

Rajvaibhav Rahane  
17u283      223045  
SE-C Comp, Viit, Pune

---

**CODE:**

```
/*
 *
 *@Rajvaibhav Rahane
 */

#include<iostream>
using namespace std;
struct Node{
    int data;
    string name,branch,semester,phone;
    struct Node*next;
};
int size=0;
void printNode(Node *node){
    cout<<node->data<<" "<<node->name<<" "<<node->branch<<" "<<node->semester<<" ";
    cout<<node->phone<<endl;
}
Node* insertAtStart(Node* head,Node *nn){
    /*Node* nn=new Node;
    nn->data=data;           //init all data here;*/
    nn->next=head;
    head=nn;
    size++;
    return head;
}
Node* insertAtEnd(Node* head,Node *nn){
    /*Node* nn=new Node;
    nn->data=data;           //init all data here;*/
    nn->next=NULL;
    size++;
    Node* p=head;
    if(p!=NULL){
        while(p->next!=NULL)
            p=p->next;
        p->next=nn;
        return head;
    }else return nn;
}
Node* deleteNode(Node* head,int position){
    Node* ptr,*q;
    if(position==0){ //for first position
        ptr=head;
        head=head->next;
        ptr->next=NULL;
        delete(ptr);
    }else{ //for others
        ptr=head;
        for(int i=1;i<position;i++){
            ptr=ptr->next;
        }
        q=ptr->next;
        ptr->next=ptr->next->next;
        q->next=NULL;
        delete(q);
    }
    size--;
}
```

```

        return head;
    }
    Node *insertAtPosition(Node* head, Node *nn, int position){
        if(position<=0){
            return insertAtStart(head, nn);
        }
        else if(position>=size){
            return insertAtEnd(head, nn);
        }
        else{
            Node* ptr=head;
            for(int i=1; i<position; i++){
                ptr=ptr->next;
            }
            /*Node* nn=new Node;
            nn->data=data;*/
            nn->next=ptr->next;
            ptr->next=nn;
            size++;
        }
        return head;
    }
    void printAllNodes(Node* head){
        Node* ptr=head;
        while(ptr!=NULL){
            printNode(ptr);
            ptr=ptr->next;
        }
    }
    int indexOfNode(Node *head, int data){
        int index=-1;
        Node *p=head;
        while(p!=NULL){
            index++;
            if(p->data==data){
                return index;
            }
            p=p->next;
        }
        return -1;
    }
    Node *createNode(){
        Node *nn=new Node; nn->next=NULL;
        cin>>nn->data>>nn->name>>nn->branch>>nn->semester>>nn->phone;
        return nn;
    }
    void printMenu(){
        cout<<"1)Insert Node at start\t";
        cout<<"2)Insert Node at position\t";
        cout<<"3)Insert Node at end\n";
        cout<<"4)Delete Node at position\t";
        cout<<"5)Search Node\t";
        cout<<"6)Display SLL\t";
        cout<<"7)Print size of SLL\t";
        cout<<"8)End\n";
    }
    int main(){
        Node*head=NULL;
        /*head=insertAtStart(head,45);head=insertAtStart(head,47);
        head=insertAtPosition(head,46,1);head=insertAtEnd(head,70);printAllNodes(head);
        head=deleteNode(head,3);
        printAllNodes(head);cout<<size<<endl;

```

```

cout<<indexOfNode(head,47);*/
int choice,position,prn;
do{
    printMenu();
    cin>>choice;
    switch(choice){
        case 1:{
            head=insertAtStart(head,createNode());
            break;
        }
        case 2:{
            cout<<"position:";
            cin>>position;
            head=insertAtPosition(head,createNode(),position);
            break;
        }
        case 3:{
            head=insertAtEnd(head,createNode());
            break;
        }
        case 4:{
            cin>>position;
            head=deleteNode(head,position);break;
        }
        case 5:{
            cin>>prn;
            cout<<indexOfNode(head,prn)<<endl;break;
        }
        case 6:{
            printAllNodes(head);break;
        }
        case 7:{
            cout<<size<<endl;break;
        }
        case 8:{
            while(head!=NULL){
                head=deleteNode(head,0);
            }
            break;
        }
    }
}while(choice!=8);
return 0;
}

```

## Output:

```
rajrahane@visrajlenovo-g500: ~/Desktop/c++/LLUsingStruct
7)Delete Node at position      8)Delete Node      9)Display SLL      1)Print size of SLL      0)End
1
42 jugal comp 3 9082895951
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
3
46 shreya comp 3 8209481226
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
2
position:1
45 rajvaibhav comp 3 7083496708
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
7
3
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
6
42 jugal comp 3 9082895951
45 rajvaibhav comp 3 7083496708
46 shreya comp 3 8209481226
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
5
45
1
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
4
0
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
6
45 rajvaibhav comp 3 7083496708
46 shreya comp 3 8209481226
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
4
1
1)Insert Node at start 2)Insert Node at position      3)Insert Node at end
4)Delete Node at position      5)Search Node      6)Display SLL      7)Print size of SLL      8)End
6
45 rajvaibhav comp 3 7083496708
```