process them as follows.

Accepted Order	Starting Time	Completion Time
J_2	0	4
J_3	4	11
J_5	11	14
J_6	14	19

Note that the production line is never idle.

Input Data

Each test case is described by one `input` file that contains all the relevant data: The first line contains the number n of orders (n can be as large as 800000 for some test cases). It is followed by n lines. Each of which describes an order made of two integer values: the amount of steel (in tons) required for the order (lower than 1000) and its due date (in seconds; lower than 2×10^6).

Sample Input

```
6
7 15
8 20
6 8
4 9
3 21
5 22
```

Output

You are required to compute an optimal solution and your program has to write the number of orders that are accepted.

Sample Output

A valid output corresponding to the input file is then :

```
4
```

Some Hints from Hogdson and Moore

- Hogdson and Moore claim that it is optimal to sequence *accepted* orders in non-decreasing order of due dates.
- They also claim that there is an optimal solution such that for any two orders J_u and J_v with $q_u > q_v$ and $d_u < d_v$, if J_u is accepted then J_v is also accepted.
- Finally, Hogdson and Moore advise you to "Keep the Customer Satisfied"

Keep the Customer Satisfied

```
Gee but it's great to be back home
Home is where I want to be.
I've been on the road so long my friend,
And if you came along
I know you couldn't disagree.

It's the same old story
Everywhere I go,
I get slandered,
Libeled,
I hear words I never heard
In the bible
And I'm on step ahead of the shoe shine
Two steps away from the county line
Just trying to keep my customers satisfied,
Satisfied.

Deputy sheriff said to me
Tell me what you come here for, boy.
You better get your bags and flee.
You're in trouble boy,
And you're heading into more.
```

© Simon & Garfunkel