Extra Lab - Introduction to CSCI 2270 Data Structures

Due on: December 13^{th}, Friday 11:59 pm

Extra Credit

Overview: In this question, you will write a program that:

- Read a "**.csv**" file containing information on national parks.
- Store the information in an array of structs.
- Print the content where the **area of the park** is greater than the minimum value.

Task: National parks

Create an array that holds the **Park struct objects**. Use the following struct declaration:

```
struct Park {
    string parkname;
    string state;
    int area;
};
```

Write a function named **addPark**:

a. The **addPark** function has the following signature:

```
// length: Number of items currently stored in the array
void addPark(Park parks[], string parkname, string state, int area,
             int length);
```

b. Instantiate a struct and store the **parkname, state, area** values in it.

c. Add the struct to the **parks** array.

Write a function named **printList**:

a. The **printList** function has the following signature:

```
// length: Number of items in the array
void printList(const Park parks[], int length);
```

b. Loop through the **parks** array.

c. Print out each element of the **parks** array in the following format.
"<PARKNAME> [<STATE>] area: <AREA>" using the below cout statement

```
cout << park.park_name << "[" << park.state << "] area: " <<
park.area << endl;
```

Example, "Badlands_National_Park [SD] area: 242756"

Write a **complete program** which includes the following:

1. The **park struct** and the **addPark, printList** functions coded above.
2. A **main()** function defined as below:
    1) Input file needs to be stored in the same directory as your program.
    2) Read from the input file, "***park.csv***":
        a. Parse each line of the file using **getline** function (link: getline) and convert each entry into its appropriate data type. **park_name** should be a string, **state** should be a string, and **area** should be an integer. (*Hint: Use* ***stoi*** (link: stoi) *functions to convert from strings to numbers*)
        b. If **area** is more than 200000, call **addPark** function to update the **parks** array.
    3) Call the **printList** function after the array has been filled with data.
    4) Make sure you close the file when you are done.

	A	B	C	D
1	Arches_National_Park	UT	76519	
2	Badlands_National_Park	SD	242756	
3	Big_Bend_National_Park	TX	801163	
4	Biscayne_National_Park	FL	172924	
5	Black_Canyon_of_the_Gunnison_National_Park	CO	32950	
6	Bryce_Canyon_National_Park	UT	35835	
7	Canyonlands_National_Park	UT	337598	
8	Capitol_Reef_National_Park	UT	241904	
9	Carlsbad_Caverns_National_Park	NM	46766	
10	Channel_Islands_National_Park	CA	249561	
11	Congaree_National_Park	SC	26546	
12	Crater_Lake_National_Park	OR	183224	
13	Cuyahoga_Valley_National_Park	OH	32950	

Figure 1: File Contents: park.csv

```
Badlands_National_Park [SD] area: 242756
Big_Bend_National_Park [TX] area: 801163
Canyonlands_National_Park [UT] area: 337598
Capitol_Reef_National_Park [UT] area: 241904
Channel_Islands_National_Park [CA] area: 249561
Denali_National_Park_and_Preserve [AK] area: 3372402
Death_Valley_National_Park [CA] area: 4740912
```

Figure 2: Your print output

Instructions to submit your Lab work

Zip the cpp file and park.csv together and submit the resulting .zip file through Moodle as Extra Lab by due date. You do not need to submit any executable files.

Keep in mind the Honor code and ensure that you do not violate any of the rules it entails.