

2017 年 C++ 第一次机考

奥恩是一位不知疲倦的锻造师，可以无间断工作，同一个时刻他只能锻造一个兵器，假设锻造的兵器需要一段时间能锻造完（可以预先知道），锻造的过程可以间断，间断之后可以继续锻造。现在不断的有兵器送到锻造师处，如果没有兵器锻造师什么都不做。

1. 当有多把兵器送到锻造师处，锻造师优先挑选锻造所需时间最短的兵器进行锻造，锻造完该兵器之后继续锻造其他兵器。
2. 如果锻造师正在锻造某个兵器的时候，有新的兵器送进来，且所需锻造的时间比正在锻造的兵器所需剩余的时间短，则将正在锻造的兵器先搁置，先锻造新的兵器。锻造完新进来的兵器之后接着去锻造所需要剩余锻造时间最短的兵器。
3. 我们额外规定每一个兵器都有一个锻造的优先级，在所需要的剩余时间相同的情况下，优先锻造优先级比较高的。
4. 如果兵器的所需剩余时间和优先级都相同，则锻造师优先锻造先送到锻造师处的兵器。不会出现同时到达且优先级相同、剩余时间相同的情况

现在规定送到锻造师处的所有兵器都有一个兵器号，兵器名称，送到锻造师处的时间，锻造所需时间和 优先级，锻造师根据上面的 4 条规则对兵器依次进行锻造。

规定：

1. 优先级的数字越大，优先级越高
2. 锻造师切换锻造两个兵器的时间忽略不计

输入：

1. 输入正整数 n ，表示将会有 n 把兵器（ $n \leq 1000$ ，每个兵器需要锻造的时间不会超过 `int` 类型的最大值）
2. 依次输入 n 把兵器的信息，每个兵器一行，一行中的每条信息使用空格隔开，分别是：

兵器号 兵器名称 送到锻造师处的时间 锻造所需时间 优先级

3. 输出：

按先执锻造完兵器的顺序依次输出每个兵器的信息，包含 兵器号，兵器名称，送到锻造师处的时间，锻造所需时间，优先级，兵器从送达锻造师处到锻造完成所花时间，兵器在哪些时间段接受锻造（时间段使用“开始时间-结束时间”的格式，开始时间和结束时间都是从

0 开始计数的单位时间数，有一个时间段则写一个，多个则写多个追加在同一行的末尾，**空格隔开**）

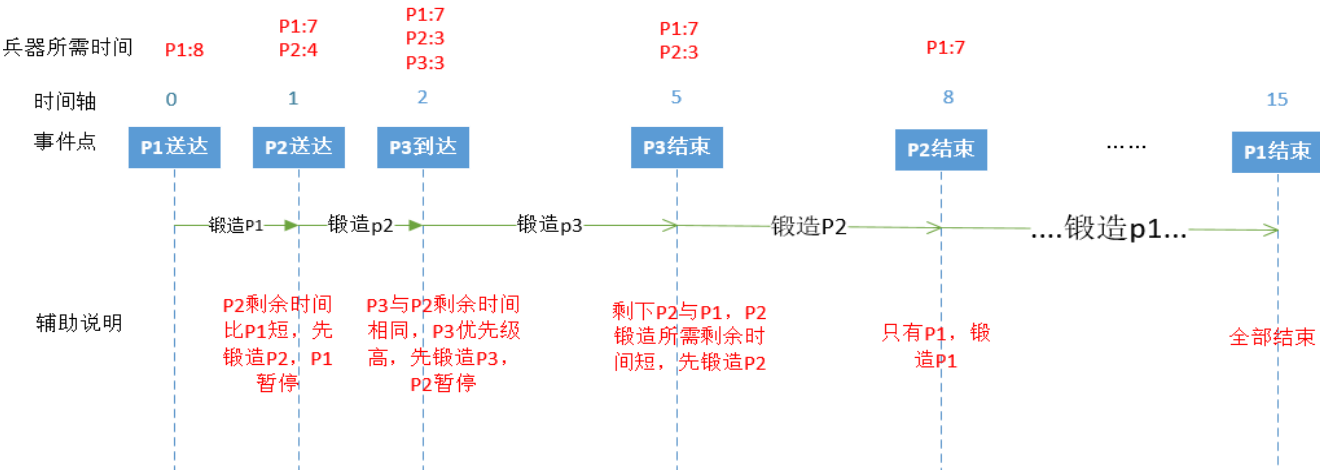
时间为从 0 开始计时的单位时间，单位为小时，1 表示 1 小时，不断往后累加

例子：

有以下兵器和信息作为输入：

兵器号	兵器名称	送达锻造师处时间	所需锻造时间	优先级
1	P1	0	8	1
2	P2	1	4	1
3	P3	2	3	2

他们锻造的顺序如下图所示：



说明：

- 0 时，P1 开始锻造，P1 锻造了 1 个时间单位
- 1 时，P2 送达，P1 搁置，锻造 P2，P2 锻造 1 个时间单位
- 2 时，P3 送达，P3 开始锻造，P2 搁置，P3 锻造 3 个时间单位
- 5 时，P3 锻造完，P3 共耗时 3，P2 被锻造 3 个时间单位
- 8 时，P2 锻造完，P2 共耗时 7，P1 接受 7 个时间单位的锻造
- 15 时，全部结束，P3 共耗时 15

程序中对应的输入如下：

3

1 P1 0 8 1

2 P2 1 4 1

3 P3 2 3 2

程序中对应输出：

3 P3 2 3 2 3 2-5

2 P2 1 4 1 7 1-2 5-8

1 P1 0 8 1 15 0-1 8-15（无换行）

输出对应的含义：

兵器号 兵器名称 送到锻造师处的时间 锻造所需时间 优先级 送达到锻造
结束时间 被锻造时的时间段序列

对于锻造时间端的补充说明以 P1 为例，P1 在时间轴上的被锻造的时间有 0-1,8-15，所以在程序中输出的时间段是 0-1 8-15，而 P3 只有 2-5 的时间段，因此只有一个

限制：

可以允许使用的头文件：stdio.h, stdlib.h, iostream, string.h, string, memory.h, memory, math.h, algorithm, limits.h

请勿使用其他头文件!!!!

额外说明：

插件中测试反馈的结果说明： AC 是通过，WA 是错误，TIE 是超时