Assignment Objective: Build skills on C class creation while implementing a list of strings.

Requirements:

* Into two files, p1.h and p1.cpp, create a class called stringList with the following members
  + Private members:
    - string \*a // A pointer to the array into which the list values will be stored.
    - int listCapacity; // memorialize the capacity of the list
    - int listSize; // an integer to track the number of entries on the list
  + Public members:
    - constructor stringList(int listCapacity) that causes the array to be sized at “listSize” entries, with a default size of 100
    - destructor ~stringList() that deletes the dynamically allocated array “a”
    - bool insert(string text) – inserts the given text at the beginning of the list; all other entries shift right. Returns true if the text was inserted; otherwise, it returns false.
    - bool add(string text) – inserts the given text at the end of the list. Returns true if the text was added; otherwise, it returns false.
    - bool insertAt(int index, string text) – inserts the given text at the index “index”; values at that position need to shift right. Returns true if the text is inserted. Returns false if the list was full or if the index was greater than listSize.
    - bool deleteAt(int index, string &text) – deletes the text at the given index; shifts the entries right of that index to the left. If the index was within the range of the list, it sets “text” to the value of the item deleted and returns true. Otherwise, it does not change “text” and returns false.
    - void clear() – causes the list to be emptied
    - void printIt() const – causes the list to be printed, one value per line; for each line, print the index and the text at that index.
    - int getIndex(string text) const – returns the first position at which text was found; otherwise returns -1.
    - bool readAt(int index, string &text) const – same as deleteAt(), save that the value is not deleted from the list.
    - int count() const – returns the number of entries in the list
  + Note: All member functions that can be declared const should be declared as const.
* Demonstrate the correctness of the implementation by doing the following, using the p1m.cpp that is given to you on D2L:
  + Compile the code into the executable file p1.exe (or p1)
    - g++ p1.cpp p1m.cpp -o p1
  + Execute the code as follows, comparing it to p1CorrectOutput.txt

./p1 p1Input.txt > p1Output.txt

* **Submission process:**
  + Create a zip file containing the following files.
    - p1.h
    - p1.cpp
    - p1Output.txt
  + In class**:**
    - Print the above three files, in the order given and submit to class when due.
  + For the p1a submission, submit the highlighted functions completed, the others stubbed out.
  + For the p1b submission, submit a full implementation.