Programming Case – Pizza Cabin Inc.



**Background**

The restaurant chain Pizza Cabin Inc. (PCI) has grown a lot recently. Now they are so big that they have some 850 people, known as Pizza Experts, taking pizza orders via phone. To create optimal schedules for their Pizza Experts they use a state-of-the-art workforce management (WFM) system. This system takes into account the demand for service and schedule start, end, break and lunch times accordingly.

The Pizza Experts are divided into teams of 8 to 16 people, and each team has a team leader. The team leaders are responsible for daily coaching, absence reporting, etc.

**Requirement**  
Team leaders need to schedule a **15-minute meeting** with the team every day. Create a C# Console Application that will display the **possible times when a meeting can be scheduled**.  
  
The program should firstly prompt the user to enter how many people are required for a meeting.

The program should analyze the schedules for the day and output **all suitable 15 minute intervals** (possible start times are 00, 15, 30 and 45) for the meeting where:

1. There are enough people at work
2. People are not on a break (either lunch or a short break).

**Implementation Details**

You are required to get the schedule data from a file (schedule.json) that is provided to you. The file contains JSON data as shown in the following snippet:

{

"ScheduleResult":{

"Schedules":[

{

"ContractTimeMinutes":480,

"Date":"\/Date(1450051200000+0000)\/",

"IsFullDayAbsence":false,

"Name":"Daniel Billsus",

"PersonId":"4fd900ad-2b33-469c-87ac-9b5e015b2564",

"Projection":[

{

"Color":"#1E90FF",

"Description":"Social Media",

"Start":"\/Date(1450080000000+0000)\/",

"minutes":120

},

{

"Color":"#FF0000",

"Description":"Short break",

"Start":"\/Date(1450087200000+0000)\/",

"minutes":15

},

|  |  |
| --- | --- |
| Schedule Result | All Schedule Data for the requested Day |
| Schedules | One Schedule per Person for the Day |
| Projection | A list of activities that a Person will do during the Day. |
| (Activity) | The ‘*Start’* attribute contains the DateTime that the activity will begin.  The ‘*minutes’* attribute is how long the activity will last.  The ‘*Description’* attribute is the type of activity. |

**Can we have a Meeting?**

In the following picture, activities are shown sequentially to give a view of part of a days scheduled activities:



\*SB = ‘Short break’

Imagine that we **require all three people in the above picture** to attend the 15 minute meeting.

* Bill Gates starts work at 8:00am, but no one else will be working at that time so we cannot have a meeting then.
* We cannot have a meeting at 9:45 as Bill Gates is on a ‘Short Break’.
* We cannot have a meeting between 10:30 – 13:00 as someone always has a ‘Short Break’ or are at ‘Lunch’.

Therefore, the output of the program for the subset of data shown above would be:

**9:00, 9:15, 9:30, 10:00, 10:15, 13:15, 13:30**