Community College of Philadelphia

CSCI211 – Data Structures and Algorithms
ASSIGNMENT 4

Instructor: Michael Hackett
Department: Computer Science
Email: mhackett@ccp.edu

Hurricane List Sort and Search

Using C++, Java, or Python, design a program that creates a doubly linked list of nodes where each node contains an object with information about a hurricane/tropical storm for a certain year. You may choose any year between 1980 and 2018, for either the Atlantic or Pacific Ocean. (Be sure there were at least 5 named storms during the year you choose.) **You may not use a built-in List ADT**.

After your program creates the list, your program will indefinitely prompt user to enter one of the following five commands which will perform the following:

- **sortwind** – Sorts the linked list by the storm's maximum windspeed (ascending order) and prints out all storms for the year.
- **searchname** – Prompts the user to enter a name, then searches the list and displays information for that storm
- **searchcategory** - Prompts the user to enter a category, then searches the list and displays information for that storm (or storms if more than one)
- **searchmonth** - Prompts the user to enter a month, then searches the list and displays information any storms in that month
- **exit** – Ends the program.

The Driver Class/Main

In a main function, you'll create an instance of your linked list. After researching information about the hurricanes in your chosen year, one at a time you will make a Storm object and add it to the list. Using literal data in your source code is fine (and preferred) when creating Storm objects; The program does not need to rely on user input for names, wind speeds, etc.

Fill your linked list with Storm objects, then prompt the user for their command. When the user enters **exit**, the program should finish. (See next page for sample input/output)

Be sure to print error messages for invalid months and categories. If there are no storms with a specified name, month, or category, print an error message. An error message should also be printed for invalid commands. The program should continue in either case.

Sample Input/Output

```
Command: sortwind
Hurricane B - Wind Speed: 75 MPH; Month Formed: August; Category 1
Hurricane A - Wind Speed: 76 MPH; Month Formed: October; Category 1
Hurricane C - Wind Speed: 100 MPH; Month Formed: August; Category 2
Hurricane D - Wind Speed: 160 MPH; Month Formed: September; Category 5

Command: searchname
Enter name: C
Hurricane C - Wind Speed: 100 MPH; Month Formed: August; Category 2

Command: searchname
Enter name: F
No storms found.

Command: searchcategory
Enter category: 3
No storms found.

Command: searchcategory
Enter category: 9
Invalid category.

Command: searchcategory
Enter category: 1
Hurricane B - Wind Speed: 75 MPH; Month Formed: August; Category 1
Hurricane A - Wind Speed: 76 MPH; Month Formed: October; Category 1

Command: searchmonth
Enter month: August
Hurricane B - Wind Speed: 75 MPH; Month Formed: August; Category 1
Hurricane C - Wind Speed: 100 MPH; Month Formed: August; Category 2

Command: searchmonth
Enter month: Septober
Invalid month.

Command: searchmonth
Enter month: July
No storms found.

Command: exit
```

The commands should be able to be entered in any order and more than once.
Submit any and all related source code files in the Assignment 4 submission link.

Grading

See Assignment Rubric in Canvas.

The following is the same as Assignment 3

Nodes

Each node (implemented as either a struct or a class) will contain one data field:

- s (Storm) – A Storm object (details below)

Since it is a doubly linked list, each node needs a next and previous pointer. For simplicity, the data field and pointers can all be public.

Storm Object/Class/Struct

Design a class/struct named Storm that contains the following data fields:

- name (a String) – The name of the hurricane/tropical storm
- maxWind (int) – The maximum recorded windspeed of the storm
- monthFormed (a String) – The month in which the storm formed
- category (int) – The storm's category (1-5); For tropical storms, use 0 as the category

For simplicity, the data fields can all be public.

The Linked List

Your linked list will need head and tail pointers. It will need a default constructor that sets both the head and tail to null. It will also need the following functions:

void push_front(Storm h)	- Prepends this Storm object h to a new node at the head of the list
void push_back(Storm h)	- Appends this Storm object h to a new node at the tail of the list)
void insert(Storm h, int i)	- Inserts this Storm object h to a new node and places it at position i
void erase(int i)	- Removes the node at position i
void printForward()	- Prints information about each storm from head to tail
void printReverse()	- Prints information about each storm from tail to head

When printing output about a storm, the format to use is (bold text indicates data stored in the Storm objects):

`Hurricane` **`Michael`** `- Wind Speed:` **`160`** `MPH; Month Formed:` **`October`**`; Category` **`5`**

When printing the name, be sure to print `Hurricane` or `Tropical Storm` depending on the storm's category.