ECS 40 Project 5

New concepts: linked lists.
Executable name: cal.out Due: 7th May at 11:55 PM
File names for calendar: calendar.cpp, day.cpp, appointment.cpp, time.cpp, year.cpp, dayofweek.cpp, linkedlist.cpp, day.h, appointment.h, time.h, year.h, dayofweek.h, linkedlist.h, and Makefile.

This is an extension of program #4. You are to modify program #4 so that the user can delete days from the Year, and the days array in the Year class is now a linked list object of the LinkedList class. You can use my source code from program #4 as the basis for your program #5.

Specifications and hints:

1. Class ListNode should be declared in linkedlist.h, and implemented in linkedlist.cpp.
   - 1.1. A ListNode contains a Day, and a next pointer.
   - 1.2. You will need to write a copy constructor for Day.
   - 1.3. There are no public methods in ListNode.
2. Class LinkedList is a singly linked list sorted by date.
   - 2.1. Class LinkedList is a friend of class ListNode.
   - 2.2. You must have a constructor and destructor for the class.
   - 2.3. You will have two overloaded[] operators—one const, and one not.
   - 2.4. You will have an operator+= to insert a Day in the list. This operator may not use the overloaded[] operator.
     - 2.4.1. Since the list must be sorted, you will have to write Day::operator< based on the date to make this elegant.
     - 2.4.2. This operator should completely replace the need for Year::addDate(), though you will need to increment Year::count after the call.
     - 2.4.3. Once you have written the [] and +=, and changed Year::AddDate() calls, your original program should compile and run!
   - 2.5. You will have an operator -= to remove a Day from the list. This operator may not use the overloaded[] operator.
     - 2.5.1. You must write a Day::operator== to make this elegant.
3. Class Year no longer needs the size data member.
   - 3.1. You will no longer need a destructor for Year.
   - 3.2. To remove a day, you will provide a Year::operator -=.
     - 3.2.1. I found I needed to place the code of case 3 in main in {} to eliminate an unusual error.
   - 3.3. Don't forget to update your dependencies in the Makefile.
4. The output of your program must be identical to that of mine.
5. Your Makefile must use the –ansi –g and –Wall options for all compiling and linking. Your program should compile with no warnings.
6. **const** must be used wherever possible.

```
$ cal.out
Calendar Menu
0. Done.
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Remove a date.
Your choice >> 2

Please enter the subject >> SADVC
Date                          Start End   Subject         Location
Friday, September 26          13:00 18:00 SADVC           SADVC (Woodland)
Wednesday, October 1          12:00 12:00 SADVC           Woodland
Friday, October 3             13:00 18:00 SADVC           SADVC (Woodland)

Calendar Menu
0. Done.
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Remove a date.
Your choice >> 4

Please enter the month and day (mm/dd) >> 9/26
Calendar Menu
0. Done.
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Remove a date.
Your choice >> 2

Please enter the subject >> SADVC
Date                          Start End   Subject         Location
Wednesday, October 1          12:00 12:00 SADVC           Woodland
Friday, October 3             13:00 18:00 SADVC           SADVC (Woodland)

Calendar Menu
0. Done.
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Remove a date.
Your choice >> 4

Please enter the month and day (mm/dd) >> 1/3
1/3 not found.

Calendar Menu
0. Done.
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Remove a date.
Your choice >> 0
$
```