// C Code For Mentioned Problem //

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>


//structure named as node is created for car of the train
struct node
{
    int passengers;
    //character array of size 100 is declared for storing name of the car
    char name[100];
    struct node *next;
};



// createnode() function is defined which creates a new node and initializes it with given values to the function
struct node* createnode(int d,char name[])
{
    //new node is created using malloc() function and address is stored in newnode pointer
    struct node *newnode=(struct node *)malloc(sizeof(struct node));
    //given values are initialized to the function
    newnode->passengers=d;
    //strcpy() fucntion is used for assigning string
    strcpy(newnode->name,name);
    newnode->next=NULL;

    //returned address of newly created node
    return newnode;
}



 // display()

//display() method is defined which prints the information of each car in the list
void display(struct node *head)
{
    //if list is empty
    if(head==NULL)
    {
        printf("\nList is Empty\n");
        return;
    }

    //created temp pointer and stored address of head node
    struct node *temp=head;

    //traversed till temp is not equal to null
    while(temp!=NULL)
    {
        //printed passengers and name from temp node
        printf("\n%d %s",temp->passengers,temp->name);
        //updated temp incrementing it by 1 position
        temp=temp->next;
    }

}
//display method ended




// length()
//length() function is defined which returns the number of nodes present in the list
int length(struct node *head)
{
    //declared a counter c and initialized it with 0
    int c=0;
    //declared temp pointer and assigned head address to it
    struct node *temp=head;
    //traversed till temp is not equal to null
    while(temp!=NULL)
    {
        //increment c by 1
        c++;
        //update temp incrementing by one position
        temp=temp->next;
    }
    //returned counter c
    return c;
}
//length() method ended



//addCar()
// addCar() method is defined which adds a new node in the ens of the list
void addCar(struct node **head,int d,char name[])
{
    //newnode and temp pointers are declared
    struct node *newnode,*temp;

    //new node is created using createnode() function and returned address is stored in newnode pointer
    newnode =createnode(d,name);

    //if list is empty then add newnode as first node of the list
    if(*head==NULL)
    {
        *head=newnode;
        return;
    }

    //*head is assigned to temp pointer
    temp=*head;
    //traversed while next of temp is not equal to null
    //( traversed till last node )
    while(temp->next!=NULL)
        temp=temp->next;

    //added new node at the end of list
    temp->next=newnode;
}
//addCar() function ended
```

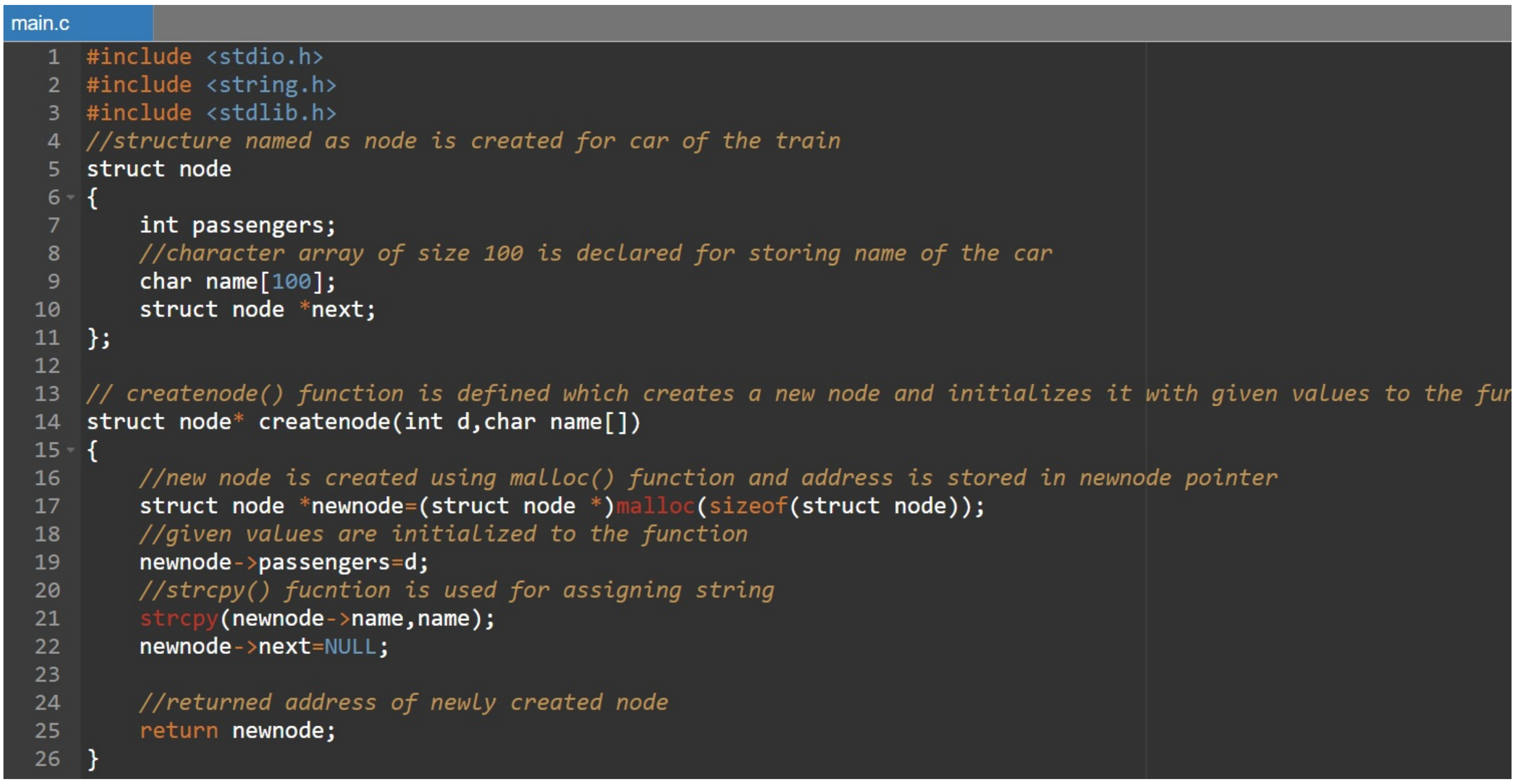
// removeCars()

