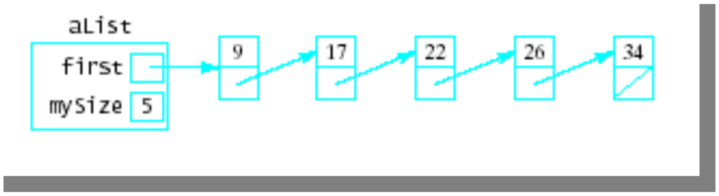
Hyo Lee
Week 1 Homework Submission
01/12/2016

CST 370
Homework (Linked Lists)

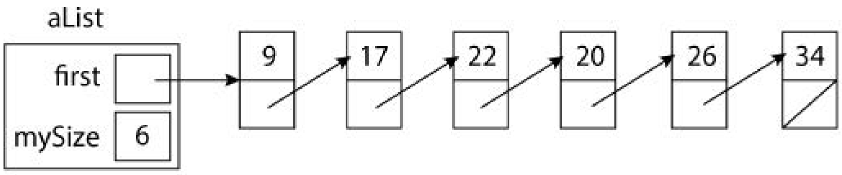
1. Suppose that you are given a linked list as shown below. You can read about **linked lists** from section 6.4 and 6.5 of the book. Source code describing the operation of a Linked List is available on iLearn (LinkedList.h, LinkedList.cpp and Sample_LinkedList_Tester.cpp). **(10 points)**

Assume that there is a function insert (as defined below) to add a node in the linked list. Read the insert function very carefully.

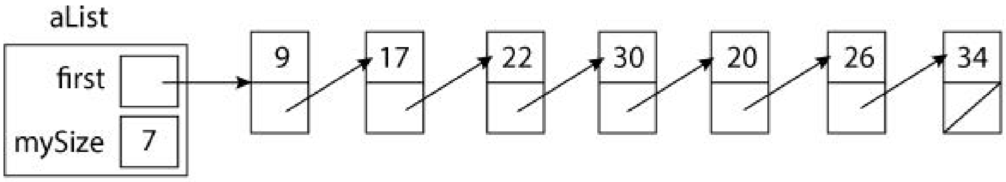
```
void List::insertnew(ElementType dataVal, int index)
{
      mySize++;
      Node * newPtr = new Node(dataVal);
      Node * predPtr = first;

      for(int i = 0; i <= index; i++)
    {
            predPtr = predPtr->next;
    }
      newPtr->next = predPtr->next;
      predPtr->next = newPtr;
}
```



(a) Draw the updated linked list after the execution of **aList.insertnew(20, 1)**.



(b) From the result of the above question (a), draw the updated linked list after the execution of **aList.insertnew(30, 1)**.



2. The following presents the **insertnew()** member function for a **static array-based list**. **(10 points)**

```
void List::insertnew (ElementType item, int pos)
{
   if (pos < 0 || pos > mySize) {
      cerr << "Illegal location: " << pos << "\n";
      return;
   }

   for(int i = mySize; i > pos; i--) {
      myArray[i] = myArray[i - 1];
   }

   myArray[pos] = item;
   mySize++;

   for (int i = 0; i < mySize; i++) {
      cout << myArray[i] << "  ";
   }
   cout << endl;
   return;
}
```

Assume that the following code is a part of a client program. Present the execution result. You can assume that mySize is initially 0. You can read about the **static array-based list** from section 6.2 of the book. Source code described in the book is available on the book's website (Figure 6.1).

```
List intList;

intList.insertnew(100, 0);
intList.insertnew(200, 2);
intList.insertnew(300, 0);
intList.insertnew(400, 2);
```

```
100
Illegal location: 2
300 100
300 100 400
```

3. Consider the Linked Lists files available on iLearn (LinkedList.h, LinkedList.cpp and Sample_LinkedList_Tester.cpp). Write a member function to find the sum of the values in a linked list **(10 points).**

```
// LinkedList.h
// Member function declaration
ElementType sum();

// LinkedList.cpp
// Member function definition
ElementType LinkedList::sum()
{
        ElementType listSum = 0;
        Node * ptr = first;

        while (ptr != 0)
        {
                listSum = listSum + ptr->data;
                ptr = ptr->next;
        }
        return listSum;
}
```