

TECHPRODEZZA

//FILE DOWNLOADED FROM TECHPRODEZZA!

```
#include<iostream>
```

```
#include<stdlib.h>
```

```
using namespace std;
```

```
int pos,i=1,j=1,cta,ctb,in=0;
```

```
class list;
```

```
class node{
```

```
    public:
```

```
        int data;
```

```
        struct node *next;
```

```
    friend class list;
```

```
};
```

```
class list{
```

```
    struct node *head=NULL;
```

```
    public:
```

```
    void create();
```

```
    void insertion();
```

```
    void insertBegin();
```

```
    void insertEnd();
```

```
    void insertPos_a();
```

```
    void insertPos_b();
```

```
    void deletion();
```

```
    void deleteBegin();
```

```
    void deleteEnd();
```

```
    void deletePos();
```

```
    void deleteValue();
```

```
    void reverse();
```

```
    void display_a();
```

```
    void display_b();
```

```
    void display_con();
```

```
    void search();
```

TECHPRODEZZA

```
void concatenate(node *a,node *b);

void display();

int count();

void main1();

}a,b;

node *new_n,*temp,*temp3,*temp3a;

void list::create(){

    new_n=new (struct node);

    cout<<"\n Enter data";

    cin>>new_n->data;

    new_n->next=NULL;

    if(head==0){

        head=new_n;

        temp=new_n;

    }

    else{

        temp->next=new_n;

        temp=new_n;

    }

}

void list::display_a(){

    temp3=a.head;

    if(a.head==NULL)

        cout<<"\n List is Empty";

    while(temp3!=0){

        cout<< temp3->data<<" ";

        temp3=temp3->next;

        cta++;

    }

}

void list::display_b(){
```

TECHPRODEZZA

```
temp3=b.head;
if(b.head==NULL)
    cout<<"\n List is Empty";
while(temp3!=0){
    cout<< temp3->data<<" ";
    temp3=temp3->next;
    ctb++;
}
}

void list::display_con(){
    temp3=a.head;
    if(a.head==NULL)
        cout<<"\n List is Empty";
    while(temp3!=0){
        cout<< temp3->data<<" ";
        temp3=temp3->next;
    }
}

void list:: insertion(){
    int ci;
insertion_choice:
    cout<<"\n Enter the type you prefer:";
    cout<<"\n Insertion at";
    cout<<"\n 1.Beginning  2.End  3.Specific Position      4.Exit";
    cin>>ci;
    if(ci==1){
        insertBegin();
        goto insertion_choice;
    }
    else if(ci==2){
        insertEnd();
```

TECHPRODEZZA

```
        goto insertion_choice;
    }
    else if(ci==3){
        if(in==0)
            insertPos_a();
        else
            insertPos_b();
        goto insertion_choice;
    }
    else if(ci==4)    return;
    else{
        cout<<"\n Invalid Choice";
        goto insertion_choice;
    }
}

void list:: insertBegin(){
    new_n=new(struct node);
    cout<<"\n Enter the data to be inserted:";
    cin>>new_n->data;
    new_n->next=head;
    head=new_n;
}

void list:: insertEnd(){
    new_n=new(struct node);
    cout<<"\n Enter the data to be inserted:";
    cin>>new_n->data;
    new_n->next=NULL;
    temp=head;
    while(temp->next!=NULL)
        temp=temp->next;
    temp->next=new_n;
}
```

```
}
```

```
void list:: insertPos_a(){
```

```
    cout<<"\n Enter the position where the data has to be inserted:";
```

```
    cin>>pos;
```

```
    cout<<"Data before insertion:";
```

```
    display_a();
```

```
    if(pos>cta)
```

```
        cout<<"\n Invalid position";
```

```
    else{
```

```
        new_n=new(struct node);
```

```
        temp=head;
```

```
        while(i<pos-1){
```

```
            temp=temp->next;
```

```
            i++;
```

```
        }
```

```
        cout<<"\n Enter data to be inserted";
```

```
        cin>>new_n->data;
```

```
        new_n->next=temp->next;
```

```
        temp->next=new_n;
```

```
    }
```

```
}
```

```
void list:: insertPos_b(){
```

```
    cout<<"\n Enter the position where the data has to be inserted:";
```

```
    cin>>pos;
```

```
    cout<<"Data before insertion:";
```

```
    display_b();
```

```
    if(pos>ctb)
```

```
        cout<<"\n Invalid position";
```

```
    else{
```

```
        new_n=new(struct node);
```

```
        temp=head;
```

TECHPRODEZZA

```
        while(j<pos-1){
            temp=temp->next;
            j++;
        }

        cout<<"\n Enter data to be inserted";
        cin>>new_n->data;
        new_n->next=temp->next;
        temp->next=new_n;
    }
}

void list:: deletion(){
    int ci;
    deletion_choice:
        cout<<"\n Enter the type you prefer:";
        cout<<"\n Deletion";
        cout<<"\n 1.At Beginning      2.At End      3.At Specific Position      4.Of Specific Value\n 5.Exit";
        cin>>ci;
        if(ci==1){
            deleteBegin();
            goto deletion_choice;
        }
        else if(ci==2){
            deleteEnd();
            goto deletion_choice;
        }
        else if(ci==3){
            deletePos();
            goto deletion_choice;
        }
        else if(ci==4){
```

TECHPRODEZZA

```
        deleteValue();

        goto deletion_choice;

    }

    else if(ci==5)    return;

    else{

        cout<<"\n Invalid Choice";

        goto deletion_choice;

    }

}

void list::deleteBegin(){

    if(head==NULL)

        printf("\n List is Empty");

    else{

        temp=head;

        head=head->next;

        free(temp);

    }

}

void list::deleteEnd(){

    node *prevnode;

    temp=head;

    while(temp->next!=NULL){

        prevnode=temp;

        temp=temp->next;

    }

    if(temp==head)

        head=NULL;

    else

        prevnode->next=NULL;

    free(temp);

}
```

