In this project, you are going to introduce a Netflix-like (or Amazon prime video/Hulu/Apple+/Disney+) menu, composed by categories, each of which consists of a list of movies. Namely, there are a number of rows. For each row, users can browse movies in a circular way. For all rows/categories, users could browse up and down but not in a circular way.

With the above background, please implement a Netflix-like menu using

(1) Double Linked List (DLL) for connecting the beginning of all rows/categories;

(2) each row/category is constructed by a Circular Linked Lists (CLL). A possible illustration of the above concepts is shown at the end of the file. Note that there is more than one way to implement the relationship between DLL and CLLs here (and the header of a CLL is not shown here because of a variety of possible implementations. You are required to introduce a header for each CLL).

**For the entire Netflix, here are the features included:**  
a. Add a new category by name (e.g., void addCategory(string name))  
  
b. Search a category by name. If the category is found, return the pointer of the category. If the category is not available, return a null pointer.  
  
c. Search a movie by name. In this case, users do not know the category that this movie belongs to. All they know is movie name.  
  
**d.**Remove a category by name. If the category is available, remove all the movies within the category. Also, remove the header node of this category/CLL. The category itself should be deleted, too.  
  
**e.**Delete all the movie before a date (e.g., void deleteMovies(Date\* date)). Each movie has a title and an uploaded date from Netflix. All the movies uploaded prior to date should be removed by this function.  
  
**f.** Print all the movies of all categories.  
  
For each category/CLL, here are the features included:  
  
a. Add a movie to the category in alphabetic order (based on movie title). As mentioned, a movie should have a title and uploaded date when it is added to the category. You should check if there exists a movie that has the same title already. If the movie already exists, do nothing. Otherwise, add this movie to the category so that all the movies are listed in alphabetic order.  
  
b. Search a movie by name. Note that you are required to implement binary search algorithm here.  
  
c. Get movie’s index. In this case, index is counted in clockwise order. For example, in category “children”, Barbie is at index 0; Harry Potter is at index 1; Lion King is at index 2; and  
  
d. Print all the movies of the category. In addition to above’s two classes, you will need Date and Node classes at least. You are welcome to reuse your lab work. You should at least check whether a date falls within valid range, especially for leap year case.  
  
A main function should be included in the project that calls all the functions at least one