```c
//removeCars() function is defined below
//it removes nodes where number of passengers are 0
void removeCars(struct node **head)
{
    //two pointers named as prev and temp are declared and initialized with *head
    //prev will point one node before the temp in the list
    struct node *prev,*temp;
    temp = *head;
    prev = *head;

    //traversed till temp is not equal to NULL
    while(temp!=NULL)
    {
        //if passengers in temp node is equal to 0
        if(temp->passengers == 0 )
        {
            //checked if temp is first node or not
            if(temp == *head)
            {
                printf("\n%d %s removed ",temp->passengers,temp->name);
                //if it is first node then make second node as head node
                *head = temp->next;
                //free memory for temp node using free() function
                free(temp);
                //re initialized temp and prev pointer
                temp = *head;
                prev = *head;

            }

            //if temp is not the first node then
            else
            {
                //remove temp node by changing the address field of the prev node
                printf("\n%d %s removed ",temp->passengers,temp->name);
                prev->next = temp->next;
                free(temp);
                temp = prev->next;

            }

        }

        //if passengers in temp node is not equal to 0
        else
        {
            //increment prev
            prev = temp;
            //increment temp
            temp = temp->next;
        }

    }

}
//removeCars() function ended
```

//main()

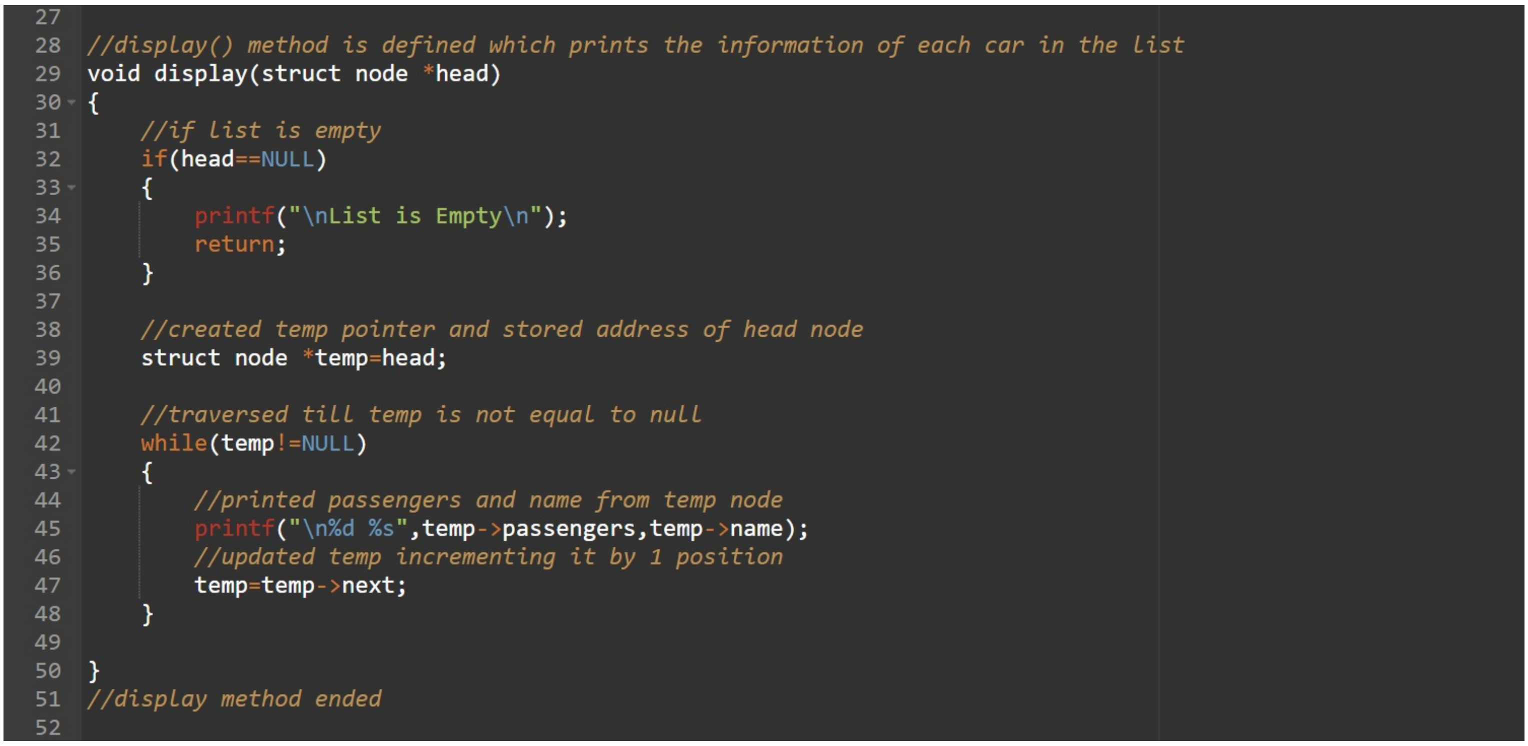
```c
//main() function is defined below
int main()
{
    //head node is created and initialized to NULL
    struct node *head=NULL;

