CMIS 2720 Data Structures and Algorithms for Games

Instructor: Ying Zhu

Assignment #1

Due date: February 8, 2023

In this assignment, you will learn the following:

- How to query and manipulate a dynamic array
- How to write recursive functions

General information

1. You must write your programs in C#.
2. This assignment contains two separate programs: A1a and A1b.

A1a requirements

1. Implement a card guessing game in which a human player plays with a computer player.
2. From a deck of 52 playing cards, deal 13 random cards to the computer player.
    1. Use 11 for J, 12 for Q, 13 for K, and 14 for A
3. The human player can ask the computer three types of questions.
    1. The human player can ask how many cards the computer has in a specific suit. For example, how many cards do you have in Spade?
    2. The human player can ask how many cards the computer has in a specific value. For example, how many 7s do you have?
    3. The human player can ask for the sum of the cards in a specific suit. For example, what is the sum of all the Hearts?
4. The human player can ask X number of questions. You determine the optimal number of X that makes the game interesting.
5. The human player has 5 chances to guess the computer's cards. The goal is to make as many correct guesses as possible.
    1. The computer must respond if the guess is correct.
    2. If the guess is correct, the computer must drop that card from the hand.
6. Questioning and guessing can happen in any order.
7. Analyze the time complexity of your program and write it as a comment at the top of your program.

8. You must use object-oriented programming. You must define the following classes: CardGuessing (which contains the Main() method), GameManager, HumanPlayer, ComputerPlayer, and Card.

9. **You must use List<T> to store the cards. Use the methods in List<T> to implement the required functions**.

10. The user interface can be as follows. The human player can choose any

    1. Query
        1. Ask about the card count for a suit
        2. Ask about the card count for a value
        3. Ask about the sum for a suit
    2. Guess
    3. Quit

A1b requirements

1. Calculate and display the first 20 **odd** Lucas numbers.

2. **You must use recursive functions to calculate the Lucas numbers.**
    a. I will check the source code.
    b. If your program simply prints out the numbers, you will get 0 credit.

3. Your program output should be like this.
   > The first 20 odd Lucas numbers are: 1, 3, 7, 11, 29, 47, 123, 199, 521, 843, …

4. Analyze your algorithm's time complexity and write the result as a comment at the beginning of the program.

Lucas number is a number sequence similar to the Fibonacci number but with different starting numbers. For more information on Lucas numbers, see the following references:

- https://en.wikipedia.org/wiki/Lucas_number
- https://oeis.org/A000032

The first 20 Lucas numbers are 2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, 5778, 9349

Deliverables

1. Submit the C# source file to iCollege under the folder Assessments → Assignments → Assignment1. Your filename should be firstname_lastname_1a.cs and firstname_lastname_1b.cs.

Only submit the C# source file (.cs file). Do not submit the entire Visual Studio project. I will create the project myself.