**Practical No: 02**

**Implement the following for Linked List:**

1. **Write a program to create a single linked list and display the node elements in reverse order.**

**Code:**

#include<iostream>

#include<cstdlib>

using namespace std;

struct node{

int data;

struct node \*next;

};

struct node \*head=NULL;

void End(int value) {

struct node \*n=(struct node\*)malloc(sizeof(struct node));

n->data = value;

n->next = NULL;

if (head == NULL) {

head = n;

return;

}

struct node \*temp = head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = n;

}

void reverse(){

struct node \*t1=NULL;

struct node \*t2=NULL;

while(head!=NULL){

t2=head->next;

head->next=t1;

t1=head;

head=t2;

}

head=t1;

cout<<"The reverse list is: "<<endl;

}

void display(){

struct node \*temp=head;

while(temp!=NULL){

cout<<temp->data<<" ";

temp=temp->next;

}

}

int main(){

End(10);

End(20);

End(30);

End(40);

display();

cout<<endl;

reverse();

display();

}

**Output:**

**….…………………**

10 20 30 40

The reverse list is:

40 30 20 10

**…………………….**

1. **Write a program to search the elements in the linked list and display the same**

**Code:**

#include<iostream>

#include<cstdlib>

using namespace std;

struct node{

int data;

node \*next;

};

struct node \*head=NULL;

void inse(int v){

struct node \*n=(struct node \*)malloc(sizeof(struct node));

n->data=v;

n->next=NULL;

if(head==NULL){

head=n;

head->next=n;

}

else{

struct node \*t=head;

while(t->next!=head) {

t=t->next;

}

t->next=n;

n->next=head;

}

}

void search(int v){

int count=0;

struct node \*t=head;

while(t->next!=head){

if(v==t->data){

count=1;

}

t=t->next;

}

if(count==1){

cout<<endl<<"Element "<<v<<" found!";

}

else{

cout<<endl<<"Element "<<v<<" not found!";

}

}

void display(){

struct node \*s=head;

if(head!=NULL){

cout<<"The original list is: "<<endl;

while(s->next!=head){

cout<<s->data<<" ";

s=s->next;

}

cout<<s->data;

}

else{

cout<<"List is empty";

}

}

int main(){

inse(10);

inse(20);

inse(30);

inse(40);

display();

search(30);

}

**Output:**

**….………………..**

The original list is:

10 20 30 40 50

Element 30 found!

**….…………………**

1. **Write the program to create double linked list and sort the element in the linked list.**

**Code:**

#include <iostream>

#include <cstdlib>

using namespace std;

struct node

{

int data;

struct node \*prev;

struct node \*next;

};

struct node \*head = NULL;

void insert(int v)

{

struct node \*n=(struct node \*)malloc(sizeof(struct node));

n->prev=NULL;

n->data=v;

n->next=NULL;

if(head==NULL)

{

head=n;

}

else

{

struct node \*t=head;

while(t->next!=NULL)

{

t=t->next;

}

t->next=n;

n->prev=t;

}

}

void display(){

struct node \*t=head;

while(t!=NULL)

{

cout<<t->data<<"-->";

t=t->next;

}

}

void sort(){

struct node \*i=NULL;

struct node \*j=NULL;

cout<<"Sorted doubly linked list :"<<endl;

for(i=head;i->next!=NULL;i=i->next)

{

for(j=i->next;j!=NULL;j=j->next)

{

if(i->data > j->data)

{

i->data=i->data + j->data;

j->data=i->data - j->data;

i->data=i->data - j->data;

}

}

}

}

int main()

{

insert(10);

insert(40);

insert(30);

insert(20);

insert(50);

display();

cout<<endl;

sort();

display();

}

**Output:**

**….…………………………..**

10-->40-->30-->20-->50-->

Sorted doubly linked list :

10-->20-->30-->40-->50-->

**………………………………**