    //cars are added in the list using addCar() fucntion
    addCar(&head,83,"B1");
    addCar(&head,72,"B2");
    addCar(&head,0,"B3");
    addCar(&head,69,"B4");
    addCar(&head,0,"B5");

    printf("Cars in the list:");
    //display() fucntion is called
    display(head);

    //removeCars() fucntion is called
    printf("\nremoveCars() fucntion is called ");
    removeCars(&head);
    //after removing empty cars ,display() fucntion is called
    printf("\nAfter removing empty cars:");
    display(head);
    //length() function is called and returned length of list is stored in l variable
    int l = length(head);
    //length of the list is printed
    printf("\nNumber of cars left in Train : %d\n",l);

    return 0;
}
//main() method ended
```

Editor Window:

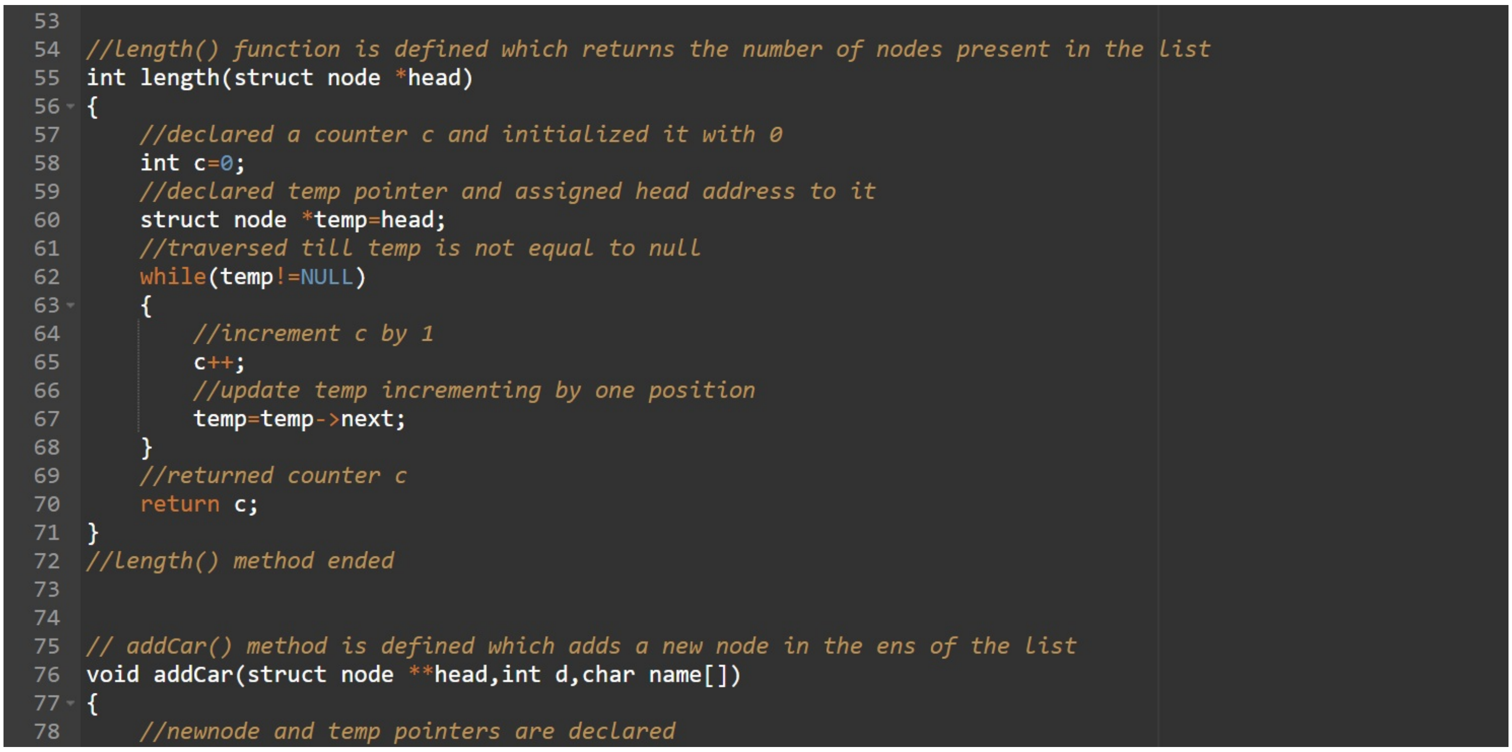


main.c

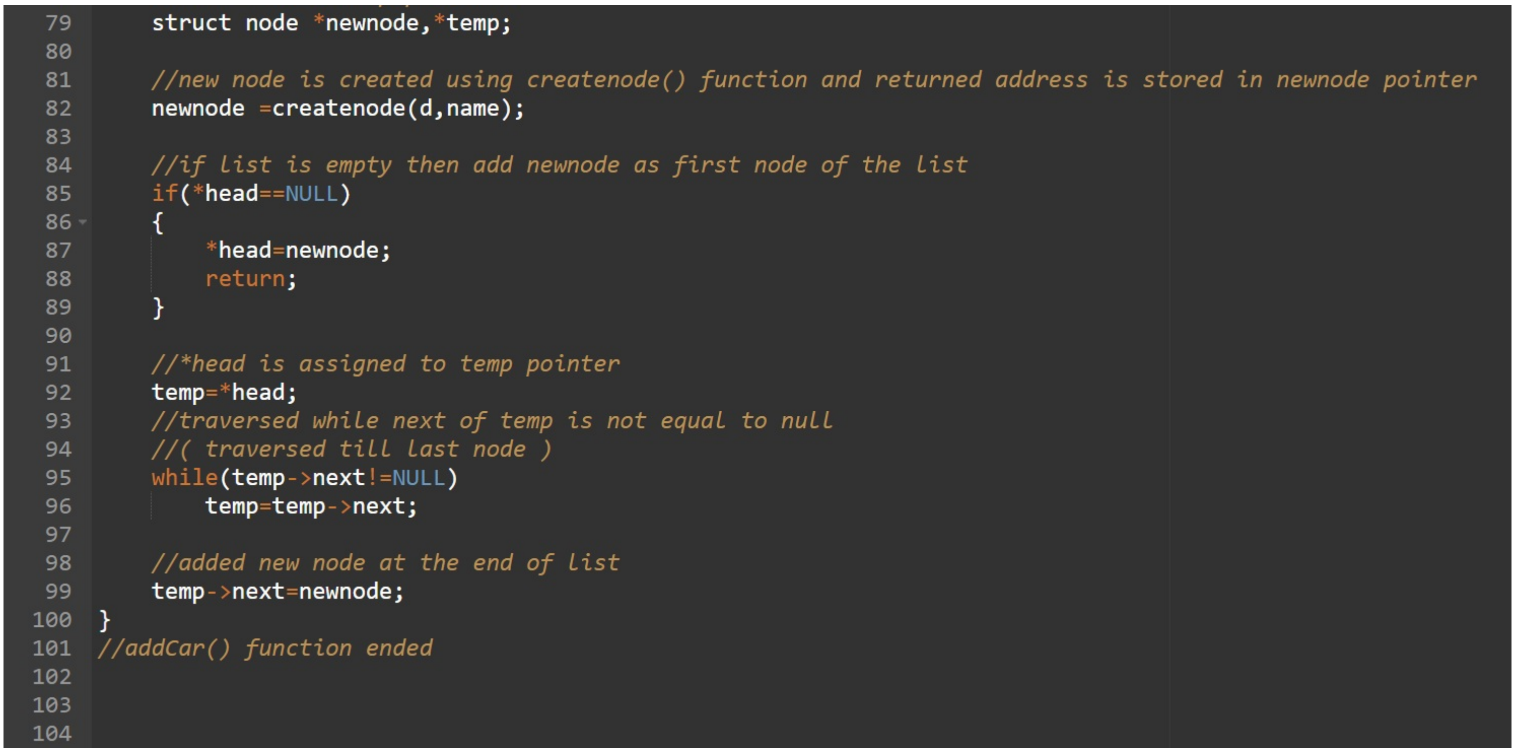
```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
//structure named as node is created for car of the train
struct node
{
    int passengers;
    //character array of size 100 is declared for storing name of the car
    char name[100];
    struct node *next;
};