TECHPRODEZZA

```
void list::deletePos(){
    node *nextnode;
    int pos,i=1;
    temp=head;
    cout<<"\n Enter the position where the node has to be deleted:";
    cin>>pos;
    while(i<pos-1){
        temp=temp->next;
        i++;
    }
    nextnode=temp->next;
    temp->next=nextnode->next;
    free(nextnode);
}
```

```
void list::deleteValue(){
    int value;
    cout<<"Enter the value you wish to delete:";
    cin>>value;
    temp=head;
    node *temp1;
    if(temp->data==value)
    {
        head=head->next;
        free(temp);
    }
    else
        while(temp)
        {
            if(temp->next->data==value)
            {
                temp1=temp->next;
```


TECHPRODEZZA

```
        temp->next=temp1->next;

        free(temp1);

        return;

    }

    temp=temp->next;

}

cout<<"Number not found in the list";

}
```

void list::reverse(){

```
    struct node *prevnode,*cnode,*nnode;

    prevnode=NULL;

    cnode=nnode=head;

    while(nnode!=NULL){

        nnode=nnode->next;

        cnode->next=prevnode;

        prevnode=cnode;

        cnode=nnode;

    }

    head=prevnode;

}
```

void list::concatenate(node *a,node *b)

```
{

    if( a != NULL && b!= NULL )

    {

        if (a->next == NULL)

            a->next = b;

        else

            concatenate(a->next,b);

    }

    else

    {
```

TECHPRODEZZA

```
        cout << "Either a or b is NULL\n";
    }
}

void list::search(){
    int sa,csa=0;
    tempa=head;
    cout<<"\n Enter the element to search for:";
    cin>>sa;
    if(head==NULL){
        cout<<"\n List is Empty";
        return;
    }
    while(tempa!=NULL){
        if(tempa->data==sa){
            break;
        }
        tempa=tempa->next;
        csa++;
    }
    if(csa==0)
        cout<<"\n Element is not found";
    else{
        cout<<"\n Element is found at position " << csa+1;
        return;
    }
}

void list::main1(){
    int ch,chd,clc,chin,chdel,chrev,chs;
    printf("\n Thank you for downloading from Techprodezza!");
    choice:
    cout<<"\n Enter your choice";
```

TECHPRODEZZA

```
cout<<"\n 1.Create 2.Display 3.Insert 4.Delete 5.Reverse 6.Concatenate 7.Search 8.Exit";
cin>>ch;
if(ch==1){
create_node:
    cout<<"Create  1.Node A or    2.Node B 3.Go to Previous Menu";
    cin>>chc;
    if(chc==1){
        a.create();
        goto create_node;
    }
    else if(chc==2){
        b.create();
        goto create_node;
    }
    else if(chc==3)
        goto choice;
    else{
        cout<<"Invalid Choice";
        goto create_node;
    }
    goto choice;
}
else if(ch==2){
display_node:
    cout<<"\n Display      1.Node A or    2.Node B 3.Concatenated 4.Go to Previous
Menu";
    cin>>chd;
    if(chd==1){
        display_a();
        goto display_node;
    }
}
```

TECHPRODEZZA

```
else if(chd==2){
    display_b();
    goto display_node;
}
else if(chd==3){
    display_con();
    goto display_node;
}
else if(chd==4)
    goto choice;
else{
    cout<<"Invalid Choice";
    goto display_node;
}
goto choice;
}

else if(ch==3){
insert_node:
    cout<<"Insert for      1.Node A or    2.Node B 3.Go to Previous Menu";
    cin>>chin;
    if(chin==1){
        a.insertion();
        goto insert_node;
    }
    else if(chin==2){
        in=1;
        b.insertion();
        goto insert_node;
    }
    else if(chin==3)
        goto choice;
```

TECHPRODEZZA

```
else{
    cout<<"Invalid Choice";
    goto insert_node;
}
goto choice;
}
else if(ch==4){
delete_node:
    cout<<"Delete for      1.Node A or      2.Node B 3.Go to Previous Menu";
    cin>>chdel;
    if(chdel==1){
        a.deletion();
        goto delete_node;
    }
    else if(chdel==2){
        b.deletion();
        goto delete_node;
    }
    else if(chdel==3)
        goto choice;
    else{
        cout<<"Invalid Choice";
        goto delete_node;
    }
}
else if(ch==5){
rev_node:
    cout<<"Reverse      1.Node A or      2.Node B 3.Go to Previous Menu";
    cin>>chrev;
    if(chrev==1){
        a.reverse();
```

TECHPRODEZZA

```
        goto rev_node;
    }
    else if(chrev==2){
        b.reverse();
        goto rev_node;
    }
    else if(chrev==3)
        goto choice;
    else{
        cout<<"Invalid Choice";
        goto rev_node;
    }
}
else if(ch==6){
    concatenate(a.head,b.head);
    goto choice;
}
else if(ch==7){
search_list:
    cout<<"\n Choose where to search for:";
    cout<<"\n 1.List A      2.List B 3.Go to Previous Menu";
    cin>>chs;
    if(chs==1){
        a.search();
        goto search_list;
    }
    else if(chs==2){
        b.search();
        goto search_list;
    }
    else if(chs=3)
```

TECHPRODEZZA

```
        goto choice;

    else{

        cout<<"n Invalid Choice";

        goto search_list;

    }

}

else if(ch==8){

    cout<<"\n Program is terminated!";

}

else{

    cout<<"\n Invalid Choice";

    goto choice;

}

}

main(){

    list l;

    l.main1();

}
```