

問題：速率单调（RM）调度算法和最早截止时间优先（EDF）调度算法分别是固定优先级分配和动态优先级分配的实时调度中的最优算法。但是，我们可能偶尔需要使用其中一种算法，而不是混合使用，这样程序才能方便地执行。您可能需要考虑调度程序必须永远运行，并且尽可能高效。

計畫：您需要为这两种算法设计一个高效的程序，根据以下输入格式（从 `stdin` 输入）和输出格式（从 `stdout` 输出），为任务集生成完整的时间表（最多超周期，周期的最小公倍数（LCM），但最好不使用 LCM 的概念）。所有输入和输出均为非负整数，中间用空格隔开，不带注释，但实际当前时间和间隔时间为浮动数，单位为秒，精度可达纳秒。请注意，如果两个任务的优先级相同，则任务 ID 较小的任务优先级较高。

輸入：

1. R- 运行次数 > 0
2. S- 调度策略，0:RM（最短周期获得更高的优先级），1:EDF（最早截止日期获得更高的优先级）
3. D- 显示计划，1，或不显示，0
4. N- 0 < 任务数 ≤ 128，任务编号从 1 开始

T1 P1 - 第一个任务的执行时间、周期，周期不排序

T2 P2 - 第 2 个任务的执行时间、周期

...

Tn Pn - 第 n 个任务的执行时间、周期

輸出：

1. RUN - 运行编号
2. TIME [TASK ACTION] ... - 事件顺序很重要，时间不会重复，空闲时间单位可能会消失。
 - TIME - 以整数表示的模拟时间
 - TASK - 任务 ID
 - ACTION – start:0; end:1; preempted:2; resume:3
3. 结束抢占
 - END - 结束模拟时间，如果任务集不可调度，则为 0
 - PREEMPTED times- 抢占次數

輸入範例

```
2 // 2 runs
1 // the 1st run, EDF
1 // display schedule
2 // 2 tasks
1 2 // (execution time 1) (period 2), for the 1st task
2 6 // (execution time 2) (period 6), for the 2nd task
0 // the 2nd run, RMS
0 // display schedule
2 // 2 tasks
1 2 // execution time 1, period 2, for the 1st task
1 3 // execution time 1, period 3, for the 2nd task
```

輸出範例

```
1 // start the 1st run
0 1 0 // at time 0, task 1 starts;
1 1 1 2 0 // at time 1, task 1 ends, then task 2 starts;
2 2 2 1 0 // at time 2, task 2 is preempted, then task 1 starts;
3 1 1 2 3 // at time 3, task 1 ends, then task 2 resumes;
4 2 1 1 0 // at time 4, task 2 ends, then task 1 starts;
5 1 1 // at time 5, task 1 ends;
6 1 // at time 6, end of the 1st run; preempted 1 time
2 // start the 2nd run
6 0 // at time 6, end of the 2nd schedule; preempted 0 time
```

If the case is not schedulable, the output of the program might look like this:

```
1 // start the 1st run
0 0
```