

NAME : Jadhav Vaibhav Satish

Batch : A4

PRN No. : 202201040027

ROLL NO. : 63

ASSIGNMENT NO. 4

CODE :

```
#include <iostream>
#include <string>
using namespace std;

class node
{
public:
    int marks;
    string skill;
    node *next;

    node *prev;
    node()
    {
        prev = NULL;
        next = NULL;
    }
};

class DLL
{
private:
    node *head;

public:
    DLL()
    {
        head = NULL;
    }

    void create();
    void display();
    void merge(DLL &l1, DLL &l2);
    void sort();
};

// create
void DLL::create()
{
    node *p = NULL;
```

```

node *temp = NULL;
int ch = 1;
while (ch == 1)
{
    if (head == NULL)
    {
        head = new (node);
        cout << "Enter the marks and skill: ";
        cin >> head->marks >> head->skill;
        p = head;
    }
    else
    {
        temp = new (node);
        cout << "Enter the marks and skill: ";
        cin >> temp->marks >> temp->skill;
        p->next = temp;
        temp->prev = p;
        p = p->next; // or p = temp
    }

    cout << "Enter the choice:" << endl
        << " 1.add" << endl
        << " 2.exit : ";
    cin >> ch;
}
};

void DLL::display()
{
    node *p1;
    p1 = head;
    while (p1 != NULL)
    {
        cout << p1->marks << endl
            << p1->skill << endl;
        p1 = p1->next;
    }
    cout << " _ _ _ _ _ " << endl;
};

void DLL::sort()
{
    node *p, *i;
    node *ptr2 = NULL;
    node *ptr1 = NULL;
    node *temp = NULL;
    for (i = head; i != NULL; i = i->next)
    {
        p = head;
        while (p->next != NULL)
        {

```

```

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

```

```

void DLL::merge(DLL &l1, DLL &l2)
{
    node *ptr1 = l1.head;
    node *ptr2 = l2.head;
    node *temp = new node;
    node *curpos = temp;
    while (ptr1 != NULL && ptr2 != NULL)
    {
        if (ptr1->marks <= ptr2->marks)
        {
            curpos->next = ptr1;
            ptr1->prev = curpos;
            ptr1 = ptr1->next;
        }
        else
        {
            curpos->next = ptr2;
            ptr2->prev = curpos;
            ptr2 = ptr2->next;
        }
        curpos = curpos->next;
    }
}

```

```

    }
    if (ptr1 != NULL)
    {
        curpos->next = ptr1;
        ptr1->prev = curpos;
    }
    if (ptr2 != NULL)
    {
        curpos->next = ptr2;
        ptr2->prev = curpos;
    }
    if (temp->next != NULL)
    {
        temp->next->prev = NULL;
    }
    head = temp->next;
    delete (temp);
}

int main()
{
    DLL l1, l2;
    cout << "Enter list 1: " << endl;
    l1.create();
    cout << "_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*" << endl;

    cout << "Enter list 2: " << endl;
    l2.create();
    cout << "_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*" << endl;

    cout << "LIST 1 :" << endl;
    l1.display();
    cout << "_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*" << endl;

    cout << "LIST 2 :" << endl;
    l2.display();
    cout << "_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*" << endl;

    l2.sort();
    cout << "SORTED LIST 1 :" << endl;
    l1.sort();
    l1.display();
    cout << "_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*" << endl;

    cout << "SORTED LIST 2 :" << endl;
    l2.sort();
    l2.display();
    cout << "_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*" << endl;

    DLL l3;
    l3.merge(l1, l2);
    l3.sort();
    cout << "Merged and sorted list: " << endl;
}

```

}

OUTPUT:

```
Enter list 1:
```

Enter the marks and skill: 94

html

Enter the choice:

1.add

```
2.exit : 1
```

Enter the marks and skill: 96

CSS

Enter the choice:

1.add

```
2.exit : 1
```

Enter the marks and skill: 85

JavaScript

Enter the choice:

1.add

```
2.exit : 2
```

-*-*-*-*-*

Enter list 2:

Enter the marks and skill: 81

AppDevelopment

Enter the choice:

1.add

```
2.exit : 1
```

Enter the marks and skill: 86

programming

Enter the choice:

1.add

```
2.exit : 1
```

Enter the marks and skill: 83

communication

Enter the choice:

1.add

```
2.exit : 2
```

- * - * - * - * - * - * - * - * - * - *

