This Lab Exercise involves designing and implementing a class called **`IntList`**, which creates a linked list of integer values. The general "policy" of this linked list is to maintain the list in sorted order. Notice, however, that the APPEND command does not honor that policy: it appends the new node to the end of the list, regardless of the data value.

Due Date:

You must demonstrate the solution to this lab exercise to the instructor by **Wednesday, November 27**, in order to receive full credit for this work.

Programming Assignment

Create **IntList.h** and **IntList.cpp** as specified below. Also create a **Lab18.1.cpp** program. This program contains the "main()" function.

Class Specification File (IntList.h)

In the **IntList.h** file, declare a class named **`IntList`**. The class should declare a **`struct`** for a linked list of integers:

```
struct ListNode
{
     int value;
     struct ListNode *next;
}
```

The **`IntList`** class should contain a member variable that is a pointer to ListNode:

```
ListNode *head;     //  List head pointer
```

The **`IntList`** class should define a constructor, which takes no arguments and sets the **`head`** member variable to the NULL pointer value.

The **`IntList`** class should also declare function prototypes for the following member functions:

```
~IntList();
void appendNode(int);      // Append a new node at the end of the list.
void insertNode(int);     // Insert a new node into the list, maintaining a sorted order.
void deleteNode(int);      // remove a node from the list.
void print();              // output the contents of the list to the screen
```

Class Implementation File (IntList.cpp)

The **`IntList.cpp`** file should contain code for all of the functions declared in **`IntList.h`**.

Interactive "Main" Program

In the main function, implement an interactive command-loop similar to those we have used in other labs. (Feel free to copy/paste from other labs if you wish.)

Interactive Commands

The program must support the following commands:

- **a** APPEND a new node at the end of the list
- **d** DELETE a node from the list
- **i** INSERT a node into the list, maintaining the sorted order.
- **p** PRINT the contents of the list on the console. Format the output as shown in the **Sample Output** section of this document. That is, the output should include the memory address of each node, the value field, and the value of the next pointer. (We should be able to look at the console output and follow the links from one node to the next in the list.)
- **h** HELP text
- **q** QUIT (end the program)

The APPEND, DELETE, and INSERT commands should prompt the user to enter an integer value.

Sample Output

In the sample session shown below, the text that the user types is shown in **bold** font. In actuality, all text appears in the same font.

Sample Output

```
Command: a
Enter number to append to the list: 43

Command: i
Enter number to insert into the list: 978

Command: i
Enter number to insert into the list: 3

Command: i
Enter number to insert into the list: 55

Command: p
head=0063D268
0063D268:  value=   3 next=  0063D1F8
0063D1F8:  value=  43 next=  0063D2A0
0063D2A0:  value=  55 next=  0063D230
0063D230:  value= 978 next=  00000000

Command: d
Enter number to delete from the list: 3
```

Sample Output

```
Command: p
head=0063D1F8
0063D1F8:  value=  43 next=  0063D2A0
0063D2A0:  value=  55 next=  0063D230
0063D230:  value= 978 next=  00000000

Command: a
Enter number to append to the list: 6

Command: p
head=0063D1F8
0063D1F8:  value=  43 next=  0063D2A0
0063D2A0:  value=  55 next=  0063D230
0063D230:  value= 978 next=  0063D268
0063D268:  value=   6 next=  00000000

Command: d
Enter number to delete from the list: 123
Data value 123 not found.

Command: p
head=0063D1F8
0063D1F8:  value=  43 next=  0063D2A0
0063D2A0:  value=  55 next=  0063D230
0063D230:  value= 978 next=  0063D268
0063D268:  value=   6 next=  00000000

Command: d
Enter number to delete from the list: 6

Command: p
head=0063D1F8
0063D1F8:  value=  43 next=  0063D2A0
0063D2A0:  value=  55 next=  0063D230
0063D230:  value= 978 next=  00000000

Command: q
Exitting program with status = 0
Press any key to continue . . .
```