CS526
Homework Assignment 4

Due: 2/25

This assignment has two parts. The first part is a practice of analyzing running times of small code segments. The second part is a practice of manipulating a singly linked list.

Part 1 (10 points)

Consider the following five methods:

```
public static int example1(int[] arr) {
  int n = arr.length, total = 0;
  for (int j=0; j < n; j++)
    total += arr[j];
  return total;
}

public static int example2(int[] arr) {
  int n = arr.length, total = 0;
  for (int j=0; j < n; j += 2)
    total += arr[j];
  return total;
}

public static int example3(int[] arr) {
  int n = arr.length, total = 0;
  for (int j=0; j < n; j++)
    for (int k=0; k <= j; k++)
      total += arr[j];
  return total;
}

public static int example4(int[] arr) {
  int n = arr.length, prefix = 0, total = 0;
  for (int j=0; j < n; j++) {
    prefix += arr[j];
    total += prefix;
  }
  return total;
}

public static int example5(int[] first, int[] second) { // assume equal-length arrays
  int n = first.length, count = 0;
  for (int i=0; i < n; i++) {
    int total = 0;
    for (int j=0; j < n; j++)
      for (int k=0; k <= j; k++)
        total += first[k];
    if (second[i] == total) count++;
  }
  return count;
}
```

Express the running time of each method using the *big-Oh* notation. You need to justify your answers. Your justification doesn't have to be formal (or mathematical) but must be logically correct. If you show only answers without justification, no points will be given even though your answers are correct.

<u>Part 2</u> (20 points)

For this part, you are required to implement a *remove* method that removes a node from a singly linked list of cars. The singly linked list you will use is *CarLinkedList*, which is modified from textbook's *SinglyLinkedList*. An incomplete code of *CarLinkedList.java* is posted on Blackboard. You must implement a remove method **within the *CarLinkedList* class**. The class file also includes a *main* method, which you can use to test your *remove* method. Note that the *CarLinkedList* class uses the *Car* class, which is also posted on Blackboard. A test input *car_info.txt* file is also posted on Blackboard.

The signature of the remove method must be:

```
public Car remove(String VIN)
```

A pseudocode of a remove method is given below:

```
Algorithm remove
Input: VIN; this is the VIN of the car to be removed
Output:
  -  Return the car object that was removed
  -  If the linked list is empty or there is no car with given VIN, return null

     // code

     If the list is empty, return null

     // traverse the list with two pointers
     // previous pointer follows the current pointer
     // initially both are pointing to the head node
    current = head of the linked list
    previous = current

    while (current is not null) {

        currentVIN = VIN of the car pointed to by current

        // the body of if statement is intentionally left empty
        // you must implement this part
        if (VIN is the same as currentVIN) { // we found the car
            // remove and return the car object from the list
            // you need to consider two cases:
            // Case 1: current is pointing to the head node of the list
            // Case 2: current is pointing to non-head node
        } // end of if

        advance current and previous pointers // to examine the next car
    } // end of while

    Return null
```

Note that the above pseudocode is a high-level pseudocode, which do not include some details, and you must translate it to a Java code. Also note that a part of the pseudocode is intentionally left empty. It is your responsibility to implement this part.

Documentation

No separate documentation is needed. However, you must include sufficient inline comments within your program.

Deliverables

You need to prepare two files. The first file contains the answers to Part 1 questions. Name this file *hw4_part1.EXT*, where *EXT* is an appropriate file extension, such as *pdf* or *docx*. The second file is a completed *CarLinkedList.java* file. Combine these two files, and additional files if any, into a single archive file and name it *LastName_FirstName_hw*4*.EXT*, where *EXT* is an appropriate file extension, such as *zip* or *rar*. Upload this file to Blackboard.

Grading

Part 1: For each wrong answer, 2 points will be deducted

Part 2: I will run your program with a different input file and try to remove 4 cars. Up to 3 points will be deducted for each wrong output.

Up to 4 points will be deducted if there are no sufficient inline comments.