```
#include<iostream>
using namespace std;
class node
{
	public:
		int id;
		string name;
		int prn;
		float cgpa;
		node *next;
};
class SLL
{
	node *head;
	public:
		SLL()
		{
			head=NULL;
		}
		void create();
		void display();
		void search();
		void insert();
		void update();
		void del();
		void sort();
		void rev();
};

void SLL::create()
```

```
{
        node *temp=NULL;
        node *p=NULL;
        int c=1;
        temp=new(node);
        while(c!=0)
        {
                if(head==NULL)
                {
                        cout<<"Enter Roll No.: ";
                        cin>>temp->id;
                        cout<<"Enter Name: ";
                        cin>>temp->name;
                        cout<<"Enter PRN: ";
                        cin>>temp->prn;
                        cout<<"Enter CGPA: ";
                        cin>>temp->cgpa;
                        temp->next=NULL;
                        head=temp;
                        p=head;
                }
                else
                {
                   temp=new(node);
                        cout<<"Enter Roll No.: ";
                        cin>>temp->id;
                        cout<<"Enter Name: ";
                        cin>>temp->name;
                        cout<<"Enter PRN: ";
                        cin>>temp->prn;
```

```
                cout<<"Enter CGPA: ";
                cin>>temp->cgpa;
                temp->next=NULL;
                p->next=temp;
                p=temp;
        }
        cout<<"Enter 0 to exit: ";
        cin>>c;
    }
}

void SLL::display()
{
    node *p;
    p=head;
    if(head==NULL)
    {
        cout<<endl<<"Linked List is empty..."<<endl;
    }
    else
    {
        while(p!=NULL)
    {
        cout<<p->id<<" ";
        cout<<p->name<<" ";
        cout<<p->prn<<" ";
        cout<<p->cgpa<<" "<<endl;
        p=p->next;
    }
}
```

```
}
void SLL::search()
{
        node *p=NULL;
        int no,flag;
        p=head;
        cout<<"\nEnter Roll No. to search: ";
        cin>>no;
        while(p!=NULL)
        {
           if(p->id==no)
           {
                        flag=1;
                              break;
                }
                        p=p->next;
        }

        if(flag==1)
        {
           cout<<"Roll no. Present";
        }
        else
        {
                cout<<"Element not Present"<<endl;
        }
}
void SLL::update()
{
        node *p=NULL;
```

```
int no,c,flag;
p=head;
cout<<"\nEnter Roll No. to be updated: ";
cin>>no;
while(p!=NULL)
{
   if(p->id==no)
   {
                    flag=1;
                        break;
         }
                p=p->next;
}

if(flag==1)
{
  cout<<"\nElement found\n";
         do
         {
                cout<<"\tEnter what to be updated";
                cout<<"\n1.Roll number\n2.Name\n3.PRN\n4.CGPA\n5.Exit";
                cout<<"\nEnter choice:";
                cin>>c;
                switch(c)
                {
                        case 1:
                                cout<<"\nEnter new Roll number:";
                                cin>>p->id;
                                break;
                        case 2:
```

```
                                        cout<<"\nEnter new Name:";
                                        cin>>p->name;
                                        break;
                                case 3:
                                        cout<<"\nEnter new PRN:";
                                        cin>>p->prn;
                                        break;
                                case 4:
                                        cout<<"\nEnter new CGPA:";
                                        cin>>p->cgpa;
                                        break;
                        }
                }while (c!=5);
        }
        else
        {
                cout<<"Element not Present"<<endl;
        }
}
void SLL::insert()
{
        int ch;
        do
        {
                cout<<"\nEnter where to insert:\n1.Beginning\n2.Ending\n3.At Specific Position\n4.Between Value\n5.Exit Insertion"<<endl;
            cout<<"Enter your choice: ";
            cin>>ch;
            switch(ch)
            {
                case 1:
```

```
                {
                        node *temp=NULL;
                   temp=new(node);
                   cout<<"Enter Roll No.: ";
                   cin>>temp->id;
                   cout<<"Enter Name: ";
                                cin>>temp->name;
                                cout<<"Enter PRN: ";
                                cin>>temp->prn;
                                cout<<"Enter CGPA: ";
                                cin>>temp->cgpa;
                   temp->next=NULL;
                   temp->next=head;
                   head=temp;
                   cout<<"\nLinked List is: "<<endl;
display();
                   break;
              }
          case 2:
                {
                        node *p;
                   node *temp=NULL;
                   temp=new(node);
                   cout<<"Enter Roll No.: ";
                        cin>>temp->id;
                        cout<<"Enter Name: ";
                                cin>>temp->name;
                                cout<<"Enter PRN: ";
                                cin>>temp->prn;
                                cout<<"Enter CGPA: ";
```

```
                cin>>temp->cgpa;
            temp->next=NULL;
            p=head;
            while(p->next!=NULL)
            {
                p=p->next;
                }
                p->next=temp;
                cout<<"\nLinked List is: "<<endl;
display();
                break;
            }
        case 3:
            {
                int pos;
                node *p;
                node *temp=NULL;
                cout<<"Enter position where to insert: ";
                cin>>pos;
                temp=new(node);
                cout<<"Enter Roll No.: ";
                cin>>temp->id;
                cout<<"Enter Name: ";
                cin>>temp->name;
                cout<<"Enter PRN: ";
                cin>>temp->prn;
                cout<<"Enter CGPA: ";
                cin>>temp->cgpa;
                temp->next=NULL;
                p=head;
```

```
                    for(int i=1;i<pos-1;i++)
                    {
                        p=p->next;
                    }
                    temp->next=p->next;
                    p->next=temp;
                    cout<<"\nLinked List is: "<<endl;
display();
                    break;
                }
            case 4:
                {
                    node *p=NULL;
                    int no,flag;
                    node *temp=NULL;
                    p=head;
                    cout<<"\nEnter value after which Roll No. to be
inserted: ";
                    cin>>no;
                    while(p->next!=NULL)
                    {
                        if(p->next!=NULL)
                        {
                            if(p->id==no)
                            {
                                flag=1;
                                break;
                            }
                            p=p->next;
                        }
                    }
```

```
                    temp=new(node);
                    cout<<"\nEnter Roll No. to be inserted: ";
                    cin>>temp->id;
