CSC 340.02 - Fall 2021 - Assignment 03
Due: 10/19/2021 11:55pm
Grade: 150pt - 15%

Guidelines for the assignment submissions:

What to submit?

- One zip/tar.gz file:
  - Name: `<First Name><Last Name>-Assignment-##.zip`
  - 3 directories total: each for 1 of the following questions
  - Source code should include only files that we create and edit (*.cpp and/or *.h)

How to submit?

iLearn, Assignment Submission section

1. Recursion (20pt)

Implement a recursive function to determine whether a string is palindrome or not, considering only alphanumeric characters and ignoring cases. Write a program to test your function by asking the user to input a string and determine if it's palindrome.

Example:

```
Input: s = "A man, a plan, a canal: Panama"
Output: true
Explanation: "amanaplanacanalpanama" is a palindrome.

Input: s = "race a car"
Output: false
Explanation: "raceacar" is not a palindrome.
```

2. TicTacToe class (40pt)

Create a class TicTacToe that will enable you to write a complete program to play the game of tic-tac-toe. The class contains as private data a 3-by-3 two-dimensional array of integers. The constructor should initialize the empty board to all zeros. Allow two human players. Wherever the first player moves, place a 1 in the specified square. Place a 2 wherever the second player moves. Each move must be to an empty square. After each move, determine whether the game has been won or is a draw. (35pt)

Readme and code style: (5pt)

Use the provided files for this exercise.

Sample output:

```
   0    1    2

0    |    |
 ____|____|____
     |    |
1    |    |
 ____|____|____
     |    |
2    |    |

Player X enter move: 1 1

   0    1    2

0    |    |
 ____|____|____
     |    |
1    |  X |
 ____|____|____
     |    |
2    |    |

Player O enter move: 2 1

   0    1    2

0    |    |
 ____|____|____
     |    |
1    |  X |
 ____|____|____
     |    |
2    |  O |

Player X enter move:
```

3. Stack (90pt)

   Write an implementation of the ADT stack that uses a resizable array to represent the stack items. Anytime the stack becomes full, double the size of the array. Maintain the stack's bottom entry at the beginning of the array. (40pt)

   - DEFAULT_SIZE for stack is 6

   Test your stack implementation: take a string input and action to perform from users, then output the result (40pt)

   1. Reverse the words in a sentence using your stack.
      - ○ Example: "this is fun" ⇒ "fun is this"

   2. Check if a string is palindrome using stack
      - ○ Example: as question 1

   Readme file and code style (10 pt)

   Use the included stack interface for this exercise.