As you must have experienced, instead of landing immediately, an aircraft sometimes waits in a holding loop close to the runway. This holding mechanism is required by air traffic controllers to space apart aircraft as much as possible on the runway (while keeping delays low). It is formally defined as a "holding pattern" and is a predetermined maneuver designed to keep an aircraft within a specified airspace (see Figure 1 for an example).

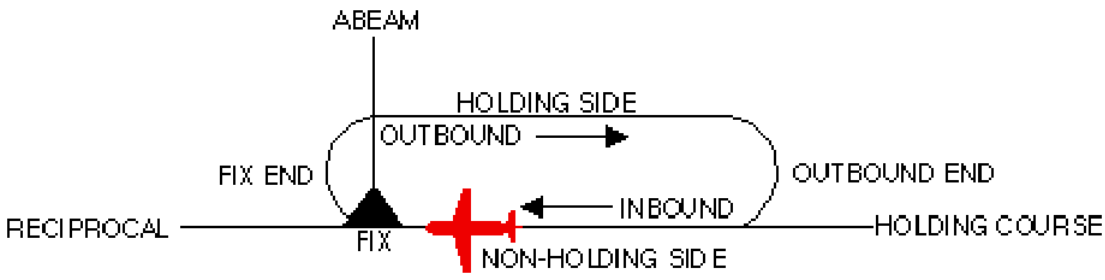

Figure 1: A simple Holding Pattern as described in a pilot text book.

Jim Tarjan, an air-traffic controller, has asked his brother Robert to help him to improve the behavior of the airport.

The TRACON area

The Terminal Radar Approach CONtrol (TRACON) controls aircraft approaching and departing when they are between 5 and 50 miles of the airport. In this final scheduling process, air traffic controllers make some aircraft wait before landing. Unfortunately this "waiting" process is complex as aircraft follow predetermined routes and their speed cannot be changed. To reach some degree of flexibility in the process, the basic delaying procedure is to make aircraft follow a holding pattern that has been designed for the TRACON area. Such patterns generate a constant prescribed delay for an aircraft (see Figure 1 for an example). Several holding patterns may exist in the same TRACON.

In the following, we assume that there is a *single runway* and that when an aircraft enters the TRACON area, it is assigned an *early landing time*, a *late landing time* and a possible holding pattern. The early landing time corresponds to the situation where the aircraft does not wait and lands as soon as possible. The late landing time corresponds to the situation where the aircraft waits in the prescribed holding pattern and then lands at that time. We assume that an aircraft enters at most one holding pattern. Hence, the early and late landing times are the only two possible times for the landing.

The *security gap* is the minimal elapsed time between consecutive landings. The objective is to *maximize the security gap*. Robert believes that you can help.

Example

Assume there are 10 aircraft in the TRACON area. Table 1 provides the corresponding early and late landing times (columns "Early" and "Late").

Aircraft	Early	Late	Solution
A_1	44	156	Early
A_2	153	182	Early
A_3	48	109	Late
A_4	160	201	Late
A_5	55	186	Late
A_6	54	207	Early
A_7	55	165	Late
A_8	17	58	Early
A_9	132	160	Early
A_{10}	87	197	Early

Table 1: A 10 aircraft instance of the problem.

The maximal security gap is 10 and the corresponding solution is reported in Table 1 (column "Solution"). In this solution, the aircraft land in the following order: A_8, A_1, A_6, A_{10}, A_3, A_9, A_2, A_7, A_5, A_4. The security gap is realized by aircraft A_1 and A_6.

Input

The input file, that contains all the relevant data, contains several test cases

Each test case is described in the following way. The first line contains the number n of aircraft ($2 \leq n \leq 2000$). This line is followed by n lines. Each of these lines contains two integers, which represent the early landing time and the late landing time of an aircraft. Note that all times t are such that $0 \leq t \leq 10^7$.

Output

For each input case, your program has to write a line that conttains the maximal security gap between consecutive landings.

Note: The input file corresponds to Table 1.

Robert's Hints

Optimization vs. Decision Robert advises you to work on the decision variant of the problem. It can then be stated as follows: Given an integer p, and an instance of the optimization problem, the question is to decide if there is a solution with security gap p or not. Note that, if you know how to solve the decision variant of an optimization problem, you can build a binary search algorithm to find the optimal solution.

On decision Robert believes that the decision variant of the problem can be *modeled as a very particular boolean satisfaction problem*. Robert suggests to associate a boolean variable per aircraft stating whether the aircraft is early (variable takes value "true") or late (value "false"). It should then be easy to see that for some aircraft to land at some time has consequences for the landing times of other aircraft. For instance in Table 1 and with a delay of 10, if aircraft A_1 lands early, then aircraft A_3 has to land late. And of course, if aircraft A_3 lands early, then aircraft A_1 has to land late. That is, aircraft A_1 and A_3 cannot both land early and formula $(A_1 \Rightarrow \neg A_3) \wedge (A_3 \Rightarrow \neg A_1)$ must hold.

And now comes Robert's big insight: our problem has a solution, if and only if we have no contradiction. A contradiction being something like $A_i \Leftrightarrow \neg A_i$.

Sample Input

```
10
44 156
153 182
48 109
160 201
55 186
54 207
55 165
17 58
132 160
87 197
```

Sample Output

```
10
```