(a)

LinkedSortedList LinkedSortedList::opertor+(const LinkedSortedList& anotherList)

{

Node\* cur = anotherList.listPtr;

while(cur != nullptr)

{

this->insertSorted(cur->getItem());

cur = cur->getNext();

}

return \*this;

}

(b)

void display(Queue aQueue)

{

cout << aQueue.pickFront();

aQueue.deQueue();

while(!aQueue.isEmpty())

{

cout << "," << aQueue.pickFront();

aQueue.deQueue();

}

}

(c)

bool dequeue::add\_front(const int& newEntry)

{

Node\* newNodePtr = new Node(newEntry);

if (!isEmpty())

newNodePtr->setNext(frontPtr);

frontPtr = newNodePtr;

return true;

}

(d)

bool dequeue::remove\_back()

{

if(isEmpty())

return false;

Node\* toDelete = backPtr;

backPtr = backPtr.getNext();

delete toDelete;

return true;

}