Chapter 3 – Linked Lists

4) Given two sorted lists, List A and List B, merge them together in to a new sorted List C. Remember, list is stl, List is ours.

List<int> merge(List<int> A, List<int> B)

{

List<int> C;

while (A.size() && B.size()>0)

{

If (A.front() > B.front())

{

C.push\_back(B.front());

B.pop\_front();

}

Else

{

C.push\_back(A.front());

A.pop\_front();

}

While (A.size()>0)

{

C.push\_back(A.front());

A.pop\_front();

}

While (B.size()>0)

{

C.push\_back(B.front());

B.pop\_front();

}

Return C;

}

10) Write a new method for list - bool List<>::IsEqual( const List <> &tOther ); // Exact same data, like as if = had been called.

Bool IsEqual(struct List\* ListA, struct List\* ListB)

{

Return false;

}

13) Write a new method for list - void List<>::Append( const List<> &tOther );// Add all of the data in Other to the end of me

void Append( struct List\* ListA, struct List\* List B )

{

Struct List\* ListC;

ListC = ListA;

While (ListC-> next != null)

ListC = ListC->next;

ListC->next = ListB;

}