COMP 2710 Project 4
Points Possible: 100
Deadline: 11:59pm Nov. 5th, 2021 **(US Central Time)**

Goals:

- To learn how to use linked data structures (Note: **no array is allowed**)
- To use strings
- To learn creating multiple versions via conditional compilation
- To design and implement functions
- To perform unit testing

In this homework assignment, you will write a simple trivia quiz game. Your program first allows players to create their trivia questions and answers. Multiple questions should be organized and managed using **a linked data structured; no array is allowed in this homework assignment.** Then, your program asks a question to the player, input the player's answer, and check if the player's answer matches the actual answer. If so, award the player the award points for that question. If the player enters the wrong answer your program should display the correct answer. When all questions have been asked, the total award points that the player has won should be displayed.

Please perform the following steps to finish this assignment.

- **Step 1:** Create a TriviaNode **structure** that contains (1) information about a single trivia question and (2) a pointer pointing to other TriviaNode. This structure must contain a **string** for the question, a **string** for the answer to the question, an integer representing points the question is worth, and **a pointer of the TriviaNode type.** Please keep in mind that a harder question should be worth more points.

- **Step 2:** Create a linked list of Trivia using the TriviaNode structure defined in step 1.

- **Step 3:** Design and implement a function that initialize the Trivia linked list by hard-coding the following three trivia questions (including answers and award points). The
    - Trivia 1:
        - Question: How long was the shortest war on record? (Hint: how many minutes)
        - Answer: 38
        - Award points: 100
    - Trivia 2:
        - Question: What was Bank of America's original name? (Hint: Bank of Italy or Bank of Germany)
        - Answer: Bank of Italy
        - Award points: 50
    - Trivia 3:
        - Question: What is the best-selling video game of all time? (Hint: Call of Duty or Wii Sports)?

        - Answer: Wii Sports
        - Award points: 20

- **Step 4:** Design and implement a function to create and add a new TriviaNode into the linked list. You must use operator **new** to dynamic allocate memory to a new TriviaNode. Please remember to check that a new TriviaNode is successfully created.

- **Step 5:** Design and implement a function that asks a question to the player, input the player's answer, and check if the player's answer matches the actual answer. If so, award the player the dollar amount for that question. If the player enters the wrong answer your program should display the correct answer.
    - **Input:** a linked list of TriviaNode, the number of trivia to be asked in the list
    - **Output:** void or int – 0 indicates success and 1 indicates failure.

- **Step 6:** Write a test driver to perform unit testing for the function implemented in step 5. Assume there are three trivia in your created list, you must cover at least the following cases: (see **Fig. 1** on page 3 for the sample user interface.)
    - Case 1: ask 0 question
    - Case 2: ask 1 question (i.e., the first one) from the list
        - Correct answer
        - Wrong answer
    - Case 3: ask 3 questions (i.e., all the questions) from the list.
        - Correct answer
        - Wrong answer
    - Case 4: ask 5 questions that exceed the number of available trivia in the linked list (i.e., the index of the trivia is larger than the size of the linked list).

- **Step 7:** Write the main function that performs the following: (see **Fig. 2** on page 4 for the sample user interface)
    - Create hard-code trivia quizzes (i.e., questions/answers/awards) (Note: just call the function implemented in step 3).
    - Create more than 1 trivia quiz from a keyboard (Note: just call the function implemented in step 4).
    - Write a for loop; in each iteration do the following:
        - asks a question to the player
        - takes the player's answer
        - compares the input answer with the actual answer
            - if the player's answer matches the actual answer, then award the player the corresponding award points for that question,
            - else (i.e., the player enters the wrong answer) your program should display the correct answer.
    - When all questions have been asked, display the total award points the player has won.

- **Step 8:** Create two versions using conditional compilation.

- Version 1: simply run the test driver implemented in step 6.
- Version 2: a regular version run the main function implemented in step 7.
  Note: this version should not include the test driver.

You must provide the following user interface for the **debug version**. In this version, your program must run the test driver you build in step 6.

```
***This is a debugging version ***
Unit Test Case 1: Ask no question. The program should give a warning message.
Warning - the number of trivia to be asked must equal to or be larger than 1.
Case 1 Passed

Unit Test Case 2.1: Ask 1 question in the linked list. The tester enters an incorrect
answer.
Question: How long was the shortest war on record?
Answer: 85
Your answer is wrong. The correct answer is 38
Your total points 0

Case 2.1 passed

Unit Test Case 2.2: Ask 1 question in the linked list. The tester enters a correct
answer.
Question: How long was the shortest war on record?
Answer: 38
Your answer is correct! You receive 100 points.
Your total points: 100

Case 2.2 passed

Unit Test Case 3: Ask all the questions of the last trivia in the linked list.
Question: How long was the shortest war on record?
Answer: 38
Your answer is correct! You receive 100 points. Your
total points: 100

Question: What was Bank of America's original name? (Hint: Bank of Italy or Bank of
Germany)?
Answer: Bank of Germany
Your answer is wrong. The correct answer is Bank of Italy.
Your total points 100

show question here
Add your answer here
. . .
Case 3 passed

Unit Test Case 4: Ask 5 questions in the linked list.
Warning - There is only 3 trivia in the list.
Case 4 passed

*** End of the Debugging Version ***
```