                    cout<<"Enter Name: ";
                    cin>>temp->name;
                    cout<<"Enter PRN: ";
                    cin>>temp->prn;
                    cout<<"Enter CGPA: ";
                    cin>>temp->cgpa;
                    temp->next=NULL;
                    if(flag==1)
                    {
                        if(p->next!=NULL)
                        {
                            temp->next=p->next;
                            p->next=temp;
                        }
                        else
                        {
                            p->next=temp;
                        }
                    }
                    else
                    {
                        cout<<"Element not Present"<<endl;
                    }
                    cout<<"\nLinked List is: "<<endl;
display();
                    break;
                }
```

```
                default:
                        {
                                break;
                        }
                }
        }while(ch!=5);
}

void SLL::del()
{
        int ch;
        do
        {
                cout<<"\n\nEnter which to delete:\n1.First\n2.Last\n3.At specific position\n4.Specific value\n5.Exit"<<endl;
                cin>>ch;
                switch(ch)
                {
                        case 1:
                                {
                                        node *p;
                                        p=head;
                                        head=p->next;
                                        delete(p);
                                        cout<<"\nLinked List is: "<<endl;
                   display();
                                        break;
                                }
                        case 2:
                                {
                                        node *p=NULL;
```

```
                node *temp=NULL;
                p=head;
                while(p->next!=NULL)
                {
                        temp=p;
                        p=p->next;
                }
                temp->next=NULL;
                delete(p);
                cout<<"\nLinked List is: "<<endl;
display();
break;
        }
        case 3:
                {
                        node *p=NULL;
                        node *temp=NULL;
                        int pos;
                        cout<<"\nEnter position which to be deleted: ";
                        cin>>pos;
                        p=head;
                        for(int i=1;i<pos-1;i++)
                        {
                                p=p->next;
                        }
                        temp=p->next;
                        p->next=p->next->next;
                        delete(temp);
                        cout<<"\nLinked List is: "<<endl;
  display();
```

```
break;
                }
        case 4:
                {
                        node *p=NULL;
                        node *temp=NULL;
                        int no,flag,c=0;
                        p=head;
                        temp=head;
                        cout<<"\nEnter Roll No. to be deleted: ";
                        cin>>no;
                        while(p->next!=NULL)
                        {
                                if(p->id==no)
                                {
                                        flag=1;
                                        break;
                                }
                                p=p->next;
                                c++;
                        }
                        for(int i=1;i<c;i++)
                        {
                                temp=temp->next;
                        }
                        if(flag==0)
                        {
                                cout<<"Element not present!";
                        }
                        else
```

```
                                        {
                                                if(p==head)
                                                {
                                                        head=p->next;
                                                        delete(p);
                                                }
                                                else if(p->next==NULL)
                                                {
                                                        temp->next=NULL;
                                                        delete(p);
                                                }
                                                else
                                                {
                                                        temp->next=p->next;
                                                        delete(p);
                                                }
                                        }
                                                cout<<"\nLinked List is: "<<endl;
                        display();
                        break;
                                        }
                           default:
                                   {
                                        break;
                                   }
                   }
           }while(ch!=5);
}

void SLL::sort()
```

```
{
    node *p;
    node *s;
    p=head;
    int z, w;
    string x;
    float y;
    while (p!=NULL)
    {
          s=p->next;
       while(s!=NULL)
       {
          if (p->id > s->id)
          {
             z=p->id;
             p->id=s->id;
             s->id=z;
             x=p->name;
             p->name=s->name;
             s->name=x;
             y=p->cgpa;
             p->cgpa=s->cgpa;
             s->cgpa=y;
             w=p->prn;
             p->prn=s->prn;
             s->prn=w;
             s=s->next;
          }
          else
          {
```

```
                s=s->next;
                    }
        }
        p=p->next;
    }
    cout<<"\nLinked List is: "<<endl;
        display();
}

void SLL::rev()
{
        node *p;
        node *q=NULL;
        p=head;
        node *r;
        r=p->next;
        while(p!=NULL)
        {
                p->next=q;
                q=p;
                p=r;
                if(p!=NULL)
                {
                        r=r->next;
                }
        }
        head=q;
        cout<<"\nLinked List is: "<<endl;
        display();
}
```

```
int main()
{
	SLL l;
	int c;
	do
	{
		cout<<"\n\tLinked list";

	cout<<"\n1.Create\n2.Display\n3.Search\n4.Insertion\n5.Update\n6.Deletion\n7.Sorti
ng\n8.Reversing\n9.Exit";
		cout<<"\nEnter choice:";
	   cin>>c;
	   switch(c)
	   {
		case 1:
			{
				l.create();
				break;
				}
			case 2:
				{
					cout<<"\nLinked List is: "<<endl;
						l.display();
						break;
				  }
		case 3:
			{
				l.search();
				break;
				}
		case 4:
```

```
{
        l.insert();
        break;
        }
case 5:
        {
                l.update();
                        cout<<"\nLinked List is: "<<endl;
                        l.display();
                break;
        }
case 6:
        {
                l.del();
                break;
        }
case 7:
        {
                l.sort();
                break;
        }
case 8:
        {
                l.rev();
                break;
        }
default:
        {
                break;
        }
```

```
        }
    }while(c!=9);
    return 0;
}
```