Qatar University
Dept. of Computer Science& Engineering
Data Structure (CMPS303)

CMPS303- Term Project
Events Management System

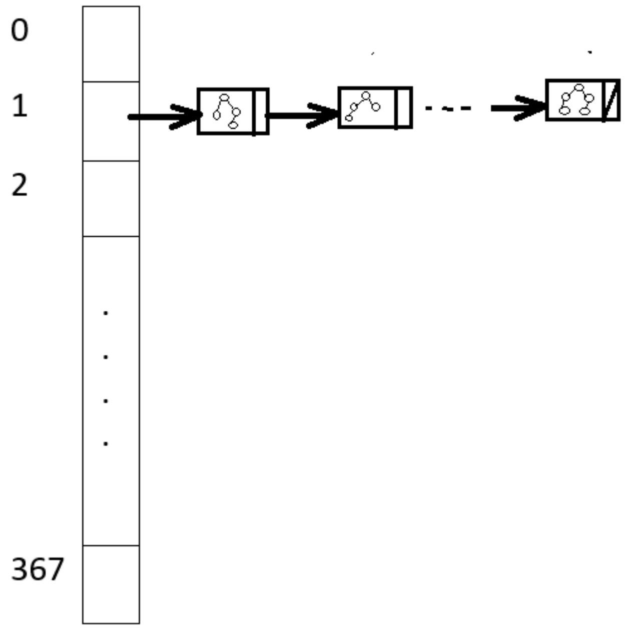
Instructions:

1- This is a teamwork project; each team should have maximum four students.
2- Ensure that the work you submit is original. Plagiarism detection software will be used to check for similarities with other submissions. Any attempt to plagiarize will result in the loss of all marks for the assignment and will be reported to the department.
3- You should only use the implementation of data structures that we had in the lab. Any other implementations or using Java collections classes will not be accepted.
4- Only one student out of the group should submit the work as a zipped file (From exporting the project), put team names as comments in the beginning of the class that has the main method (**Test.Java**).
5- The deadline **is Saturday 29/11/2025 at 11:59pm.**

You need to build events management system that manages events. The system will involve creating, updating, and searching events, and it will use multiple data structures to efficiently handle different aspects of event management, such as storing events by date, searching for attendees, etc.

You need the following classes:

1- Class Attendee: it has the following data: id(int), name (String).
2- Class Event: it has the following data: day(int), month(int), year(int), title(string), attendees (Tree). Every node in the tree represents an attendee, the key is attendee's id.
The class has the following methods:
- **addAttendee(Attendee a)** which inserts attendee a to the tree.
- **search (int k)** returns true if the attendee with id=k is in the tree (event's attendees), false if not.
- **attendeesList():** returns an array list contains all attendees in the event, the list should be based on pre-order approach.
3- Class **EventsHashTable**: this is a hash table structure, its size is 367 (to represent 366 days of the year+ position 0 is not used), therefore the hash function should receive day, month, year and returns the day number within the year.
(Hint: you can use method getDayOfYear())



The hash table is based on separate chaining approach, where each element in the table is a singly linked list (so it is an array of SinglyLinked Lists). Every element in the list is an event.
The class has the following methods:
-addEvent(int day, int month, int year, String name) which creates a new event, and insert it to the hash table based on the hash function that maps the date to the day of the year.
-addAttendee(String eventName, int id, String name): the method creates an attendee object, and inserts this object in the tree in an event that has eventName. You should search the table for that even, if there is no event with eventName, display an error message. (Assume the events has unique names).

-removeEvent(String eventName): which receives a string represents the event name, and removes the node in the linked list that has the data of that even. If there is no event with that name, it displays an error message.
-attendeesList(int day, int month, int year): which receives day, month, year and returns the names of all attendees in all events in that date.
-attendeeEvents(int id): which receives an attendee's id and returns array lists of the names of all events that attendee registered in.

4- You are required to develop an application (call it **Test.java**) that will interact with the user. It creates an object of **EventsHashTable** and displays the following menu:
Enter your choice:
1- Add a new event.
2- Register attendee in an event
3- Display attendees list in a specific day
4- Display attendee's events.
5- Remove an event.
6- Exit

- When the user selects 1, your application should ask for event's data and add it to the hash table.
- When the user selects 2, your application should ask for attendees' data, and the event name, and add the attendee to the tree in that event.
- When the user selects 3, your application should ask for day, month, year and displays all attendees' names in all events in that day.
- When the user selects 4, your application should ask for attendee's id and displays all events that this attendee register in.
- When user selects 5, your application should ask for event name, and removes that event from the system.
- When the user selects 6, your application should save the data structure object "EventsHashTable" to a file (Object serialization).

Note: when your application starts, it should search for the file (if it exists) that contains the data structure object and upload it to memory. There will be no file when we execute the application for the first time.

Instructions:
- You need to submit your code as exported project.

Grading:
30%: The project submitted code (group task), your project should work

70% : Your ability to update the submitted project (individual task).