```
Please enter the number of values you want to enter:
6
Enter the element number 1 in your vector:
12
Enter the element number 2 in your vector:
34
Enter the element number 3 in your vector:
56
Enter the element number 4 in your vector:
34
Enter the element number 5 in your vector:
12
Enter the element number 6 in your vector:
7
Element number 1 is: 12 and it has occured 2 times
Element number 2 is: 7 and it has occured 3 times
Element number 3 is: 34 and it has occured 2 times
Element number 4 is: 56 and it has occured 1 times
Element number 5 is: 34 and it has occured 2 times
Element number 6 is: 12 and it has occured 2 times
Element number 7 is: 7 and it has occured 3 times
Element number 8 is: 7 and it has occured 3 times
Sum of elements is: 169

After Removal!

Element number 1 is: 12 and it has occured 2 times
Element number 2 is: 7 and it has occured 3 times
Element number 3 is: 34 and it has occured 2 times
Element number 4 is: 34 and it has occured 2 times
Element number 5 is: 12 and it has occured 2 times
Element number 6 is: 7 and it has occured 3 times
Element number 7 is: 7 and it has occured 3 times
Sum of elements is: 113
```

```
void add_node(int n){
    Node*new1=new Node;
    new1->data=n;
    numofoccurences(new1);
    if(head==NULL){head=new1;head->next=NULL;}
    else{
        Node*p=head;
        while(p->next!=NULL){p=p->next;}
        new1->next=NULL;
        p->next=new1;
    }
}
```

```
void printelements(){
    Node*p; p=head; int i=0; int sum;
    while(p!=NULL){
        cout<<"Element number "<<i+1<<" is: "<<p->data<<" and it has occured "<<p->occur<<" times\n";
        p=p->next;
        i++;
    }
    sum=this->sumOfnodes();
    cout<<"Sum of elements is: "<<sum<<"\n";
}
int sumOfnodes(){
    Node*p=p=head;
    int sum=0;
    while(p!=NULL){
        sum=sum+p->data;
        p=p->next;
    }
    return sum;
}
void numofoccurences(Node *h){
    Node*p;p=head;
    while(p!=NULL){
        if(h->data==p->data){
            h->occur++;
            p->occur++;
        }
        p=p->next;
    }
}
void vectorToLL(vector<int> v){
    for(int i=0;i<v.size();i++){
        this->add_node(v[i]);
    }
}
```

```
int main(){
    vector<int> x;
    int n;
    cout<<"Please enter the number of values you want to enter: \n";
    cin>>n;
    x.resize(n);
    for(int i=0;i<n;i++){
        cout<<"Enter the element number "<<i+1<<" in your vector:\n";
        cin>>x[i];
    }
    insertAfter(x,12,7);

    LinkedList LL;
    LL.vectorToLL(x);
    LL.printelements();
    LL.remove_node(3);
    cout<<"\n"<<"After Removal!"<<"\n"<<"\n";
    LL.printelements();

    return 0;
}
```

The special features of my code is the numofoccurences() function that takes a node as a parameter and checks every element in the list for any recurrence and increases the number of occurrences for both elements as the code shows. I reuse this function in the add_node() function to test for every new integer placed in the list. That way, the number of occurrences is always updated whenever a node is added. That is why the vectorToLL() function is short and concise. The print elements function is the function that prints all the info of the list: the elements, the sum, and the number of occurrences for each element. In the main part of the

code, I initialize a vector and allow the user to input the elements. Then after every 12, a 7 is inserted into the vector. Afterwards, the elements are put inside a Linkedlist and are printed with the correct number of occurrences as the output shows. Afterwards the 4th element is removed (it has index 3 as shown in the main). Then the list is printed again with every information accurate and to the point.