C++ OOP – Regular Exam – 25 August 2024

3. Calendar Manager

You are tasked to help completing a *Calendar Manager* program. The *Calendar Manager* does what the name implies: it manages *Calendar*, which contains *Meetings* and *Participants*. Each *Participant* can be onlya *Contributor*, (for now, other types of Participants, such as *Resource* , do not exist, but the class hierarchy is there).

The *Meetings* have start date and time, duration, and a list of *Participants*. When read from the console, they're ordered by their **start time and length**, e.g., from two meetings that start at the same time, the one in higher order is the shorter one.

The *Calendar Manager* holds all *Meetings*, ordered by their start times.

The skeleton provides the definition and the implementation of some of the needed classes. Your task is to study the input and output below and the provided skeleton code, and to complete the program by adding the necessary **.H** or .**CPP files**, so that the program compiles and runs successfully.

Input

The input contains the whole calendar and all participats:

- The list of all the meetings
  - The meeting data format is **`M MeetingID YYYY MM DD HH mm LH LM`**. For example **`M WelcomeMeeting1 2024 08 07 15 00 02 00`** describes a meeting ("**`M`**") with ID "**`WelcomeMeeting1`**", which starts of August 7, 2024 ("**`2024 08 07`**"), at 15:00 ("**`15 00`**") and continues two hours ("**`02 00`**").
- The list of all participants (contributors) for each meeting.
  - The Contributor data format is "**`C MeetingID FirstName SecondName`**". For example "**`C WelcomeMeeting1 Stamat Stamatov`**" decsribes a contributor to the meeting **`WelcomeMeeting1`**, with names "**`Stamat Stamatov`**".
- The input data ends with a single line with contents single dot: **`"."`**

The program reads all information and initalizes the Calendar structure with the data it has read.

Output

After reading the input, the program analyzes the Calendar data and outputs a list of all *Problems*, one problem on each line. The output format for each *Problem* is:

- For a Participant (Contributor), signed up for a non-existing meeting:
  - "**`{FirstName} {SecondName}: No meeting {MeetingID}.`** ", for example "**`Stamat Stamatov: No meeting StrangeMeeting.`**"
  - This processing happens first, during the reading of a Participant (as at that time all Meetings are already read).
- For empty meeting (meeting with no participants), the output should be: "**`Meeting {MeetingID} has no participants.`** ", for example "**`Meeting WelcomeMeeting2 has no participants.`**"
- For overlap in Participant's calendar:
  - For a Contributor, which has overlapping meetings, the output should be "**`{FirstName} {SecondName} cannot attend {MeetingID} meeting: still in {MeetingID}`**

meeting.", for example "**Stamat Stamatov cannot attend WelcomeMeeting2: still in WelcomeMeeting1 meeting.**"

- For a rare case where the whole calendar has no issue, the output will be a single line: "**No errors.**"

The order of verifications is as follows:

1. During the reading of the participants we check if a participant is asked to take part in a meeting with invalid id
2. Once participants are all read, the meetings list is are checked for empty meetings
3. Finally, each participant is checked for meeting overlap. There cannot be more than one overlap (e.g., a participant can't be asked to sit in three or more meetings at the same time).

Note:

1. Although the output may contain overlaps and invalid meetings, the format of the output will always be semantically correct, e.g. you cannot expect six-digit meeting year, or start hour 25:67
2. **MeetingID** is alpha-numeric, case-sensitive small and big letters and numbers, without space, e.g. meeting IDs **meeting1** and **Meeting1** are different
3. There will be no duplicate meetings or participants (Different contributors with the same First and Last name)
4. There will be no participants with the same **FirstName** and **SecondName**
5. There will be no more than 999 meetings in the input

Examples

Input	Output
M DesignReview 2024 09 12 09 00 02 00 M CodeReview 2024 09 12 10 30 01 00 C DesignReview Alice Smith C CodeReview Bob Johnson .	No errors.
M DailyMeeting 2024 09 10 09 30 00 30 M StrategyMeet 2024 09 10 10 00 01 00 M EmptyMeeting 2024 09 10 14 00 01 30 C DailyMeeting John Doe C StrategyMeet John Doe .	Meeting EmptyMeeting has no participants.
M Planning 2024 09 11 10 00 02 00 M ClientMeeting 2024 09 13 11 00 02 00 M InternalDiscussion 2024 09 13 11 30 01 00 C Planning Alice Smith C Planning John Doe C NonExistent John Doe C ClientMeeting John Doe C ClientMeeting Jane Doe C InternalDiscussion John Doe .	John Doe: No meeting NonExistent. John Doe cannot attend InternalDiscussion meeting: still in ClientMeeting meeting.