

Problem G. Scheduling

Input file: *standard input*
 Output file: *standard output*
 Time limit: 3 seconds
 Memory limit: 512 mebibytes

Your boss made you find a valid schedule for his meetings. He has n meetings, where the i -th meeting has to start and end in the time interval $[\ell_i, r_i]$, where time is measured in hours. Each meeting takes exactly one hour, and your boss obviously cannot attend two meetings at the same time.

As a competitive programmer, you have already solved this classical scheduling problem countless times, and this time, you finally notice that it is completely unrealistic. How can one person attend over a hundred thousand meetings at different places without any time to travel between them and without taking any breaks either? There is no way every meeting takes exactly one hour – speaking from experience – and there will surely be meetings that start late because someone is running late.

These deep-rooted philosophical questions may be a big problem for humanity, but since you do not have to attend these meetings yourself, you simply do not care enough to solve all of them. However, impressing your boss will certainly be worth it for you, so you try to at least improve the schedule a little bit.

Find any schedule where there is at least one hour between every pair of meetings or report that there is none. Surely this is just one insignificant change to the classical problem, right?

Input

Each test contains multiple test cases. The first line of input contains a single integer t ($1 \leq t \leq 10^5$) – the number of test cases. The description of the test cases follows.

The first line of each test case contains one integer n ($1 \leq n \leq 2 \cdot 10^5$) – the number of meetings.

The next n lines each contain two integers ℓ_i and r_i ($1 \leq \ell_i < r_i \leq 10^6$) – the time interval in hours the i -th meeting has to start and end in.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

If there is no valid schedule, output -1 in a single line. Otherwise, output n integers a_i in a single line where a_i is the start time of the i -th meeting.

Example

| <i>standard input</i> | <i>standard output</i> |
|-----------------------|------------------------|
| 4 | 1 5 3 |
| 3 | -1 |
| 1 3 | -1 |
| 1 7 | 4 2 |
| 2 4 | |
| 2 | |
| 1 2 | |
| 2 3 | |
| 4 | |
| 1 5 | |
| 2 6 | |
| 3 4 | |
| 4 7 | |
| 2 | |
| 1 5 | |
| 2 3 | |

Note

In the first test case, the first meeting starts at the start of hour 1 and goes on for one hour until the start of hour 2. After that, your boss takes a break of one hour until he attends the third meeting from the start of hour 3 to 4. Then, he takes another one hour break until he finally attends the second meeting, which happens from the start of hour 5 to 6.

Note that it is **not** possible to start the third meeting at the start of hour 4 as this meeting would end at 5.