## Requirement

**Problem Two: 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 before 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.

***Test 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

***Test 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

## Requirement Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conference** | **Track** | **Sessions** | **Talks** | **Duration Constraint** |
| Conference | Track 1 | MoringSession | Talk1 Xmin | Start>=9AM End<=12PM |
| Talk2 Lighting |
| … |
| Lunch |  | Start=12PM End=1PM |
| EveningSession | Talk1 Xmin | Start>=13PM End<=5PM |
|
|
| NetworkEvent |  | 4PM<=Start<=5PM |
| Track 2 | MoringSession | Talk1 Xmin | … |
| Talk2 Lighting |
| … |
| Lunch |  |
| EveningSession | Talk1 Xmin |
| Talk2 Lighting |
| … |
| NetworkEvent |  |
| … |  |  |  |

Note: Talk Title can’t contain number

### Talk

|  |  |  |
| --- | --- | --- |
|  | Type | Length constraint |
| Talk | NormalTalk | 1minutes<=Legth<=180minutes |
| LightingTalk | 5miuntes |

Note: Talk Title can’t contain number

Quesiton:

Normal talk must be more than 5 minutes?

### Session

|  |  |  |
| --- | --- | --- |
|  | **Type** | **Length constraint** |
| Session | MoringSession | Start>=9AM End<=12PM |
| Lunch | Start=12PM End=1PM |
| EveningSession | Start>=13PM End<=5PM |
| NetworkEvent | 4PM<=Start<=5PM |

### How Schedule the Talks

## Schedule Algorithm

1. Calculate all Tracks count for all talks
2. Sort all talks by length from long to short(descending)
3. Try to assign the talk to morning sessions of all tracks (check the current session’s remain time is enough for the talk ,if there is no enough time ,it failed and try assign to next track’s morning session )
4. If 3) step failed try to assign the talk to evening sessions of all tracks (check logic as same the morning session)
5. After assign all talks , assign the networking event.

### Problem

The algorithm need to improve.

## How to use source code

How run the application?

### Console

When you run the application.

1. Build the solution.

2. Run the application TW.CTM.Shell.exe in output directory which named 'bin' in the solution directory.Note:

1. You must input the path of talks List text file as first parameter.

2. The talks List file must be encoding with unicode with BOM.

3. You can give a second parameter for output text file path.

Example:

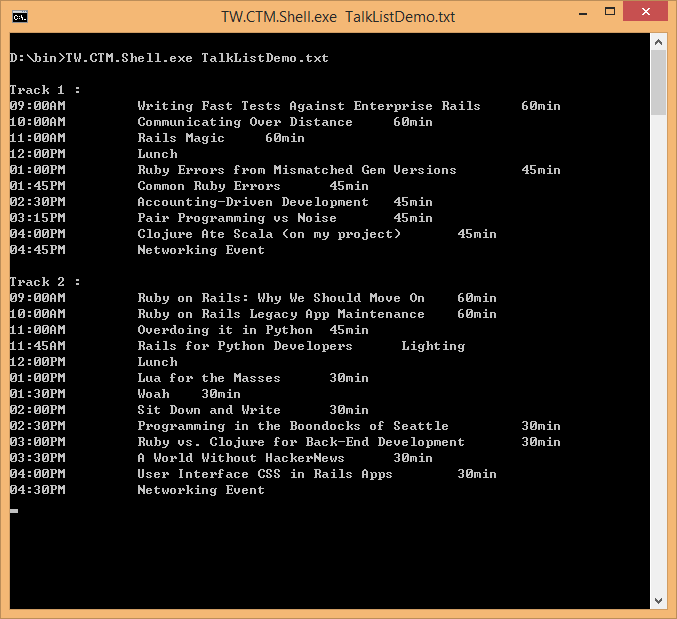
1. You have talks list text file. you don't want to export the result to file:

