

## 问题描述

2023 年 5 月, 麦当劳在北邮开业。大量的学生去那里订餐。正因为如此, 麦当劳的在线点餐系统经常关闭以避免拥挤, 尤其是在午餐和晚餐时间。该系统的关闭时间不确定。北邮的学生认为这非常麻烦。

然而, 北邮学生无所畏惧。北京邮电大学最优秀的学生之一(也是北邮 ICPC 团队的一员)Zhai Xie (ThomasX)在飞书上开发了一个实时监控机器人, 它告诉我们麦当劳在线点餐系统的实时状态。有了这个机器人, 北邮学生可以更方便地点餐。

在这个问题中, 需要你像他一样完成这个任务, 开发一个系统来模拟麦当劳的在线点餐系统。

北邮的麦当劳和它的点餐系统在07:00:00开始工作, 在22:00:01关闭。麦当劳一共有 $N$ 种食物和 $M$ 种套餐类型, 每种套餐中包含多种食物, 具体配置信息将在菜单文件 (dict.dic) 中提供。对于制作和存储每种食物, 规定第 $i$ 种食物在 $t_i$ 秒内完成, 其最大存储容量为 $cap_i$ , 表示该种食物最多可以存储 $cap_i$ 个。麦当劳系统每天开放前, 所有食物存储容量都为 0, 在任何时间点如果某种食物的存储量小于 $cap_i$ , 则会立即制作该食物, 直到达到 $cap_i$ 。其中, 不同种类食物可以同时制作, 同种类食物只能依次制作。

从07:00:00到22:00:00(含), 学生可以在系统中点餐(如果系统未关闭)。每一天按照顺序有 $n$ 个订单, 第 $i$ 个订单发生在时间 $a_i:b_i:c_i$ , 其要求一份 $type_i$ 类型( $type_i \in M_{combo} \cup N_{food}$ , 其中 $M_{combo}$ 和 $N_{food}$ 分别表示全体的套餐和食物的集合)的套餐或食物。如果点餐时系统关闭, 会导致点餐失败。22:00 以后如果还有之前的订单未完成, 则麦当劳会继续加班, 且保证 23:59:59(含)前一定能完成所有订单。

对于订单处理存在如下规则:

- 在每一秒的开始, 如果有新的食物完成, 则首先存储食物, 然后接受订单(如果存在)。
- 订单按照“先来先到, 异步处理”原则进行处理。
  - 先来先到: 指的是对于有存量的食物, 总会被分配给时间最早的订单(套餐或单点)。
  - 异步处理: 指的是当一个订单(套餐或单点)因为请求的食物没有被全部满足时, 不必等待该订单完成, 可以直接处理下一个订单。
- 食物一旦被分配给订单, 就不能撤销。食物被分配给订单后, 即便该订单尚未完成, 该食物也不再占用对应类型的容量。
- 当订单(套餐或单点)中要求的所有食物, 均已被分配给该订单, 则该订单会立刻完成。

- 如果在某个时刻 $t_0$ ，有人下了一个订单，并且该订单无法立刻完成，导致未完成订单的数量大于 $W_1$ ，则系统立即自动关闭(不再接受订单)，但该订单仍然算作成功下单。
- 如果在某个时刻 $t_1$ ，未完成订单的数量小于 $W_2$ ，则系统将在 1 秒后重新打开。即系统可以接受 $t_1 + 1$ 时刻的订单，而不能接受 $t_1$ 时刻的订单。

你的系统需要输出：每一个订单是否下单成功，以及完成的时间。

## 菜单文件

本题为大家提供麦当劳的菜单文件(dict.dic)，按如下格式给出：

第一行给出 $N$ 和 $M$ ，其中 $N$ 表示食物的种类数， $M$ 表示套餐的种类数。

第二行包含 $N$ 个字符串，每个字符串  $name_i^{food}$  表示第 $i$ 种食物的名称。

接下来 $M$ 行，其中的第 $i$ 行包含多个字符串，第一个字符串 $name_i^{combo}$ 表示第 $i$ 个套餐的名称，后续的第 $j$ 个字符串 $name_{i,j}^{food}$ 表示第 $i$ 个套餐中包含的第 $j$ 种食物的名称。

注：系统每次运行时所读取的菜单文件内容可能不一样。

## 输入

第一行包含一个整数 $n(1 \leq n \leq 54001)$ 表示订单个数。

第二行包含两个整数 $W_1, W_2(2 \leq W_2 \leq W_1 \leq 100)$ 。

第三行包含 $N$ 个整数 $t_1, t_2, \dots, t_N(1 \leq t_i \leq 70)$ ，其中 $t_i$ 表示第 $i$ 种食物的制作时长。

第四行包含 $N$ 个整数 $cap_1, cap_2, \dots, cap_N(1 \leq cap_i \leq n)$ ，其中 $cap_i$ 表示第 $i$ 种食物的最大存储容量。

对于接下来的 $n$ 行，用格式类似于11:11:11的方式，给出第 $i$ 个订单的时间。然后输入一个字符串 $type_i$ ，表示套餐或食物的名称(参见 dict.dic)。所有订单时间一定在[07:00:00,22:00:00]内，同一个时间点不可能出现多个订单，第 $i - 1$ 个订单一定早于第 $i$ 个( $2 \leq i \leq n$ )，且保证 23:59:59(含)前一定能完成所有订单。

具体参见 input.txt

## 输出

输出包括 $n$ 行，按照订单顺序输出订单完成时间。对于第 $i$ 行，如果第 $i$ 个订单不成功，则输出 Fail；否则，输出这个订单完成的时间，时间格式与输入格式(11:11:11)一致。

具体参见 output.txt