MIDTERM EXAMINATION

Course: **CS162 – COMPUTER SCIENCE II**	
Time: **100** minutes	Term: **2** – Academic year: **2014-2015**
Lecturer(s): **Dr. Dinh Ba Tien**	
Student name:	Student ID:

(Notes: Closed book exam)

Question 1. What is a pointer? How to **allocate** and **de-allocate** memory for pointers?

Question 2. In a singly linked list with the 2 pointers, **pHead** and **pTail**, removing a node from **the beginning** or **the end**, which one is easier and faster? Please explain why.

Question 3. Given a singly linked list whose values are sorted in **descending order**, write a function to **insert a new node** with value K into the list and still keep it sorted

E.g: List: 134 → 82 → 76 → 65 → 12 → NULL

Insert 67: ==> List: 134 → 82 → 76 → **67** → 65 → 12 → NULL

Question 4. Assuming that we are maintaining a book store in which the list of books is structured by a **singly linked list** controlled by a **head** node and each book defined as below:

```
struct Book
{
      int  ID; // an unique ID of a book
      float price; //the price of the book
      char sTitle[100];
      int  yearPublished;
      int  stock; //number of copies of this book
      Book* pNext;
};
```

You are asked to write the following functions:

a. Remove a book from the list based on its ID. If there are more than 1 node with that ID, remove them all.

b. Print out the details of the books whose **IDs** are **duplicated** (the detail includes ID, price, title, and year. Print out **once for each ID**).

c. Assuming that all of the IDs of the books are kept sorted in **ascending order** and there is no duplicated ID in the list, merge the 2 book stores (i.e. 2 book lists) into 1 list to make sure the IDs are **still sorted** and there is **no duplicated ID**.

Notes: if 2 nodes from the lists have the same book ID, merge them by adding the stock levels of the two.

-- GOOD LUCK --