// createnode() function is defined which creates a new node and initializes it with given values to the fur
struct node* createnode(int d,char name[])
{
    //new node is created using malloc() function and address is stored in newnode pointer
    struct node *newnode=(struct node *)malloc(sizeof(struct node));
    //given values are initialized to the function
    newnode->passengers=d;
    //strcpy() fucntion is used for assigning string
    strcpy(newnode->name,name);
    newnode->next=NULL;

    //returned address of newly created node
    return newnode;
}
```



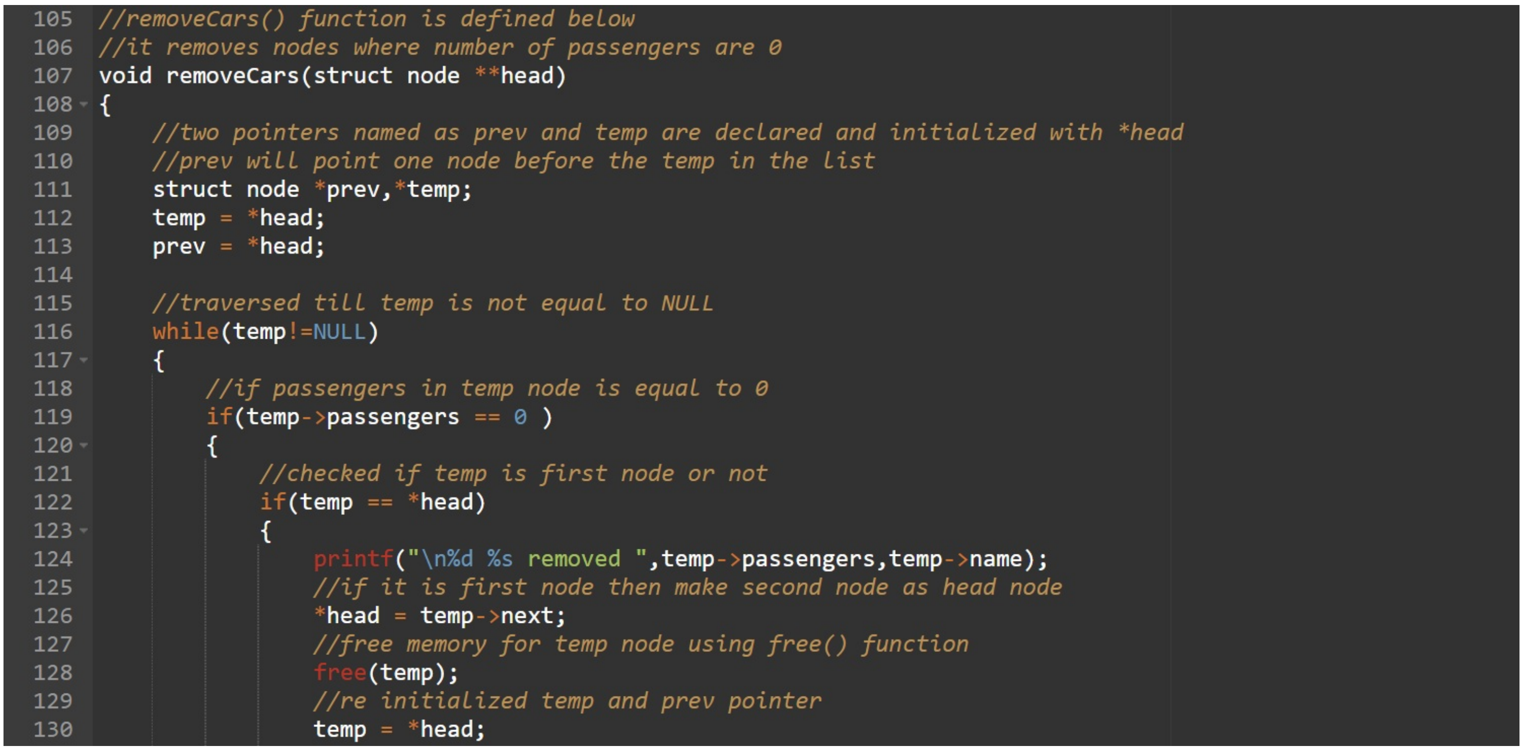
```c

//display() method is defined which prints the information of each car in the list
void display(struct node *head)
{
    //if list is empty
    if(head==NULL)
    {
        printf("\nList is Empty\n");
        return;
    }

    //created temp pointer and stored address of head node
    struct node *temp=head;

    //traversed till temp is not equal to null
    while(temp!=NULL)
    {
        //printed passengers and name from temp node
        printf("\n%d %s",temp->passengers,temp->name);
        //updated temp incrementing it by 1 position
        temp=temp->next;
    }

