Continuous Assessment 1

The FlixNet Movie Library

1. Design and implement a class Movie, with fields name, director, dateadded and viewed(and you could have other fields for e.g. the cast list, the releasedate, the actual video file, the size of the file, the genre, etc, but we don't need them in this exercise). The dateadded field will contain 8 digit numbers in the form YYYYMMDD (technically, DD should not go above 31, MM above 12, and YYYY above 2017, but we will ignore that). Your Movie class should have an __init__(...) method, a __str__() method, a get_title()method and a play() method to simulate the movie being played -- for this exercise, all it needs to do is ensure that viewed is set to True, and then return a string representation of the movie.

2. Design and implement class FlixNetLibrary, which is essentially a sequence of movie objects. The library should offer the following methods (in addition to __init__(...)):

- __str__(), which should return the title of each movie in sequence in the string, indicating the current selected movie, if any
- add_movie(movie), which should add a new movie to the library in decreasing order of dateadded
- get_current(), which should display the currently selected movie info
- next_movie(), which should change the current selection to the next one
- prev_movie(), which should change the current selection to the previous one
- reset(), which should reset the curent movie to the list head
- play(), which plays the currently selected movie - it should print a message saying the movie is being played, and should call the movie's own play() method
- remove_current(), which should remove the current movie from the library
- length(), which should report the number of movies in the library

The class FlixNetLibrary must be implemented as a doubly-linked list. You can create a separate Doubly-Linked List class, and then use an instance of that class as a field inside your FlixNetLibrary, or you can implement FlixNetLibrary to deal directly with a sequence of DLLNodes.

For this exercise, for invalid input, if we are obliged to return an item, we will return None, and otherwise we will do nothing. If you delete the currently selected movie, make whetever came after it the new current selection; if that goes off the end of the list, reset the current movie to be the head.

3. Test your class using the following sequence of operations:

i. create an empty library
ii. create the movie ("Bladerunner 2049", "Villeneuve", 20171004, False) and add it to the library
iii. create the movie ("Hail, Caesar!", "Coen & Coen", 20160304, False) and add it to the library
iv. create the movie ("Wonder Woman", "Jenkins", 20170601, False) and add it to the library
v. display the library to the screen

vi. set the current movies to be the next one (i.e. first in library)
vii. play the current movie
viii. move current to the next movie
ix. report the current movie
x. move the current to the previous movie
xi. delete the current movie
xii. display the library to the screen
xiii. create the movie ("The Imitation Game", "Tyldum", 20141114, False) and add it to the library
xiv. move current to the next movie
xv. play the current movie
xvi. display the library to the screen

which should give you output something like:

```
---
"Bladerunner 2049"
"Wonder Woman"
"Hail, Caesar!"
---
Currently playing: ("Bladerunner 2049", "Villeneuve", 20171004, True)
Current movie: ("Wonder Woman", "Jenkins", 20170601, False)
---
>>>"Wonder Woman"
"Hail, Caesar!"
---
Currently playing: ("Hail, Caesar!", "Coen & Coen", 20160304, True)
"Wonder Woman"
>>>"Hail, Caesar!"
"The Imitation Game"
---
```

Note that you should also test using more extensive test methods than this. We will be using different methods to test your submitted software.

4. Extend your design and implementation of the library to allow search for movies by substrings of the movie name or the director name. The methods should search the list from the current movie onwards, and wrap around. If a matching movie is found, the current movie should be updated, and the method should return True; if no movie is found, return False.