```
#include<iostream>

using namespace std;

class node

{

        public:

                int rno,marks;

                string name;

                node *prev;

                node *next;

};


class DLL

{

        node *head;

        public:

                DLL()

                {

                        head=NULL;

                }

                void create();

                void disp();

                void sort();

                void merge(DLL D1,DLL D2);

};


void DLL::create()

{

        node *temp;

        int c;

        do
        {
                temp=new(node);
                cout<<"\nEnter roll number, name and marks:";
                cin>>temp->rno>>temp->name>>temp->marks;
                temp->next=NULL;
                temp->prev=NULL;
                if (head==NULL)
                {
                        head=temp;
                }
                else
                {
                        node *p;
                        p=head;
                        while (p->next!=NULL)
                        {
                                p=p->next;
                        }
                        p->next=temp;
                        temp->prev=p;
                }
                cout<<"Enter 1 to EXIT: ";
        cin>>c;
        }while(c!=1);
}

void DLL::disp()
{
        if (head==NULL)
     cout<<"List Empty";
```

```
    else
    {
        node *p;
        p=head;
        while(p!=NULL)
        {
            cout<<p->rno<<"\t"<<p->name<<"\t"<<p->marks<<endl;
            p=p->next;
        }
    }
}

void DLL::sort()
{
            node *p, *q, *ptr1, *ptr2, *temp;
            p=head;
            while (p->next!=NULL)
            {
                        q=head;
                        while (q->next!=NULL)
                        {
                                    if (q->marks>q->next->marks)
                                    {
                                                ptr1=q;
                                                ptr2=q->next;
                                                temp=ptr2->next;
                                                if (ptr1->prev!=NULL)
                                                {
                                                            ptr1->prev->next=ptr2;
                                                }
```

```
                    else
                    {
                        head=ptr2;
                    }
                    ptr2->prev=ptr1->prev;
                    ptr2->next=ptr1;
                    ptr1->prev=ptr2;
                    ptr1->next=temp;
                    if (temp!=NULL)
                    {
                        temp->prev=ptr1;
                    }
                    q=ptr2;
                }
                q=q->next;
            }
            p=p->next;
        }
        disp();
}

void DLL::merge(DLL D1, DLL D2)
{
        node *p, *q, *r;
        if (D1.head==NULL && D2.head==NULL)
        {
            cout<<"\nLinks are empty!!";
        }
        else if (D1.head==NULL)
        {
```

```
        head=D2.head;
}
else if (D2.head==NULL)
{
        head=D1.head;
}
else
{
        p=D1.head;
        q=D2.head;
        if (p->marks<=q->marks)
        {
                head=p;
                p=p->next;
        }
        else
        {
                head=q;
                q=q->next;
        }
        r=head;
        while (p!=NULL && q!=NULL)
        {
                if (p->marks<=q->marks)
                {
                        r->next=p;
                        p->prev=r;
                        p=p->next;
                        r=r->next;
                }
```

```
				else
				{
					r->next=q;
					q->prev=r;
					q=q->next;
					r=r->next;
				}
			}
			if (p==NULL)
			{
				r->next=q;
				q->prev=r;
			}
			if (q==NULL)
			{
				r->next=p;
				p->prev=r;
			}
		}
		disp();
}

int main()
{
		DLL D1,D2,D3;
		int ch;
		cout<<"\tTechnical Scheme Exam of Recruitment Cell";
		do
		{
			cout<<"\n1.Create and display\n2.Sort\n3.Merge\n4.Exit\nEnter your choice:";
```

```
                cin>>ch;
                switch (ch)
                {
                        case 1:
                                cout<<"\nFist link list:";
                                D1.create();
                                cout<<"\nRoll\tName\tMarks\n";
                                D1.disp();
                                cout<<"\nSecond link list:";
                                D2.create();
                                cout<<"\nRoll\tName\tMarks\n";
                                D2.disp();
                                break;
                        case 2:
                                cout<<"\nFist sorted link list:\nRoll\tName\tMarks\n";
                                D1.sort();
                                cout<<"\n";
                                cout<<"\nSecond sorted link list:\nRoll\tName\tMarks\n";
                                D2.sort();
                                break;
                        case 3:
                                cout<<"\nMerged and sorted link list:\nRoll\tName\tMarks\n";
                                D3.merge(D1,D2);
                                break;
                }
        }while(ch!=4);
}
```