Group HW4 (25 pts)

Everyone will submit one copy of this HW

(8 pts) Multiple Choices Questions: 2 pts each, choose one or more than one answers.

1. Analyze the following code:

```
public class Test {
        public static void main(String[] args){
                NClass nc = new NClass();
                nc.t = nc.t++;
        }
}
class NClass {
        int t;
        private NClass() {
        }
}
```

   a. The program has a compile error because the NClass class has a private constructor.
   b. The program does not compile because the parameter list of the main method is wrong.
   c. The program compiles, but has a runtime error because t has no initial value.
   d. The program compiles and runs fine.

2. MyArrayList is more efficient than MyLinkedList for the following operations:
   a. Insert/delete an element in the middle of the list.
   b. Insert/delete an element in the beginning of the list.
   c. Retrieve an element at given the index.

3. MyLinkedList is more efficient than MyArrayList for the following operations:
   a. Insert/delete an element in the middle of the list.
   b. Insert/delete an element in the beginning of the list.
   c. Retrieve an element at given the index.

4. Which of the following operations are supported by a list?
   a. Retrieve an element from this list.
   b. Insert a new element to this list.
   c. Delete an element from this list.
   d. Find how many elements are in this list.
   e. Find whether an element is in this list.

(7 pts) Coding Exercise Question: Remove Nth node from end of list

Given a LinkedList, create a method to remove the nth node from the end of list and return its head. (The minimum number of nodes in list is n.)

Example:

Input: 1 -> 2 -> 3 -> 4 -> 5 -> null, n = 2

Output: 1 -> 2 -> 3 -> 5 -> null

Coding: (Copy and Paste Source Code here.)

Testing: (Paste the screenshot of your result here.)

(10 pts) Coding Exercise Project: Validating Credit Cards

Problem Description:
Credit card numbers follow certain patterns. A credit card number must have between 13 and 16 digits. It must start with:

4 for Visa cards
5 for Master cards
37 for American Express cards
6 for Discover cards

In 1954, Hans Luhn of IBM proposed an algorithm for validating credit card numbers. The algorithm is useful to determine if a card number is entered correctly or if a credit card is scanned correctly by a scanner. Almost all credit card numbers are generated following this validity check, commonly known as the Luhn check or the Mod 10 check, which can be described as follows (for illustration, consider the card number 4388576018402626):

1. Double every second digit from right to left. If doubling of a digit results in a two-digit number, add up the two digits to get a single-digit number.
2 * 2 = 4
2 * 2 = 4

4 * 2 = 8
1 * 2 = 2
6 * 2 = 12 (1 + 2 = 3)
5 * 2 = 10 (1 + 0 = 1)
8 * 2 = 16 (1 + 6 = 7)
4 * 2 = 8

2. Now add all single-digit numbers from Step 1.
4 + 4 + 8 + 2 + 3 + 1 + 7 + 8 = 37

3. Add all digits in the odd places from right to left in the card number.
6 + 6 + 0 + 8 + 0 + 7 + 8 + 3 = 38

4. Sum the results from Step 2 and Step 3.
37 + 38 = 75

5. If the result from Step 4 is divisible by 10, the card number is valid; otherwise, it is invalid. For example, the number 4388576018402626 is invalid, but the number 4388576018410707 is valid.

Write a program that prompts the user to enter a credit card number as a long integer. Display whether the number is valid or invalid.

Here are sample runs of the program:

Sample 1:

```
Enter a credit card number as a long integer: 4246345689049834
4246345689049834 is invalid
```

Sample 2:

```
Enter a credit card number as a long integer: 4388576018410707
4388576018410707 is valid
```

Coding: (Copy and Paste Source Code here.)

Testing: (Paste the screenshot of your result here.)