

栗仁武！欢迎您登录计算机软件能力认证考试系统！

2017年12月3日 星期日

准考证号：201708803262

考试剩余时间：20分04秒

姓名：栗仁武

当前时间：17:05:41

[返回试题列表](#)

问题描述

试题
编号：

3

试题
名称：

Crontab

时间
限制：

1.666min

内存
限制：

512.0MB

问题
描述：

Crontab

问题描述

Cron（源自希腊语 χρόνος，意思是时间）是类 Unix 系统下基于时间的任务调度器，用于在固定的时间运行指定的任务（多为系统管理和维护的任务）。Cron 适合于周期性重复调度的任务，通过 crontab 文件来描述调度任务的配置。

Crontab 文件由若干行组成，每行是一条配置信息，格式如下：

```
1 | <minutes> <hours> <day of month> <month> <day of week> <command>
```

表示在满足前 5 项所指定的时间来运行第 6 项 `<command>` 所描述的命令。前 5 项用于描述时间，含义和取值范围如下：

- `<minutes>` 是分钟数，取值范围是 0-59；
- `<hours>` 是小时数，取值范围是 0-23；
- `<day of month>` 是月份中的天数，取值范围是 1-31；
- `<month>` 是月份，取值范围是 1-12，或 Jan-Dec；
- `<day of week>` 是星期几，取值范围是 0-6，或 Sun-Sat。

对于前 5 项，除了可以直接给出数字或者英文缩写（不区分大小写）外，还可以出现星号 `*`（表示任何取值）、逗号 `,`（表示多个不同的取值）或减号 `-`（表示一段连续的取值范围）。星号只能单独出现，减号和逗号可以配合出现。

Cron 每分钟检查一次系统时间，当系统时间同时满足这 5 项要求时，cron 将执行对应的命令。给出一个时间段，以及一个 crontab 文件，请你编程输出在这段时间内的任务调度执行情况。

输入格式

输入第一行有 3 个整数 n 、 s 、 t ，用空格分隔。 n 表示接下来有 n 行，描述一个 crontab 文件。 s 和 t 分别为系统运行的开始时间（包含）和结束时间（不包含），格式为 `yyyymmddHHMM`（年、月、日、小时、分钟）。

接下来有 n 行，每行是一条 crontab 配置信息，相邻两项之间用一个空格分隔。

输出格式

输出有若干行，每行表示一个任务调度，由两部分构成：第一部分是任务调度的时间，格式同样为 `yyyymmddHHMM`；第二部分是调度执行的命令。两部分之间用一个空格分隔。按照时间先后顺序输出。如果同一时刻有多条命令满足调度条件，则按照输入给出的顺序输出。

样例输入

```
1 3 201711170032 201711222352
2 0 7 * * 1,3-5 get_up
3 30 23 * * Sat,Sun go_to_bed
4 15 12,18 * * * have_dinner
```

样例输出

```
1 201711170700 get_up
2 201711171215 have_dinner
3 201711171815 have_dinner
4 201711181215 have_dinner
5 201711181815 have_dinner
6 201711182330 go_to_bed
7 201711191215 have_dinner
8 201711191815 have_dinner
9 201711192330 go_to_bed
10 201711200700 get_up
11 201711201215 have_dinner
12 201711201815 have_dinner
13 201711211215 have_dinner
14 201711211815 have_dinner
15 201711220700 get_up
16 201711221215 have_dinner
17 201711221815 have_dinner
```

样例说明

样例输入给出了3条 cron 配置信息，系统运行的开始时间是 2017-11-17 00:32（包含），结束时间为 2017-11-22 23:52（不包含）。每条配置信息的含义如下：

- 1. 在星期一、三、四、五的7点整运行 `get_up` 命令。
- 2. 在星期六、星期天的23点30分运行 `go_to_bed` 命令。
- 3. 在每天的12点15分和18点15分运行 `have_dinner` 命令。

数据规模和约定

输入数据约定：

- $1 \leq n \leq 20, s \leq t$
- 输入数据中给出的时间均在 1970-01-01 00:00 到 2099-12-31 23:59 之间。
- 输入中给出的命令只包含大小写字母、数字和下划线 `_`，不包含空格或其他符号。
- 保证输入中描述时间的部分都是合法的。对于减号描述的时间范围 $x-y$ ，一定满足 $x \leq y$ 。英文缩写的拼写保证是正确的。英文缩写和数值可以混合使用。分钟数和小时数可以有前导0，也可以没有（例如，00 和 0 都是合法的输入），其他部分不会出现前导 0。
- 输入的每行不超过 100 个字符。
- 保证输出内容不超过 10000 行。

- 提示：1970年1月1日是星期四。

输入数据特征（“✓”表示可以出现，“✗”表示不会出现）：

测试用例编号	星号	逗号	减号	英文缩写
1-2	✗	✗	✗	✗
3-6	✓	✗	✗	✗
7-8	✓	✓	✗	✗
9-10	✓	✗	✓	✗
11-12	✓	✓	✓	✗
13-14	✓	✗	✗	✓
15-20	✓	✓	✓	✓

附：月份与星期的英文缩写对照表

月份	数值	英文缩写
一月	1	Jan
二月	2	Feb
三月	3	Mar
四月	4	Apr
五月	5	May
六月	6	Jun
七月	7	Jul
八月	8	Aug
九月	9	Sep
十月	10	Oct
十一月	11	Nov
十二月	12	Dec

星期几	数值	英文缩写
星期天	0	Sun
星期一	1	Mon
星期二	2	Tue
星期三	3	Wed
星期四	4	Thu
星期五	5	Fri
星期六	6	Sat

样例输入

```
3 201711170032 201711222352
0 7 * * 1,3-5 get_up
30 23 * * Sat,Sun go_to_bed
15 12,18 * * * have_dinner
```

样例输出

```
201711170700 get_up
201711171215 have_dinner
201711171815 have_dinner
201711181215 have_dinner
201711181815 have_dinner
201711182330 go_to_bed
```

```
201711191215 have_dinner
201711191815 have_dinner
201711192330 go_to_bed
201711200700 get_up
201711201215 have_dinner
201711201815 have_dinner
201711211215 have_dinner
201711211815 have_dinner
201711220700 get_up
201711221215 have_dinner
201711221815 have_dinner
```

答题栏

试题编号: 3

试题名称: Crontab

编译环境: 1. Java ▼

答案程序:

提交确认:

- 以下必须全部满足才能提交:
- ☐ 我的程序没有使用package语句来定义包的信息。（如果定义了将无法评测）
 - ☐ 我的程序的主类名称是Main，主类的定义是public class Main。我的程序运行的入口是主类Main中的main函数，定义为public static void main(String[] args)。
 - ☐ 我的程序是从标准输入(System.in)中读入数据的（使用Scanner等类来处理System.in的输入视为满足条件），结果是输出到标准输出(System.out)的。
 - ☐ 我的程序是从标准输入中读入数据的，结果是输出到标准输出的。
 - ☐ 我的程序中没有“请输入n”之类的输入输出提示，也没有输出中间的结果，所有的输出内容都与题设中的输出格式相对应。我的输出格式（包括换行和大小写等）与题设中输出格式的要求相符。

提交

查看上一次提交

返回试题列表