```java
package asgn6;

class DoubleNode
{
   String str;
   DoubleNode prev, next;

   DoubleNode(String str)
   {
      this.str = str;
      prev = null;
      next = null;
   }

   String getStr()
   {
      return str;
   }

   DoubleNode getNext()
   {
      return next;
   }

   void setNext(DoubleNode next)
   {
      this.next = next;
   }

   void setStr(String next)
   {
          this.str=next;
   }
   DoubleNode getPrev()
   {
      return prev;
   }

   void setPrev(DoubleNode prev)
   {
      this.prev = prev;
   }


}
```

```
package asgn6;


class DoublyLinkedList
{
   DoubleNode head, tail;

   DoublyLinkedList()
   {
      head = null;
      tail = null;
   }

   DoubleNode getHead()
   {
      return head;
   }

   DoubleNode getTail()
   {
      return tail;
   }

   DoubleNode getElement(int index)
   {
      int i = 0;
      DoubleNode cur = head;
      while (cur != null && i<index)
      {
         cur = cur.getNext();
         i++;
      }
      return cur;
   }

   DoubleNode search(String s)
   {
      DoubleNode cur = head;
      while (cur != null && !s.equals(cur.getStr()))
      {
         cur = cur.getNext();
      }
      return cur;
   }

   void addFirst(String s)
   {
      DoubleNode n = new DoubleNode(s);
      if (head == null)  // case that the list is empty
      {
         head = n;
         tail = n;
      }
      else
      {
```

```
      n.setNext(head);
      head.setPrev(n);
      head = n;
   }
}

void addLast(String s)
{
   DoubleNode n = new DoubleNode(s);
   if (head == null)  // case that the list is empty
   {
      head = n;
      tail = n;
   }
   else
   {
      n.setPrev(tail);
      tail.setNext(n);
      tail = n;
   }
}

void insert(DoubleNode elem, int index)
{
   if (index == 0)
   {
      // same code as addFirst
      if (head == null)  // case that the list is empty
      {
         head = elem;
         tail = elem;
      }
      else
      {
         elem.setNext(head);
         head.setPrev(elem);
         head = elem;
      }
   }
   else
   {
      DoubleNode prev = getElement(index-1);
      DoubleNode next = prev.getNext();
      elem.setNext(next);
      elem.setPrev(prev);
      prev.setNext(elem);
      if (next != null)
         next.setPrev(elem);
      else // if insert as last element
         tail = elem;
   }
}

void remove(DoubleNode elem)
{
   DoubleNode prev = elem.getPrev();
   DoubleNode next = elem.getNext();
   if (head == tail) // removing single element
   {
      if (head == elem)
      {
         head = null;
         tail = null;
      }
   }
   else if (prev == null) // removing head
   {
      head = next;
      head.setPrev(null);
   }
   else if (next == null) // removing tail
   {
      tail = prev;
      tail.setNext(null);
   }
   else
   {
      prev.setNext(next);
      next.setPrev(prev);
   }
}

void printList()
{
   System.out.print("The list:");
   DoubleNode cur = head;
   while (cur != null)
   {
      System.out.print(" " + cur.getStr());
      cur = cur.getNext();
   }
   System.out.println();
}

void printReverse()
{
   System.out.print("The reversed list:");
   DoubleNode cur = tail;
   while (cur != null)
   {
      System.out.print(" " + cur.getStr());
      cur = cur.getPrev();
   }
   System.out.println();
}

int getSize()
{
       int count=0;
       DoubleNode cur=head;
```

```
        while(cur !=null)
        {
            cur=cur.getNext();
            count++;
        }
        return count;
    }

    void sort()
    {
        DoubleNode cur=null;
        DoubleNode next=null;
        String temp=null;

        if (head==null)
            head=null;
        else
        {
            for(cur=head;cur.getNext() !=null;cur=cur.getNext())
            {
                for (next=cur.getNext();next !=null;next=next.getNext())
                {
                    if (cur.getStr().compareTo(next.getStr())>0)
                    {
                        temp=cur.getStr();
                        cur.setStr(next.getStr());
                        next.setStr(temp);
                    }
                }
            }
        }
    }
}
```

```
package asgn6;


class SortDoublyLinkedList
{
   public static void main(String[] args)
   {
      DoublyLinkedList list = new DoublyLinkedList();

      list.addFirst("Plum");
      list.addFirst("Grape");
      list.addFirst("Banana");
      list.addFirst("Pear");
      list.addFirst("Apple");
      list.addFirst("Kiwi");

      list.sort();


      list.printList();
      list.printReverse();
   }
}
```