```c
/* BY SUBMITTING THIS FILE TO CARMEN, I CERTIFY THAT I HAVE PERFORMED ALL OF
THE WORK TO CREATE THIS FILE AND/OR DETERMINE THE ANSWERS FOUND WITHIN
THIS FILE MYSELF WITH NO ASSISSTANCE FROM ANY PERSON (OTHER THAN THE
INSTRUCTOR OR GRADERS OF THIS COURSE) AND I HAVE STRICTLY ADHERED TO THE
TENURES OF THE OHIO STATE UNIVERSITY'S ACADEMIC INTEGRITY POLICY.
*/
#include "lab4.h"
Node* insertNode(Node *list_head, Node *newNodePtr) {
        Node *priorNode;
        Node *traversePtr;
        /*If list is empty, make head point to the inserted node*/
        if (list_head == NULL) {
                list_head = newNodePtr;
                newNodePtr->next = NULL;
        }
        /*If node needs to be inserted in the first node, the current address in the
head must be copied to the next member in the newNodePtr structure and the value must
be changed to the address of newNodePtr*/
        else if (newNodePtr->student.student_ID <= list_head->student.student_ID) {
                newNodePtr->next = list_head;
                list_head = newNodePtr;
        }
        /*Non-exception case*/
        else {
                priorNode = list_head;
                traversePtr = priorNode->next;
                /*traverse the list to search for where we insert the node*/
                while (traversePtr != NULL && newNodePtr->student.student_ID >=
traversePtr->student.student_ID) {
                        priorNode = priorNode->next;
                        traversePtr = traversePtr->next;
                }
                /*check that the traversePtr is not NULL before accessing members of
the node to which it points.
                If it's NULL, we reached the end of the list and insert it there*/
                if (traversePtr == NULL) {
                        newNodePtr->next = NULL;
                        priorNode->next = newNodePtr;
                }
                else {
                        newNodePtr->next = traversePtr;
                        priorNode->next = newNodePtr;
                }
        }
        return list_head;
}
```