

题目描述: 智能手机方便了我们生活的同时,也侵占了我们不少的时间。“手机App防沉迷系统”能够让我们每天合理的规划手机App使用时间,在正确的时间做正确的事。

它的大概原理是这样的:

1、在一天24小时内,可注册每个App的允许使用时段:



2、一个时段只能使用一个App,举例说明:不能同时在09:00-10:00注册App2和App3;



3、App有优先级,数值越高,优先级越高。注册使用时段时,如果高优先级的App时间和低优先级的时段有冲突,则系统会自动注销低优先级的时段;如果App的优先级相同,则后添加的App不能注册。

举例1:

(1) 注册App3前:



(2) App3注册时段和App2有冲突:



(3) App3优先级高,系统接受App3的注册,自动注销App2的注册:



举例2:

(1) 注册App4:



(2) App4和App2及App3都有冲突,优先级比App2高,但比App3低,这种场景下App4注册不上,最终的注册效果如下:



4、一个App可以在一天内注册多个时段。



请编程实现,根据输入数据注册App,并根据输入的时间点,返回该时间点可用的App名称,如果该时间点没有注册任何App,请返回字符串"NA".

输入描述: 输入分3部分: 第一行表示注册的App数N (N≤100); 第二部分包括N行,每行表示一条App注册数据;最后一行输入一个时间点,程序即返回该时间点的可用App。

2

App1 1 09:00 10:00

App2 2 11:00 11:30

09:30

数据说明如下:

1、N行注册数据以空格分隔,四项数据依次表示: App名称、优先级、起始时间、结束时间

2、优先级1-5, 数字值越大, 优先级越高

3、时间格式HH:MM, 小时和分钟都是两位, 不足两位前面补0

4、起始时间需小于结束时间, 否则注册不上

5、注册信息中的时间段包含起始时间点, 不包含结束时间点

输出描述: 输出一个字符串, 表示App名称, 或NA表示空闲时间。

补充说明: 1、用例保证时间都介于00:00-24:00之间;

2、用例保证数据格式都是正确的, 不用考虑数据输入行数不够、注册信息不完整、字符串非法、优先级超限、时间格式不正确的问题。

示例 1

输入:

```
1
App1 1 09:00 10:00
09:30
```

输出:

```
App1
```

说明:

*App1* 注册在 9 点到 10 点间, 9 点半可用的应用名是 *App1*

示例 2

输入:

```
2
App1 1 09:00 10:00
App2 2 09:10 09:30
09:20
```

输出:

```
App2
```

说明:

*App1* 和 *App2* 的时段有冲突, *App2* 的优先级比 *App1* 高, 注册 *App2* 后, 系统将 *App1* 的注册信息自动注销后, 09:20 时刻可用应用名是 *App2*.

示例 3

输入:

```
2
App1 1 09:00 10:00
```

App2 2 09:10 09:30

09:50

输出:

NA

说明:

*App1* 被注销后, *09:50* 时刻没有应用注册, 因此输出 *NA*。

```
def get_time(time):
```

```
    h, m = time.split(':')
```

```
    if len(m) == 1:
```

```
        m = '0' + m
```

```
    return float(h+'.'+m)
```

```
try:
```

```
    n = int(input())
```

```
    app_list = []
```

```
    for i in range(n):
```

```
        app, priority, start_time, end_time = input().split()
```

```
        start_time = get_time(start_time)
```

```
        end_time = get_time(end_time)
```

```
        if end_time <= start_time:
```

```
            break
```

```
    found_list = []
```

```
    loc = 0
```

```
    for index, item in enumerate(app_list):
```

```
        a, p, s, e = item
```

```
        if e < start_time:
```

```
            loc = index + 1
```

```
        if s < start_time < e or s < end_time < e:
```

```
            found_list.append(index)
```

```
    success = True
```

```
    for index in found_list:
```

```
        if priority <= app_list[index][1]:
```

```
            success = False
```

```
            break
```

```
    if success:
```

```
        post = loc
```

```
        if len(found_list) > 0:
            post = found_list[-1] + 1
            app_list = app_list[:loc] + [[app, priority, start_time, end_time]] + app_list[post:]

this_time = get_time(input())
found = False
for item in app_list:
    a, p, s, e = item
    if s <= this_time <= e:
        found = True
        print(a)
        break
if not found:
    print('NA')
except Exception:
    print('NA')
```