**EXPT 10:**

**Develop and execute a program in C using suitable data structures to perform Searching a data item in an ordered list of items in both directions and implement the following operations:**

**a. Create a doubly linked list by adding each node at the start.**

**b. Insert a new node at the end of the list.**

**c. Display the content of a list.**

**Consider an integer number as a data item.**

#include<stdio.h>

#include<stdlib.h>

struct node{

int item;

struct node \*prev;

struct node \*next;

};

struct node\*head,\*newnode,\*temp,\*tail;

void createnode(){

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

printf("Enter the items:\n");

scanf("%d",&newnode->item);

newnode->prev=NULL;

newnode->next=NULL;

if(head==NULL){

head=temp=newnode;

}

else{

temp->next=newnode;

newnode->prev=temp;

temp=newnode;

}

}

void insertatfront(){

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

printf("Enter the item:");

scanf("%d",&newnode->item);

newnode->prev=NULL;

newnode->next=NULL;

head->prev=newnode;

newnode->next=head;

head=newnode;

}

void insertatend(){

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

printf("Enter the items");

scanf("%d",&newnode->item);

newnode->prev=NULL;

newnode->next=NULL;

tail=head;

while(tail->next!=NULL){

tail=tail->next;

tail->next=newnode;

newnode->prev=tail;

tail=newnode;

printf("Elemenets added");

}

}

void display(){

temp=head;

printf("List Content:");

while(temp!=NULL){

printf("%d->",temp->item);

temp=temp->next;

}

printf("\n");

}

void search(){

int key;

temp=head;

int position=1;

printf("Enter the key:");

scanf("%d",&key);

while(temp!=NULL && temp->item!=key){

temp=temp->next;

position++;

}

if(temp==NULL){

printf("%d not found in a list",key);

}

else{

printf("%d found at the position % d in list",key,position);

}

}

void sortlist(){

struct node \*i,\*j;

int temp;

if(head==NULL){

printf("List is empty.Nothing to sort.\n");

return;

}

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

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

if(i->item>j->item){

temp=i->item;

i->item=j->item;

j->item=temp;

}

}

}

printf("List sorted in ascending order.\n");

}

int main(){

int choice;

head=temp=tail=newnode=NULL;

do{

printf("\n1Created doubly linked list\n");

printf("2.Insert a new node at front\n");

printf("3.Insert new node at end\n");

printf("4.display the list\n");

printf("5.Search the items\n");

printf("6.Sort the elements\n");

printf("7.Exit\n");

printf("Enter the choice:");

scanf("%d",&choice);

switch(choice){

case 1:

createnode();

break;

case 2:

insertatfront();

break;

case 3:

insertatend();

break;

case 4:

display();

break;

case 5:

search();

break;

case 6:

sortlist();

break;

case 7:

exit(0);

default:

printf("Invalid Choice.Please Try again.\n");

}

}while(choice!=7);

return(0);

}