A picture containing text

Description automatically generated



Lab Journal: 7

Date: 28 / 10 / 24

Student : Abdul Rafay

Enrollment : 01-131232-004

Department of Software Engineering

Bahria University, Islamabad

Data Structures & Algorithms Lab (Spring-2024)

Teacher: RAHEELA AMBRIN

**Comments:**

**Signature**

### Code:

#### List.h

#pragma once  
#include<iostream>  
  
typedef struct Node\* Nodeptr;  
  
struct Node  
{  
 int registrationNum;  
 int cc;  
 Nodeptr next;  
 Nodeptr prev;  
};  
  
class List {  
 Nodeptr head = nullptr;  
public:  
 void sort(Nodeptr car){  
 if (isEmpty()) {  
 Nodeptr newNode = car;  
 newNode->next = head;  
 newNode->prev = nullptr;  
 head = newNode;  
 return;  
 }  
 else if (head->next == nullptr) {  
 if (head->cc << car->cc) {  
 Nodeptr newNode = car;  
 newNode->next = head;  
 head->prev = newNode;  
 newNode->prev = nullptr;  
 head = newNode;  
 return;  
 }  
 else {  
 Nodeptr newNode = car;  
 newNode->prev = head;  
 head->next = newNode;  
 newNode->next = nullptr;  
 head = newNode;  
 return;  
 }  
 }  
 else {  
 for (Nodeptr p = head; p != nullptr; p = p->next) {  
  
 if (p->cc < car->cc) {  
 Nodeptr newNode = car;  
 newNode->next = p;  
 newNode->prev = p->prev;  
 if (p->prev != nullptr) {  
 p->prev->next = newNode;  
 }  
 p->prev = newNode;  
 return;  
 }  
 }  
 }  
 }  
  
 void filter(std::vector<Nodeptr> cars) {  
 for (Nodeptr i : cars) {  
 if (i->registrationNum % 2 == 0) {  
 sort(i);  
 }  
 else {  
 std::cout << "\nCar with Reg Num: " << i->registrationNum << " not ready for departure.\n";  
 }  
 }  
 }  
  
 bool isEmpty() {  
 return head == nullptr;  
 }  
  
 void display() {  
 if (!(isEmpty())) {  
 Nodeptr lastNode = nullptr;  
 for (Nodeptr p = head; p != nullptr; p = p->next) {  
 if (p->next == nullptr) {  
 lastNode = p;  
 }  
 }  
 int i = 1;  
 for (Nodeptr q = lastNode; q != nullptr; q = q->prev) {  
 if (q != nullptr) {  
 std::cout << "Car #" << i << ": \n";  
 std::cout << "Reg No. : " << q->registrationNum << "\n";  
 std::cout << "Engine Capacity: " << q->cc << "\n";  
 std::cout << "-----------------------------\n";  
 i++;  
 }  
 }  
 }  
 else {  
 std::cerr << "List is Empty";  
 }  
 }  
  
 void remove() {  
 if (!(isEmpty())) {  
 Nodeptr lastNode = nullptr;  
 for (Nodeptr p = head; p != nullptr; p = p->next) {  
 if (p->next == nullptr) {  
 lastNode = p;  
 }  
 }  
  
 std::cout << "\nCar Removed: \n";  
 std::cout << "Reg No. : " << lastNode->registrationNum << "\n";  
 std::cout << "Engine Capacity: " << lastNode->cc << "\n";  
 std::cout << "-----------------------------\n";  
 if (lastNode->prev != nullptr) {  
 lastNode->prev->next = nullptr;  
 }  
 else {  
 head = nullptr;  
 }  
 delete lastNode;  
 }  
 else {  
 std::cerr << "List is Empty";  
 }  
 }  
};

#### Main

#include <iostream>  
#include <vector>  
#include "list.h"  
using namespace std;  
  
vector<Nodeptr> userInput() {  
 vector<Nodeptr> cars;  
 cout << "How many Car do you want to register?\n=> ";  
 int carNum;  
 cin >> carNum;  
 for (int i = 0; i < carNum; i++) {  
 Nodeptr newNode = new Node;  
 int ccInput;  
 cout << "Car " << i + 1 << ": \n";  
 cout << "Enter Car Engine Capacity (cc): ";  
 cin >> ccInput;  
 newNode->cc = ccInput;  
 int regInput;  
 cout << "Enter Car Reg Num (4 - Digit): ";  
 cin >> regInput;  
 newNode->registrationNum = regInput;  
 cars.push\_back(newNode);  
 cout << endl;  
 }  
 return cars;  
}  
  
int main() {  
 List list;  
 cout << "Welcome to Car List :)" << endl;  
 list.filter(userInput());  
 int input;  
 do {  
 cout << "1. Remove Low CC Car ?\n";  
 cout << "2. Display Car List\n";  
 cout << "3. Exit\n=> ";  
 cin >> input;  
 switch (input) {  
 case 1:  
 list.remove();  
 break;  
 case 2:  
 list.display();  
 break;  
 case 3:  
 cout << "Exiting....";  
 break;  
 default:  
 cout << "Enter Correct Input";  
 cin.clear();  
 break;  
 }  
 } while (input != 3);  
 return 0;  
}