}
//display method ended

```



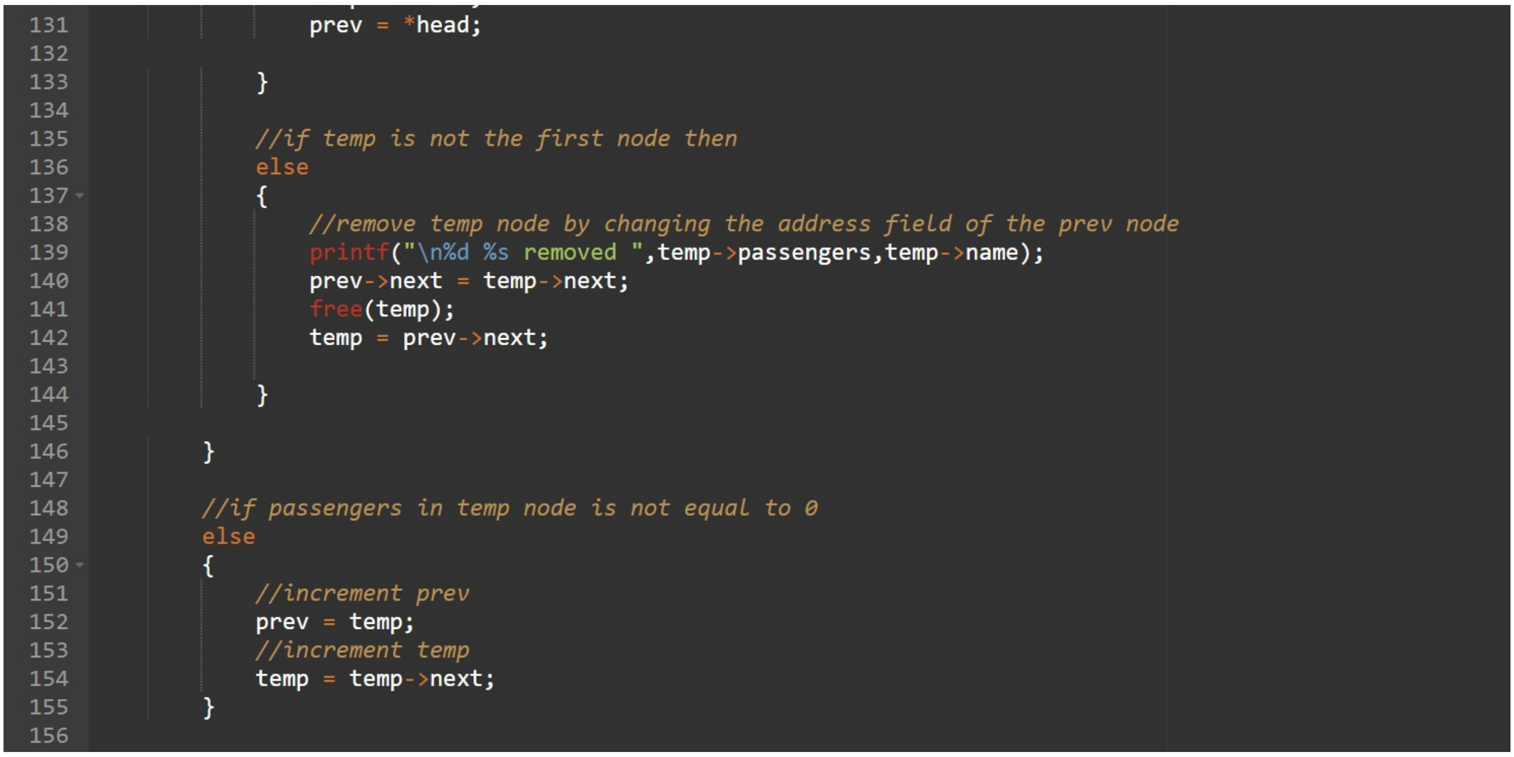
```c

//length() function is defined which returns the number of nodes present in the list
int length(struct node *head)
{
    //declared a counter c and initialized it with 0
    int c=0;
    //declared temp pointer and assigned head address to it
    struct node *temp=head;
    //traversed till temp is not equal to null
    while(temp!=NULL)
    {
        //increment c by 1
        c++;
        //update temp incrementing by one position
