323.25 Exam1 Q2: LinkedList and Reversed LinkedList

Student: Jingshi Liu

Due Date: 3/17/2022

Source Code:

```
//
//  main.cpp
//  Exam1_coding_Q2
//
//  Created by Jingshi Liu on 3/17/22.
//

#include <iostream>
#include <fstream>
using namespace std;

class Node{
public:
    int data;
    Node* next;

    Node(int number){
        data = number;
        next = NULL;
    }

    string printNode(){
        string answer = "";
        answer += "(" + to_string(data) + ", ";
        if(next == NULL){
            answer += "NULL) -> ";
        }else{
            answer += to_string(next->data) + ") -> ";
        }
        return answer;
    }
};

class LinkedList{
public:
    Node* listHead;
```

```
LinkedList(){
    listHead = new Node(-99999);
}

Node* findSpot(Node* newNode){
    int newData = newNode->data;
    Node* spot = listHead;

    while(spot->next != NULL && spot->next->data < newData){
        spot = spot->next;
    }

    return spot;
}

void insertOneNode(Node* spot, Node* newNode){
    newNode->next = spot->next;
    spot->next = newNode;
}

void buildList(ifstream& inFile){
    int data;
    Node* newNode;
    Node* spot;

    while(inFile >> data){
        newNode = new Node(data);
        spot = findSpot(newNode);
        insertOneNode(spot, newNode);
    }
}

void reverseLinkedList(){
    Node* head = listHead;
    Node* last = listHead->next;
    Node* spot;

    if(last != NULL){
        while(last->next != NULL){
            spot = last->next;
            last->next = spot->next;
            insertOneNode(head, spot);
        }
    }
}
```

```
    void printList(ofstream& outFile){
        Node* head = listHead;
        outFile << "ListHead -> ";
        string nodeInfo;

        while(head != NULL){
            nodeInfo = head->printNode();
            outFile << nodeInfo;
            head = head->next;
        }

        outFile << "NULL";
    }

};

int main(int argc, const char * argv[]) {
    ifstream inFile;
    ofstream outFile;

    inFile.open(argv[1]);
    outFile.open(argv[2]);

    LinkedList* linkedList = new LinkedList();

    outFile << "Linked-List\n";
    linkedList->buildList(inFile);
    linkedList->printList(outFile);

    outFile << "\n\n\nReversed Linked-List\n";
    linkedList->reverseLinkedList();
    linkedList->printList(outFile);

    outFile.close();
    inFile.close();

    return 0;
}
```

OutFile:

Linked-List
ListHead -> (-99999, 5) -> (5, 8) -> (8, 8) -> (8, 9) -> (9, 10) -> (10, 12) -> (12, 14) -> (14, 16) -> (16, 18) -> (18, 22) -> (22, 29) -> (29, 36) -> (36, 55) -> (55, 66) -> (66, 77) -> (77, 88) -> (88, 91) -> (91, 91) -> (91, 99) -> (99, 133) -> (133, 316) -> (316, 322) -> (322, 361) -> (361, 538) -> (538, 637) -> (637, 702) -> (702, 999) -> (999, NULL) -> NULL

Reversed Linked-List
ListHead -> (-99999, 999) -> (999, 702) -> (702, 637) -> (637, 538) -> (538, 361) -> (361, 322) -> (322, 316) -> (316, 133) -> (133, 99) -> (99, 91) -> (91, 91) -> (91, 88) -> (88, 77) -> (77, 66) -> (66, 55) -> (55, 36) -> (36, 29) -> (29, 22) -> (22, 18) -> (18, 16) -> (16, 14) -> (14, 12) -> (12, 10) -> (10, 9) -> (9, 8) -> (8, 8) -> (8, 5) -> (5, NULL) -> NULL