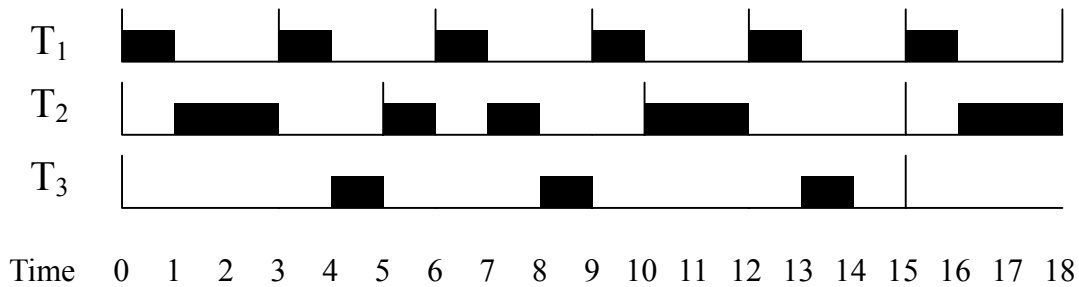


Problem D
A Preemptive EDF Scheduler
Input file: pd.txt

Problem Statement

Priority-driven scheduling approach is commonly used in modern computer operating systems to execute a set of tasks where the systems always execute the task with the highest priority. Earliest Deadline First (EDF) scheduling approach assigns a higher priority for a task with earlier deadline where the tasks are periodic and preemptive. We are interested in the earliest time when all tasks have finished at least once. Note that it would be easier and faster to simulate the problem using two priority queues, one for the ready and preempted tasks, and the other for the rest.



For example, in the above figure, tasks T_1 , T_2 , T_3 with execution time 1, 2, 3 and periods 3, 5, 15 respectively are feasible using the preemptive EDF scheduler. T_1 , T_2 , and T_3 first finish execution at time 1, 3, and 14 respectively. The whole schedule repeats at 15. T_2 is preempted by T_1 at time 6 and resume at time 7, T_3 is preempted by T_2 at time 5 and resume at time 8, preempted again by task T_1 at time 9 and resume at time 13.

Definitions and Constraints

1. A task T_2 is preempted by a task T_1 means T_1 , with a higher priority, replaces T_2 , with a lower priority, to run the CPU, and T_2 will resume later.
2. A ready task is a task ready to run as long as it gets the right to use CPU.
3. A periodic task is executed exactly once in every constant interval, called period.
4. For simplicity, a periodic task is ready at the beginning of each period and its deadline is at the end of each period.
5. A set of periodic tasks is feasible if every task finishes execution before its deadline.
6. All the input numbers are positive integers and are less than or equal to 500000.
7. The number of tasks in each task set is less than or equal to 10.
8. Each execution time is less than or equal to its period.

9. The periods are not sorted and are all different in a task set.
10. All the test task sets are feasible.

Input File Format

The first line of input file consists of a single number denoting the number of test cases in the file. There is a single line containing a '/' character separating two consecutive test cases. The end of the file is marked with a '.' character. For each test case, the first line gives the number of tasks, n , followed by n lines of blank space separated task execution time and period pairs.

Output Format

For each test cases, print on a single line the earliest time when all tasks will have finished at least once.

Sample Input

```
3
3
1 3
2 5
3 15
/
2
1 2
1 3
/
3
1 3
2 15
2 5
.
```

Sample Output

```
14
2
9
```