        temp=temp->next;
    }
    //returned counter c
    return c;
}
//length() method ended


// addCar() method is defined which adds a new node in the ens of the list
void addCar(struct node **head,int d,char name[])
{
    //newnode and temp pointers are declared
```



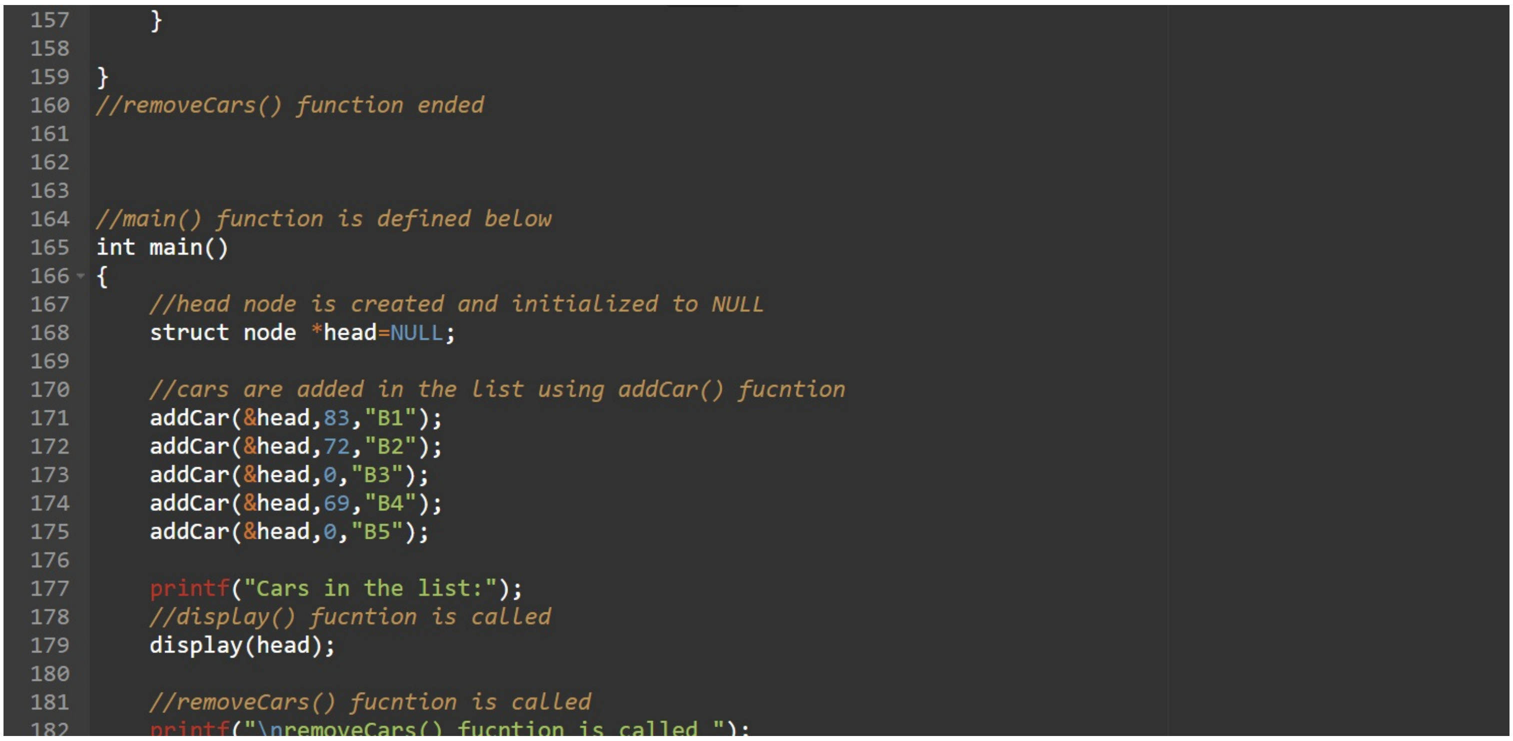
```c
    struct node *newnode,*temp;

    //new node is created using createnode() function and returned address is stored in newnode pointer
    newnode =createnode(d,name);

