General presentation guidelines:

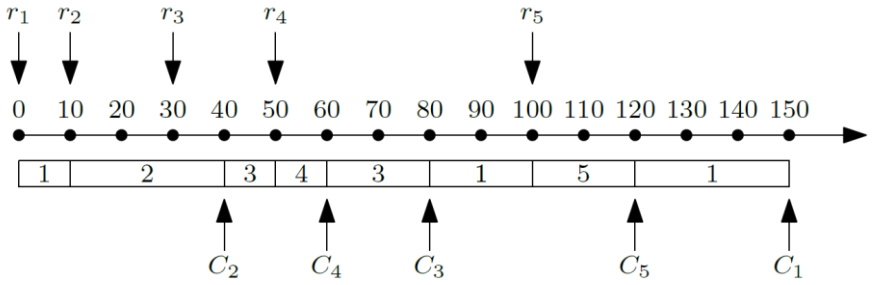
1. Explain the problem description clearly with proper language and notations.
2. Design the algorithm with pseudo code and analyze the running time with accurate asymptotic notations.
3. Implement your algorithm using any programming language (Java, C++, Python, etc.) and illustrate the execution with an example input.
4. Document and upload your implementation codes to a public GitHub account.
5. Upload your presentation slides to Moodle.

The presentation is 12 minutes plus 2-3 minutes Q/A. Asking questions will get extra credit.

Problem: Given a set of n jobs indexed as 1, 2, $\cdots$, n. Each job j has a release time r_j indicating when it becomes available for processing, and a processing time p_j indicating how long it takes to complete. The machine can process jobs preemptively, meaning that it can interrupt the processing of a job to process another job and then resume the interrupted job later. There is no overhead in switching the processing of jobs. A job j is completed when it is processed for p_j units of time, and its completion time C_j is the time when it is completed. Our goal is to decide a schedule for all the jobs so as to minimize the total completion time of all jobs, i.e., $\sum_{j=1}^{n} C_j$. For example, consider 5 jobs with release times and processing times described in the following table:

j	1	2	3	4	5
r_j	0	10	30	50	100
p_j	60	30	30	10	20

Then, the optimum schedule for the instance is illustrated in the following figure:



The total completion time of the 5 jobs is 40 + 60 + 80 + 120 + 150 = 450.

Input:

- The input is taken from the standard input (console).
- The first line of input contains one integer n, the number of jobs.
- The next n lines give the description of the n jobs. Each line contains two integers r and p, denoting the release time and processing time of a job.

Output:

- The output is printed to the standard output (console). It contains a single line, which is the total completion time of the optimum schedule.

Example Input:	**Example Output:**
5 0 60 10 30 30 30 50 10 100 20	450

Constraints:

- $1 \leq n \leq 10^6$.
- The release times are integers in [0, 10^6] and the processing times are integers in [1, 10^6].
- It is expected that your program will terminate in 10 seconds.