

## Coding problem: Conference Track Management

You are planning a big programming conference and have received many proposals which have passed the initial screen process but you're having trouble fitting them into the time constraints of the day -- there are so many possibilities! So you write a program to do it for you.

- The conference has multiple *tracks* each of which has a morning and afternoon *session*.
- Each session contains multiple *talks*.
- Morning sessions begin at 9am and must finish by 12 noon, for lunch.
- Afternoon sessions begin at 1pm and must finish in time for the *networking event*.
- The networking event can start no earlier than 4:00 and no later than 5:00.
- No talk title has numbers in it.
- All talk lengths are either in minutes (not hours) or *lightning* (5 minutes).
- Presenters will be very punctual; there needs to be no gap between sessions.

Note that depending on how you choose to complete this problem, your solution may give a different ordering or combination of talks into tracks. This is acceptable; you don't need to exactly duplicate the sample output given here.

### Sample input:

```
Writing Fast Tests Against Enterprise Rails 60min
Overdoing it in Python 45min
Lua for the Masses 30min
Ruby Errors from Mismatched Gem Versions 45min
Common Ruby Errors 45min
Rails for Python Developers lightning
Communicating Over Distance 60min
Accounting-Driven Development 45min
Woah 30min
Sit Down and Write 30min
Pair Programming vs Noise 45min
Rails Magic 60min
Ruby on Rails: Why We Should Move On 60min
Clojure Ate Scala (on my project) 45min
Programming in the Boondocks of Seattle 30min
Ruby vs. Clojure for Back-End Development 30min
Ruby on Rails Legacy App Maintenance 60min
A World Without HackerNews 30min
User Interface CSS in Rails Apps 30min
```

### Sample output:

```
Track 1:
09:00AM Writing Fast Tests Against Enterprise Rails 60min
10:00AM Overdoing it in Python 45min
10:45AM Lua for the Masses 30min
11:15AM Ruby Errors from Mismatched Gem Versions 45min
12:00PM Lunch
01:00PM Ruby on Rails: Why We Should Move On 60min
```

02:00PM Common Ruby Errors 45min  
02:45PM Pair Programming vs Noise 45min  
03:30PM Programming in the Boondocks of Seattle 30min  
04:00PM Ruby vs. Clojure for Back-End Development 30min  
04:30PM User Interface CSS in Rails Apps 30min  
05:00PM Networking Event

Track 2:

09:00AM Communicating Over Distance 60min  
10:00AM Rails Magic 60min  
11:00AM Woah 30min  
11:30AM Sit Down and Write 30min  
12:00PM Lunch  
01:00PM Accounting-Driven Development 45min  
01:45PM Clojure Ate Scala (on my project) 45min  
02:30PM A World Without HackerNews 30min  
03:00PM Ruby on Rails Legacy App Maintenance 60min  
04:00PM Rails for Python Developers lightning  
05:00PM Networking Event

**In-office Extensions:**

For an in-office pairing exercise, choose one or more of the following extensions and see how it can be worked into the candidate's code.

- The lunch break may be delayed by up to half an hour if it allows more talks to fit in the morning sessions. The lunch break must still be an hour, and must start at the same time for all tracks.
- Talks now require a break between them. A block of lightning talks can be placed together with no breaks, but require a ten minute break before and after the block. All other talks require a ten minute break between them and another talk. *Talks won't all fit into two tracks if we add breaks, so the solution will require 3 tracks or multiple days of sessions.*