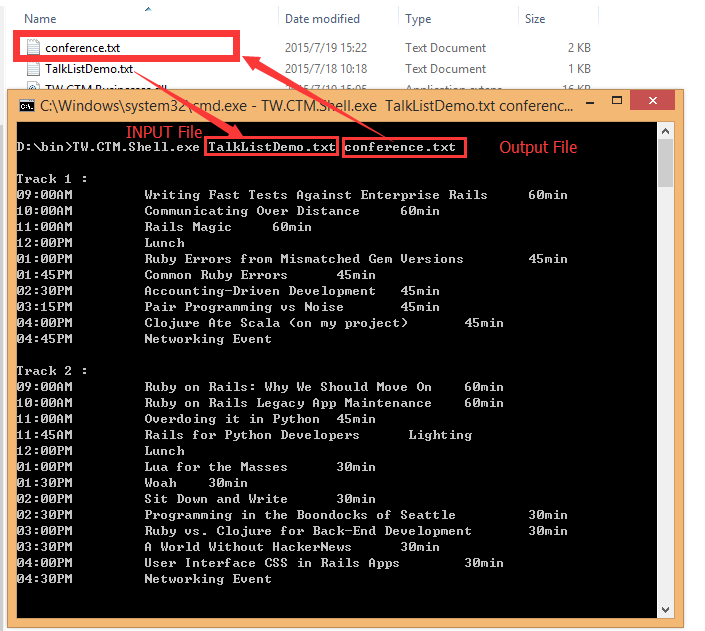
D:>TW.CTM.Shell.exe TalkListDemo.txt

2. You have talks list text file ,and want to output the result to file

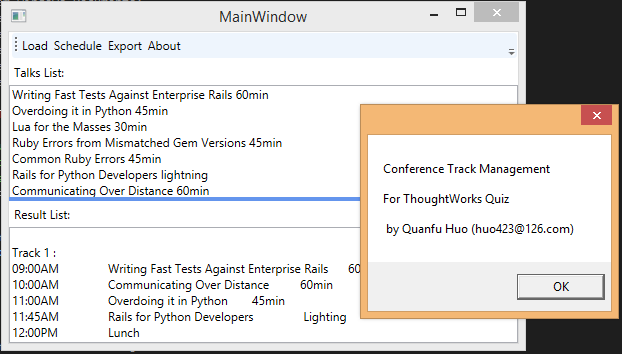
D:>TW.CTM.Shell.exe TalkListDemo.txt confrence.txt

As follow illustrations:





### WPF UI

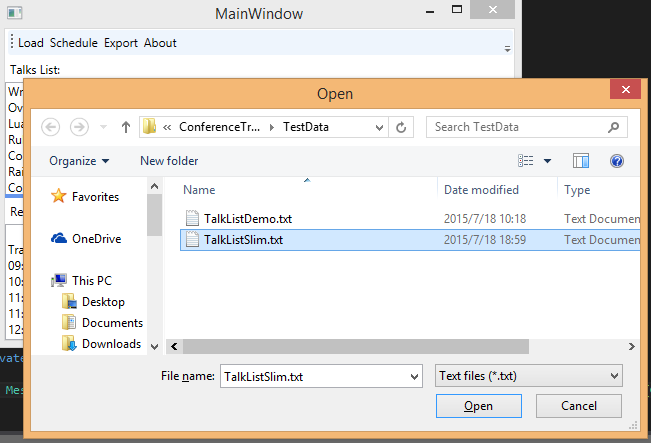


#### How to use?

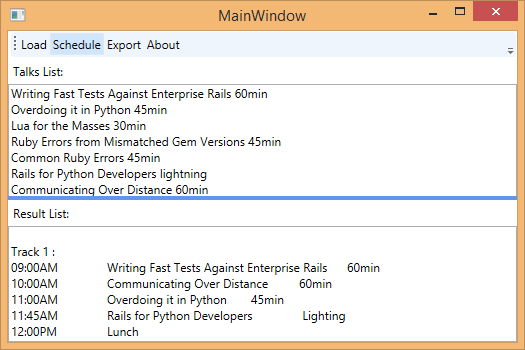
[Load] load talks list from a text file which is Unicode Encoding. Read all talks into Talk list box. [Schedule] Schedule the talks to conference and will show result in result list box at bottom.

[Export] Export the result to a txt file.

##### Load dialog:



##### Schedule



##### Export dialog