    //if list is empty then add newnode as first node of the list
    if(*head==NULL)
    {
        *head=newnode;
        return;
    }

    //*head is assigned to temp pointer
    temp=*head;
    //traversed while next of temp is not equal to null
    //( traversed till last node )
    while(temp->next!=NULL)
        temp=temp->next;

    //added new node at the end of list
    temp->next=newnode;
}
//addCar() function ended



```



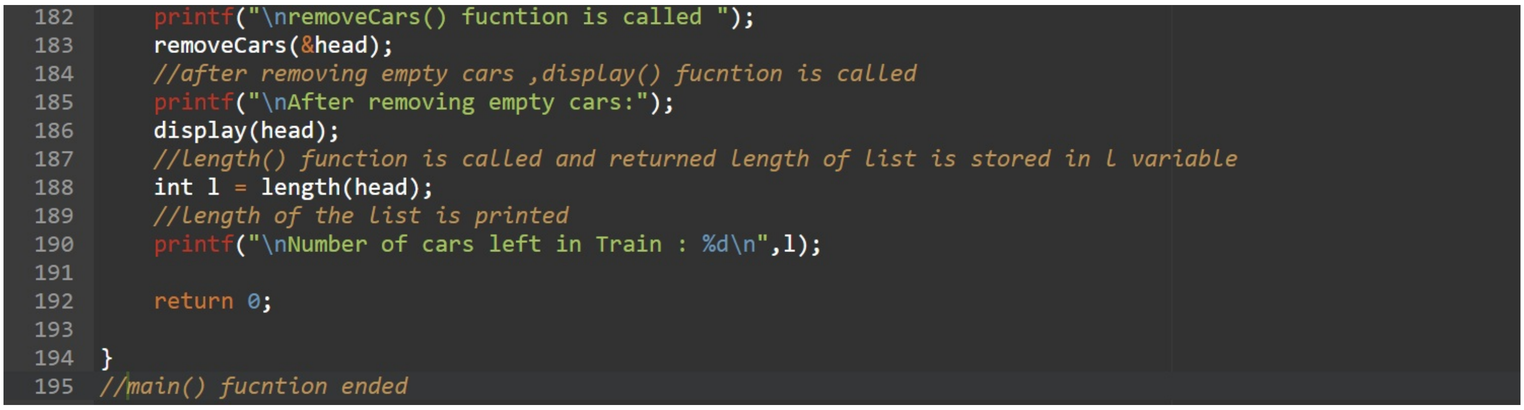
```c
//removeCars() function is defined below
//it removes nodes where number of passengers are 0
void removeCars(struct node **head)
{
    //two pointers named as prev and temp are declared and initialized with *head
    //prev will point one node before the temp in the list
    struct node *prev,*temp;
    temp = *head;
    prev = *head;

    //traversed till temp is not equal to NULL
    while(temp!=NULL)
    {
        //if passengers in temp node is equal to 0
        if(temp->passengers == 0 )
        {
            //checked if temp is first node or not
            if(temp == *head)
            {
                printf("\n%d %s removed ",temp->passengers,temp->name);
                //if it is first node then make second node as head node
                *head = temp->next;
                //free memory for temp node using free() function
                free(temp);
                //re initialized temp and prev pointer
                temp = *head;
```



```c
                prev = *head;

            }

            //if temp is not the first node then
            else
            {
                //remove temp node by changing the address field of the prev node
                printf("\n%d %s removed ",temp->passengers,temp->name);
                prev->next = temp->next;
                free(temp);
                temp = prev->next;

            }

        }

        //if passengers in temp node is not equal to 0
        else
        {
            //increment prev
            prev = temp;
            //increment temp
            temp = temp->next;
        }

```

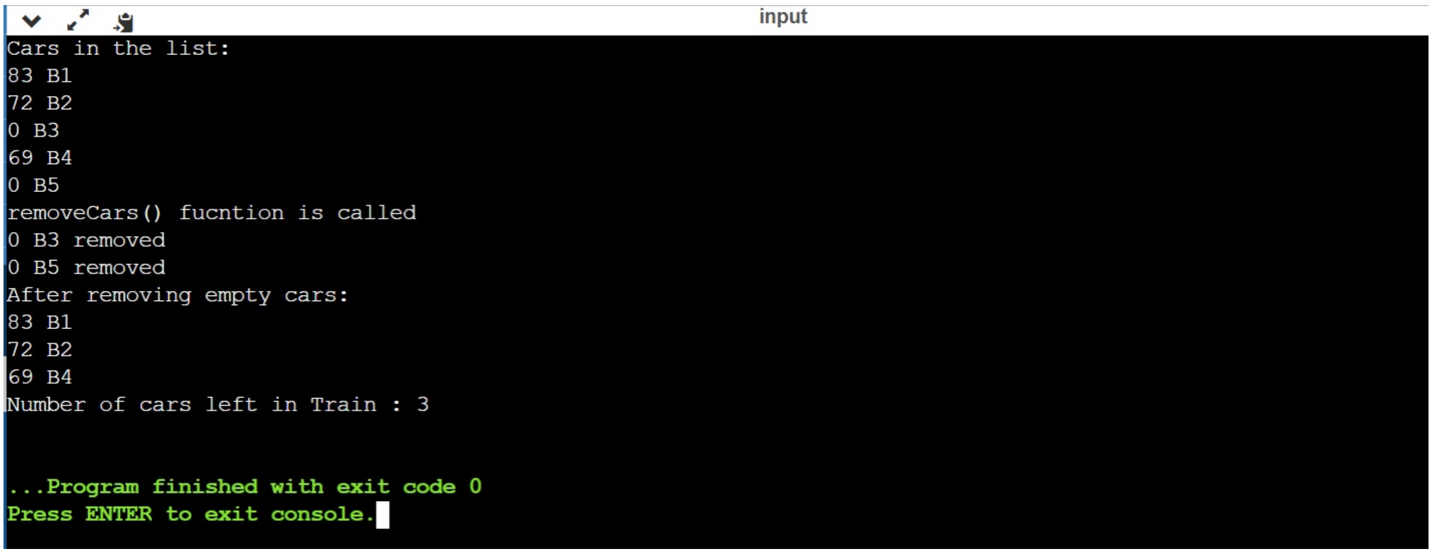


