CMPSCI 182L – Data Structures and Program Design Lab
Project 1 – Recursion and ADT's
30 points total
Due 9/20/22

1. This problem considers two ways to compute x^n for some $n \geq 0$.
   a. Write an iterative method ***power1*** to compute x^n for $n \geq 0$.
   b. Write a recursive method ***power2*** to compute x^n by using the following *recursive* formulation:

      $x^0 = 1$
      $x^n = x * x^{n-1}\ if\ n > 0$

   c. Write an accompanying main method which invokes both ***power1*** and ***power2*** for increasing values of n = 0 up to n = 10. For the purpose of this exercise, give x a fixed value of 2.0 and change only the value of n. You probably want to change the value of n in a loop, as I have done in my solution to this exercise. Here is an example test run of the main method I wrote to solve this exercise:

```
2.0^0 = 1.0 computed iteratively.
2.0^0 = 1.0 computed recursively.
2.0^1 = 2.0 computed iteratively.
2.0^1 = 2.0 computed recursively.
2.0^2 = 4.0 computed iteratively.
2.0^2 = 4.0 computed recursively.
2.0^3 = 8.0 computed iteratively.
2.0^3 = 8.0 computed recursively.
2.0^4 = 16.0 computed iteratively.
2.0^4 = 16.0 computed recursively.
2.0^5 = 32.0 computed iteratively.
2.0^5 = 32.0 computed recursively.
2.0^6 = 64.0 computed iteratively.
2.0^6 = 64.0 computed recursively.
2.0^7 = 128.0 computed iteratively.
2.0^7 = 128.0 computed recursively.
2.0^8 = 256.0 computed iteratively.
2.0^8 = 256.0 computed recursively.
2.0^9 = 512.0 computed iteratively.
2.0^9 = 512.0 computed recursively.
2.0^10 = 1024.0 computed iteratively.
2.0^10 = 1024.0 computed recursively.
```

      Notice that all of the computed values are identical, but ***power1*** computes its value using a loop, and ***power2*** computes its value by means of recursion (5 points).

2. The n^{th} Harmonic number is the sum of the reciprocals of the first n natural numbers:

$$H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n}$$

Write a **recursive** method and accompanying main method to compute and display the n^{th} Harmonic number. Your main method should prompt the user for the number n, and should pass this user-entered number to your recursive method. The main method should then display the number returned by your recursive method. Here is an example test run of the main method I wrote as a solution to this exercise:

```
Enter a number and I will determine its harmonic sum: 5
The harmonic sum of 5 is 2.283333333333333
```

It's worth noting that the Harmonic sum of 5 is computed by the summation:

$$H(5) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} = 2.283333333333333$$

The name of the method which computes and returns this harmonic sum should be named ***`harmonicSum`*** and this method should compute the sum *recursively* (5 points).

3. Design and implement an ADT named **`CreditCard`** that represents a credit card. The data of the ADT should include Java variables for the customer name, the account number, the next due date, the reward points, and the account balance. The initialization operation should set the data to client-supplied values. Include operations for a credit card charge, a cash advance, a payment, the addition of interest to the balance, and the display of the statistics of the account. Be sure to include a main class which creates an object from your **`CreditCard`** class. Do not worry too much about the correctness of the methods, **your method headers and whether they are invoked correctly from client code are what really matter in this exercise** (20 points).