Fig. 1: Sample user interface for the debug version

You must provide the following user interface for the **production version**. The user input is depicted as **Bold,** but you do not need to display user input in bold. Please replace "Li" with your name. In this version, your program must run the test driver you build in step 6.

```
*** Welcome to Li's trivia quiz game ***
Enter a question: enter your first question here.
Enter an answer: enter your first answer here.
Enter award points: enter your first award points here.
Continue? (Yes/No): Yes
Enter a question: enter your second question here.
Enter an answer: enter your second answer here.
Enter award points: enter your second award points here.
Continue? (Yes/No): No

Question: How long was the shortest war on record? (Hint: how many minutes)
Answer: 38
Your answer is correct. You receive 100 points.
Your total points: 100

Question: What was Bank of America's original name? (Hint: Bank of Italy or
Bank of Germany)?
Answer: Bank of Germany
Your answer is wrong. The correct answer is: Bank of Italy
Your Total points: 100

Question: What is the best-selling video game of all time? (Hint: Call of
Duty or Wii Sports)?
Answer: Wii Sports
Your answer is correct. You receive 20 points.
Your total points: 120

...
Display more questions/answers and information here...
...
*** Thank you for playing the trivia quiz game. Goodbye! ***
```

Fig. 2: Sample user interface for the production version

How to Create Two Versions?

You can use the preprocessor directive **#ifdef** to create and maintain two versions (i.e., a debugging version and a product version) in your program. If you have the following code:

```
1. #define UNIT_TESTING
2. #ifdef UNIT_TESTING
3. add your unit testing code here
4. #else
5. add your code for the product version here
6. #endif
```

in your program, the code that is compiled depends on whether a preprocessor macro by that name is defined or not. For example, if the `UNIT_TESTING` has been defined, i.e., line is enabled, then line 3 "`add your unit testing code here`" is compiled and line 5 "`add your code for the product version here`" is ignored. If the macro is not defined, i.e.,

line 1 is disabled, then line 5" `add your code for the product version here"` is compiled and line 3 `"add your unit testing code here"` is ignored.

These macros look a lot like if statements, but macros behave completely differently. More specifically, an **if** statement decides which statements of your program must be executed at run time, while a **#ifdef** controls which lines of code in your program are actually compiled.

Unit Testing:

Unit testing is a way of determining if an individual function or class works as expected. You need to isolate a single function or class and test only that function or class. For each function in this homework, you need to check normal cases and boundary cases.

Examples for tested values:

- String – empty string, medium length, very long
- Array – empty array, first element, last element
- Integer – zero, mid-value, high-value

You must implement a unit test driver for each function implemented in your program. You may need to use assert() to develop your unit test drivers if tested results are predictable.

Integration Testing:

Integration testing (a.k.a., Integration and Testing) is the phase in software testing in which individual software modules are combined and tested as a group. You may use the sample user interface illustrated in Fig. 2 on page 4 to perform an integration testing for your program.

Requirements:

1.(2.5 points) Use comments to provide a heading at the top of your code containing your name, Auburn UserID, filename, and how to compile your code. Also describe any help or sources that you used (as per the syllabus).
2.(2.5 points) Your source code file should be named as "project4_lastname_userID.cpp", e.g., project4_Liu_tzl0031.cpp.
3.(12.5 points) Your program must use structures and a linked list. (see steps 1-2)
4.(5 points) Your program must use string rather than char array. (see steps 1-2)
5.(5 points) A function that creates 3 hard-coding trivia quizzes. (see step 3)
6.(10 points) A function that creates new quiz from a keyboard. (see step 4)
7.(15 points) A function that asks a question and checks a player's answer. (see step 5)
8.(15 points) Write a test driver for function implemented in step 5.
9.(10 points) Correctly implement the main function. (step 7)
10. (10 points) Create two versions using conditional compilation.
11. (5 points) You must reduce number of global variables and data.
12. (5 points) Usability of your program (e.g., user interface).
13. (2.5 points) Readability of your source code.

Note: You will lose **at least 40 points** if there are compilation errors or warning messages when the TA compiles your source code. You will **lose points** if you: do not use the specific program file name, or do not have a comment, or don't have a commend block on program you hand in.

Programming Environment:

Write a short program in C++. Compile and run it using AU server (no matter what kind of editor you use, please make sure your code could run on AU server. The only test bed we accept is the AU server).

Deliverables:

Submit your source code file named as "project4_lastname_userID.cpp" through the Canvas system.

Late Submission Penalty:

- Late submissions will not be accepted and will result in a **ZERO** without valid excuses, in which case you should talk to Dr. Li to explain your situation.
- GTA/Instructor will not accept any late submission caused by internet latency.

Rebuttal period:

You will be given a period of 2 business days to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.