```c
    }

}
//removeCars() function ended



//main() function is defined below
int main()
{
    //head node is created and initialized to NULL
    struct node *head=NULL;

    //cars are added in the list using addCar() fucntion
    addCar(&head,83,"B1");
    addCar(&head,72,"B2");
    addCar(&head,0,"B3");
    addCar(&head,69,"B4");
    addCar(&head,0,"B5");

    printf("Cars in the list:");
    //display() fucntion is called
    display(head);

    //removeCars() fucntion is called
    printf("\nremoveCars() fucntion is called ");
```



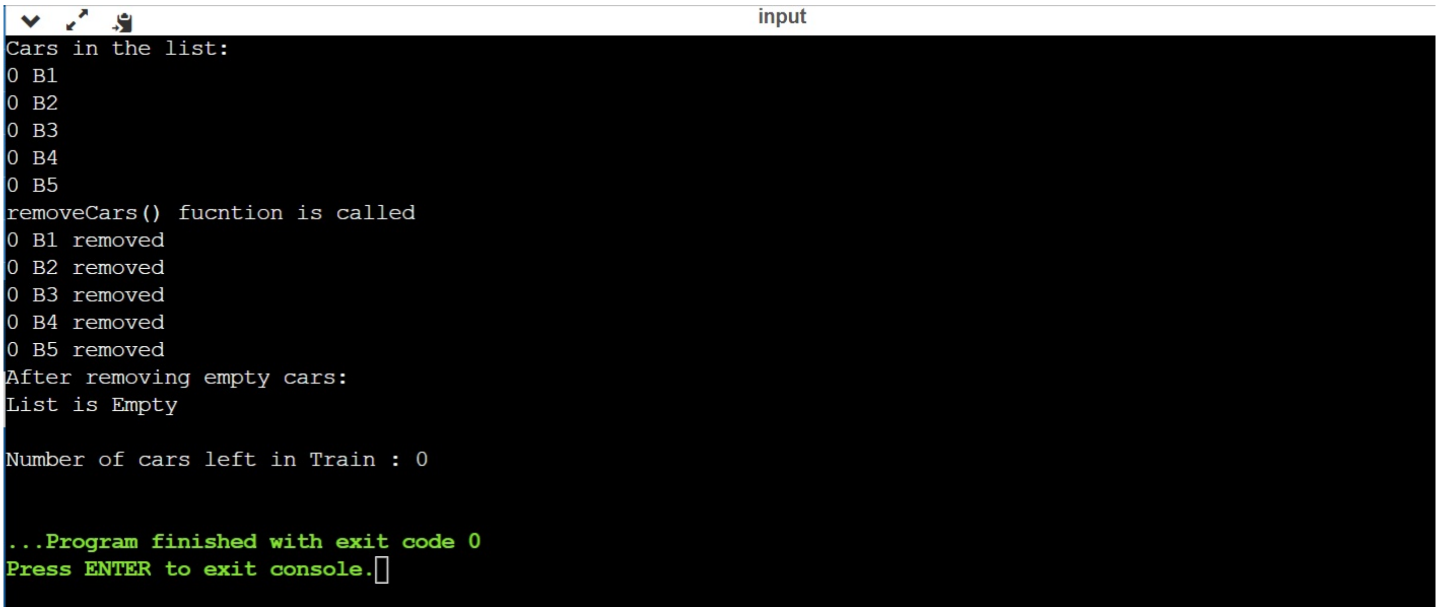
```c
    printf("\nremoveCars() fucntion is called ");
    removeCars(&head);
    //after removing empty cars ,display() fucntion is called
    printf("\nAfter removing empty cars:");
    display(head);
    //length() function is called and returned length of list is stored in l variable
    int l = length(head);
    //length of the list is printed
    printf("\nNumber of cars left in Train : %d\n",l);

    return 0;

}
//main() fucntion ended
```

Output Windnow 1:



```
Cars in the list:
83 B1
72 B2
0 B3
69 B4
0 B5
removeCars() fucntion is called
0 B3 removed
0 B5 removed
After removing empty cars:
83 B1
72 B2
69 B4
Number of cars left in Train : 3


...Program finished with exit code 0
Press ENTER to exit console.
```

Output Window 2:



```
Cars in the list:
0 B1
0 B2
0 B3
0 B4
0 B5
removeCars() fucntion is called
0 B1 removed
0 B2 removed
0 B3 removed
0 B4 removed
0 B5 removed
After removing empty cars:
List is Empty

Number of cars left in Train : 0


...Program finished with exit code 0
Press ENTER to exit console.
```

Please Upvote If You Liked This Answer :)