**Conference Track Management**

**Problem Statement**

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.

**GitHub Link** - <https://github.com/AkshataSH/ConfTrack.git/>

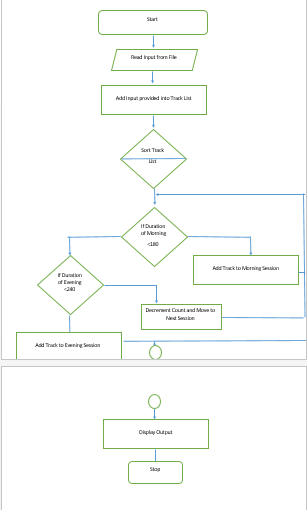
**Approach:**

**In order to address this problem, have chosen bin packing algorithm. In bin packing algorithm, objects of different volume to be packed into finite number of bins. Same way here topics of different duration to be packed into different tracks with sessions.**

**With bin packing algorithm, best fit algorithm is used in order to use time efficiently.**

**In Best Fit Algorithm, given input is sorted in descending order and given input is processed in an arbitrary order. For Each topic, it attempts to place the topic, in session that can accommodate the topic.**

**Flow Chart:**



**Assumptions:**

* **There are only two Tracks.**
* **In the provided Input, duration with min is specified without a space. For Ex : 30min- Valid, 30 min - Invalid**

**Input/output test cases:**

**Input 1:**

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

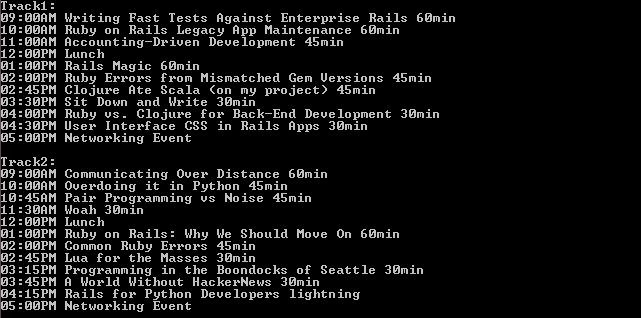
Ruby vs. Clojure for Back-End Development 30min

Ruby on Rails Legacy App Maintenance 60min

A World Without HackerNews 30min

User Interface CSS in Rails Apps 30min

**Output 1**:



**Input 2:**

Writing Fast Tests Against Enterprise Rails 60hour

Overdoing it in 6789 Python 45min

Lua for the Masses 30 seconds

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

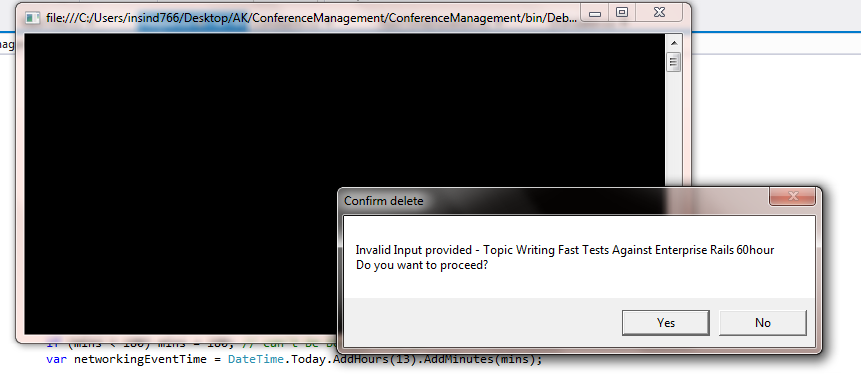
Ruby on Rails Legacy App Maintenance 60min

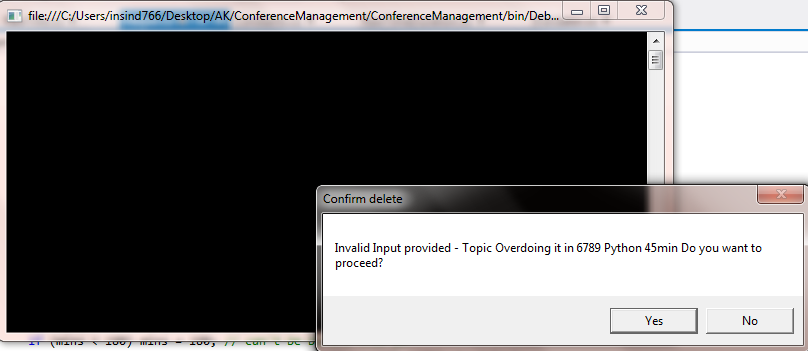
A World Without HackerNews 30min

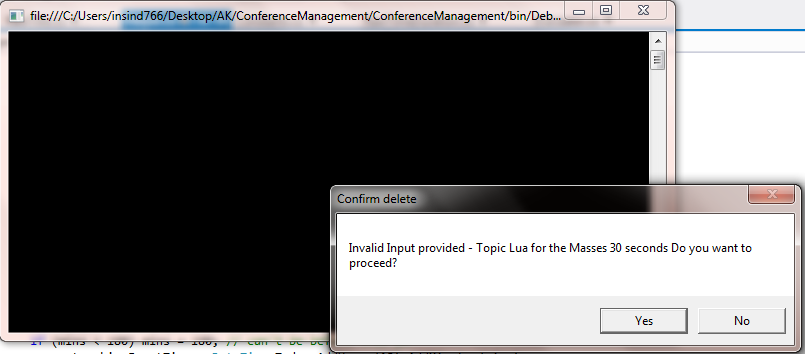
User Interface CSS in Rails Apps 30min

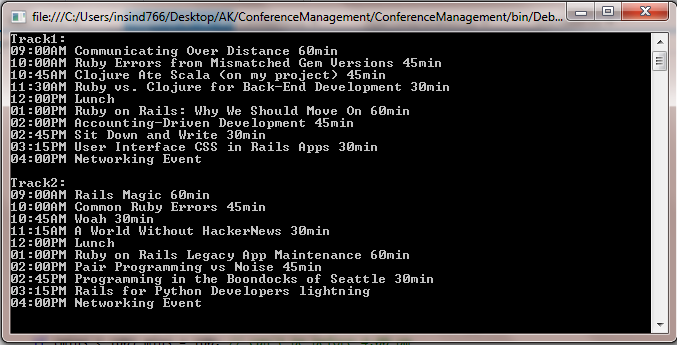
**Output 2:**

In Case of Invalid Input, user gets option to continue for Valid Input or to exit. On Click of Yes, user will be proceeded with Valid Inputs. Screenshots for the same.









On click of No, User will be exited.

