To help track down a hacker who has compromised multiple user accounts, we would like a tool that displays (potentially suspicious) activities during a certain period of time. How would you design a tool such that it can find/display activities during a certain period of time efficiently?

The goal of HW5 is to manage the activities and allow the user to specify a time range to display the corresponding activities. Also, we would like the user to be able to add and remove activities (e.g. from different sources such as log files from applications, network cards, and the operating system). To improve efficiency, your implementation uses a SkipListMap class that has at least the following operations:

- `get(k)` [p. 403]
- `put(k, v)` [p. 403]
- `remove(k)` [p. 403]
- `ceilingEntry(k)` [p. 428]
- `floorEntry(k)` [p. 428]
- `subMap(k1, k2)` [p. 428]

Use DoublyLinkedList [p. 135], which you would modify, to implement the SkipListMap class. You may also implement your own DoublyLinkedList. Use FakeRandomHeight for `put(k, v)` (to facilitate easier debugging and testing). Program files for DoublyLinkedList and FakeRandomHeight are on the course website.

**Input:** Input is from the command-line arguments for HW5.java:

- `filename` of actions, each line has one of the following actions:
  - `DisplayActivity time`
  - `AddActivity time activity`
  - `DeleteActivity time`
  - `DisplayActivitiesBetweenTimes startTime endTime`
  - `DisplayActivitiesFromStartTime startTime`
  - `DisplayActivitiesToEndTime endTime`
  - `DisplayAllActivities`
  - `PrintSkipList`

For simplicity, times are in HHMMSS format (HH is 00-23, MM and SS are 00-59) [leading zeros are optional]. You may assume the times are unique (a more detailed timestamp would be unique in the real-world). Sample input is on the course website.

**Output:** Output goes to the standard output (screen), each line has a result for the corresponding action:

- `DisplayActivity time activity/none`
- `AddActivity time activity [existingTimeError]`
- `DeleteActivity time [NoTimeError]`
- `DisplayActivitiesBetweenTimes startTime endTime time1:activity1 ... or none`
- `DisplayActivitiesFromStartTime startTime startime1:activity1 ... or none`
- `DisplayActivitiesToEndTime endTime time1:activity1 ... or none`
- `DisplayAllActivities time1:activity1 ... or none`
- `PrintSkipList`
  - `(Sh) empty`
  - `(S1) time1:activity1 ...`
  - `(S0) time1:activity1 ...

Sample output is on the course website.

**Submission:** Submit HW5.java that has the main method, SkipListMap.java, (modified) DoublyLinkedList.java, FakeRandomHeight.java and other program files. Submissions for Individual and GroupHelp have the same guidelines as HW1.

Note the late penalty on the syllabus if you submit after the due date and time as specified at the top of the assignment.