```
class Book:
    def __init__(self,id,bookName,authorName,nextNode=None):
        self.id = id
        self.bookName = bookName
        self.authorName = authorName
        self.nextNode = nextNode

    def getId(self):
        return self.id

    def getBookName(self):
        return self.bookName

    def getAuthorName(self):
        return self.authorName

    def getNextNode(self):
        return self.nextNode

    def setNextNode(self,val):
        self.nextNode = val

class LinkedList:
    def __init__(self,head = None):
        self.head = head
        self.size = 0

    def getSize(self):
        return self.size

    def AddBookToFront(self,newBook):
        newBook.setNextNode(self.head)
        self.head = newBook
        self.size+=1

    def DisplayBook(self):
        curr = self.head
        while curr:
            print(curr.getId(),curr.getBookName(),curr.getAuthorName())
            curr = curr.getNextNode()

    def RemoveBookAtPosition(self,n):
        prev = None
        curr = self.head
        curPos = 0
        while curr:
            if curPos == n:
                if prev:
                    prev.setNextNode(curr.getNextNode())
                else:
```

```
                    self.head = curr.getNextNode()
                self.size = self.size - 1
                return True
            prev = curr
            curr = curr.getNextNode()
            curPos = curPos + 1
        return False

    def AddBookAtPosition(self,newBook,n):
        curPos = 1
        if n == 0:
            newBook.setNextNode(self.head)
            self.head = newBook
            self.size+=1
            return
        else:
            currentNode = self.head
            while currentNode.getNextNode() is not None:
                if curPos == n:
                    newBook.setNextNode(currentNode.getNextNode())
                    currentNode.setNextNode(newBook)
                    self.size+=1
                    return
                currentNode = currentNode.getNextNode()
                curPos = curPos + 1
            if curPos == n:
                newBook.setNextNode(None)
                currentNode.setNextNode(newBook)
                self.size+=1
            else:
                print("cannot add",newBook.getId(),newBook.getBookName(),"at that
position")

    def SortByAuthorName(self):
        for i in range(1,self.size):
            node1 = self.head
            node2 = node1.getNextNode()
            while node2 is not None:
                if node1.authorName > node2.authorName:
                    temp = node1.id
                    temp2 = node1.bookName
                    temp3 = node1.authorName

                    node1.id = node2.id
                    node1.bookName = node2.bookName
                    node1.authorName = node2.authorName

                    node2.id = temp
                    node2.bookName = temp2
                    node2.authorName = temp3
```

```
                node1 = node1.getNextNode()
                node2 = node2.getNextNode()

myLinkedList = LinkedList()
nodeA = Book("Name: Jim","Favorite Color: Blue","Birthday: 6/17/95,Favorite Music
Artist: Kanye West")
nodeB = Book("Name: Jerry","Favorite Color: Green","Birthday: 6/18/96,Favorite Music
Artist: Rod Stewart")
nodeC = Book("Name: Ted","Favorite Color: Red","Birthday: 6/19/97,Favorite Music
Artist: Willoe Nelson")
nodeD = Book("Name: Bill","Favorite Color: Yellow","Birthday: 6/20/98,Favorite Music
Artist: Kid Cudi")
nodeE = Book("Name: Henry","Favorite Color: Indigo","Birthday: 6/21/99,Favorite
Music Artist: Aerosmith")
nodeF = Book("Name: Tim","Favorite Color: Purple","Birthday: 6/22/00,Favorite Music
Artist: Will Smith")
myLinkedList.AddBookToFront(nodeA)
myLinkedList.AddBookToFront(nodeB)
myLinkedList.AddBookToFront(nodeC)
myLinkedList.AddBookAtPosition(nodeD,1)
myLinkedList.AddBookAtPosition(nodeE,1)
myLinkedList.AddBookAtPosition(nodeF,1)
myLinkedList.RemoveBookAtPosition(2)
myLinkedList.RemoveBookAtPosition(2)
myLinkedList.DisplayBook()
myLinkedList.SortByAuthorName()
print(myLinkedList.getSize())
myLinkedList.DisplayBook()

class LinkedList:
    def __init__(self,head = None):
        self.head = head
        self.size = 0

    def getSize(self):
        return self.size

    def AddBookToFront(self,newBook):
        newBook.setNextNode(self.head)
        self.head = newBook
        self.size+=1

    def DisplayBook(self):
        curr = self.head
        while curr:
            print(curr.getId(),curr.getBookName(),curr.getAuthorName())
            curr = curr.getNextNode()

    def RemoveBookAtPosition(self,n):
        prev = None
```

```
        curr = self.head
        curPos = 0
        while curr:
            if curPos == n:
                if prev:
                    prev.setNextNode(curr.getNextNode())
                else:
                    self.head = curr.getNextNode()
                self.size = self.size - 1
                return True
            prev = curr
            curr = curr.getNextNode()
            curPos = curPos + 1
        return False

    def AddBookAtPosition(self,newBook,n):
        curPos = 1
        if n == 0:
            newBook.setNextNode(self.head)
            self.head = newBook
            self.size+=1
            return
        else:
            currentNode = self.head
            while currentNode.getNextNode() is not None:
                if curPos == n:
                    newBook.setNextNode(currentNode.getNextNode())
                    currentNode.setNextNode(newBook)
                    self.size+=1
                    return
                currentNode = currentNode.getNextNode()
                curPos = curPos + 1
            if curPos == n:
                newBook.setNextNode(None)
                currentNode.setNextNode(newBook)
                self.size+=1
            else:
                print("cannot add",newBook.getId(),newBook.getBookName(),"at that
position")

    def SortByAuthorName(self):
        for i in range(1,self.size):
            node1 = self.head
            node2 = node1.getNextNode()
            while node2 is not None:
                if node1.authorName > node2.authorName:
                    temp = node1.id
                    temp2 = node1.bookName
                    temp3 = node1.authorName
```

```
                    node1.id = node2.id
                    node1.bookName = node2.bookName
                    node1.authorName = node2.authorName

                    node2.id = temp
                    node2.bookName = temp2
                    node2.authorName = temp3
                node1 = node1.getNextNode()
                node2 = node2.getNextNode()

myLinkedList = LinkedList()
nodeA = Book("Title: The Land Before,","Author:Wellington Higgenbottom,","Genre:
Scifi, Length: 10 pages",)
nodeB = Book("Title: Woopty Doo,","Author: Geaorge Bush,","Genre: Childrens, Length:
1 page",)
nodeC = Book("Title: Doopty Who,","Author: Joe Biden,","Genre: Pre-teen, Length: 2
pages",)
nodeD = Book("Title: Game Over,","Author: Hillary Clinton,","Genre: Fiction, Length:
5 pages",)
nodeE = Book("Title: Game Start,","Author: Clint Eastwood,","Genre: Western, Length:
1000 pages",)
nodeF = Book("Title: The End,","Author: Nicholas Cage,","Genre: Auto-Biography,
Length: 12 pages",)
myLinkedList.AddBookToFront(nodeA)
myLinkedList.AddBookToFront(nodeB)
myLinkedList.AddBookToFront(nodeC)
myLinkedList.AddBookAtPosition(nodeD,1)
myLinkedList.AddBookAtPosition(nodeE,1)
myLinkedList.AddBookAtPosition(nodeF,1)
myLinkedList.RemoveBookAtPosition(2)
myLinkedList.RemoveBookAtPosition(2)
myLinkedList.DisplayBook()
myLinkedList.SortByAuthorName()
print(myLinkedList.getSize())
myLinkedList.DisplayBook()
```