**L1)**

#include<stdio.h>

#include<stdlib.h>

#include<iostream.h>

#include<conio.h>

class node

{

public:

int constant;

int exponent;

node \*next;

node()

{

constant=0;

exponent=0;

next=NULL;

}

void insertNode(node\*,int,int);

void display(node\*);

void add(node\*,node\*,node\*);

};

//Function to insert a node to the list.

void node::insertNode(node \*head, int c, int e)

{

node\* temp = head;

node\* news = new node;

news->next=head;

while (temp->next != head)

temp = temp -> next;

temp -> next = news;

news -> constant = c;

news -> exponent = e;

}

//Function to display a polynomial.

void node::display(node\* head)

{

node\* temp;

temp = head;

while(temp->next != head)

{

temp = temp->next;

cout<<" + "<<temp->constant<<" X^ "<<temp->exponent;

}

}

//Function to subtract the two polynomials and make a new linked list.

void node::add(node \*h3, node \*h1, node \*h2)

{

int min;

node \*one;

node \*two;

one = h1 -> next;

two = h2 -> next;

while((one != h1) && (two != h2))

{

if(one->exponent == two->exponent)

{

min = one->constant + two->constant;

insertNode(h3,min,one->exponent);

one = one->next;

two = two->next;

}

else if(one->exponent > two->exponent)

{

insertNode(h3,one->constant,one->exponent);

one = one->next;

}

else

{

insertNode(h3,two->constant,two->exponent);

two = two->next;

}

}

while(one != h1)

{

insertNode(h3,one->constant,one->exponent);

one = one -> next;

}

while(two !=h2)

{

insertNode(h3,two->constant,two->exponent);

two = two -> next;

}

}

//Main function.

void main()

{

clrscr();

node obj;

int c,e,num;

node \*head1, \*head2, \*head3;

head1 = new node;

head2 = new node;

head3 = new node;

head1 -> next = head1;

head2 -> next = head2;

head3 -> next = head3;

cout<<"\nEnter the number of elements in first polynomial\n";

cin>>num;

while(num>0)

{

cout<<"\nEnter the exponent\n";

cin>>e;

cout<<"\nEnter the constant for the exponent\n";

cin>>c;

obj.insertNode(head1,c,e);

num--;

}

cout<<"\nEnter the number of elements in the polynomial to be added\n";

cin>>num;

while(num>0)

{

cout<<"\nEnter the exponent\n";

cin>>e;

cout<<"\nEnter the constant for the exponent\n";

cin>>c;

obj.insertNode(head2,c,e);

num--;

}

cout<<"\nPolynomial 1 is : \n\n";

obj.display(head1);

cout<<"\nPolynomial 2 is : \n\n";

obj.display(head2);

obj.add(head3,head1,head2);

cout<<"\n\nResult of addition of the two polynomials :\n\n";

obj.display(head3);

getch();

}

**L2)**

**#include<iostream.h>**

**#include<conio.h>**

**#include<process.h>**

**#include<stdlib.h>**

**#include<stdio.h>**

**//using namespace std;**

**class node{ int data;**

**node \*next, \*prev;**

**node \*head, \*tail;**

**int count;**

**public:**

**node()**

**{head = NULL;**

**tail = NULL;**

**count=0;**

**}**

**void insert();**

**void disp();**

**void in\_after();**

**void del();**

**void traverse();**

**};**

**void main()**

**{node obj;**

**int ch,n;**

**clrscr();**

**cout<<"enter no. of elements:-"<<endl;**

**cin>>n;**

**for(int i=0 ; i<n ; i++)**

**obj.insert();**

**obj.disp();**

**cout<<endl;**

**while(1)**

**{cout<<"1.Insert after\n2.Delete";**

**cout<<"\n3.Exit \n";**

**cout<<"enter your choice"<<endl;**

**cin>>ch;**

**switch(ch)**

**{**

**case 1 : obj.in\_after();obj.disp();break;**

**case 2 : obj.del();obj.disp();break;**

**case 3 : exit(0);**

**default : cout<<"enter correctly!"<<endl;**

**}**

**getch();**

**//return 0;**

**}**

**// return 0;**

**}**

**void node::insert()**

**{node \*temp = new node;**

**temp->next = NULL;**

**cout<<"enter value"<<endl;**

**cin>>temp->data;**

**if(head!=NULL && tail!=NULL)**

**{tail->next=temp;**

**temp->prev=tail;**

**temp->next=head;**

**head->prev=temp;**

**tail=temp;**

**count++;**

**}**

**else**

**{**

**tail=head=temp;**

**head->next=head;**

**head->prev=head;**

**count++;**

**}**

**// cout<<tail->next->data<<"\n";**

**}**

**void node::disp()**

**{**

**node \*temp=head;**

**if(count==0)**

**{**

**cout<<"empty!"<<endl;**

**return;**

**}**

**cout<<temp->data<<"\t";**

**temp=temp->next;**

**while(temp!=head)**

**{cout<<temp->data<<"\t";**

**temp=temp->next;**

**}**

**cout<<endl;**

**}**

**void node::del()**

**{ cout<<"enter element to be deleted"<<endl;**

**int item;**

**cin>>item;**

**node \*temp=head;**

**node \*d=head;**

**if(head->data==item)**

**{**

**head=head->next;**

**tail->next=head;**

**head->prev=tail;**

**delete (temp);**

**count--;**

**}**

**else**

**{**

**temp=temp->next;**

**if(tail->data==item)**

**{**

**temp=tail;**

**tail=tail->prev;**

**tail->next=head;**

**head->prev=tail;**

**delete(temp);**

**count--;**

**return;**

**}**

**while(temp!=head)**

**{ if(temp->data==item)**

**{**

**cout<<temp->data<<"\n";**

**cout<<head->data<<"\n";**

**cout<<tail->data<<"\n";**

**d->next=temp->next;**

**temp->next->prev=d;**

**delete(temp);**

**count--;**

**break;**

**}**

**d=temp;**

**temp=temp->next;**

**}**

**}**

**}**

**void node::in\_after()**

**{cout<<"enter element :-"<<endl;**

**int a,pos=0,c=1;**

**cin>>a;**

**if(head->data==a)**

**{**

**node \*link=new node;**

**cout<<"enter value"<<endl;**

**cin>>link->data;**

**link->next=head->next;**

**head->next->prev=link;**

**head->next=link;**

**link->prev=head;**

**count++;**

**return;**

**}**

**int co=1;**

**node \*temp=head;**

**while(temp!=NULL)**

**{ if(temp->data == a)**

**{ pos = c-1;**

**break;**

**}**

**temp=temp->next;**

**c++;**

**}**

**if(pos==0)**

**{ cout<<"element not present!"<<endl;**

**return;**

**}**

**while(co<pos-1)**

**{temp=temp->next;**

**co++;**

**}**

**node \*link=new node;**

**cout<<"enter value"<<endl;**

**cin>>link->data;**

**if(link->data==tail->data)**

**{**

**tail->next=link;**

**link->prev=tail;**

**tail=link;**

**tail->next=head;**

**head->prev=tail;**

**count++;**

**}**

**else**

**{**

**link->next=temp->next;**

**temp->next->prev=link;**

**temp->next=link;**

**link->prev=temp;**

**count++;**

**}**

**}**

**A1)**

#include<iostream.h>

#include<conio.h>

//using namespace std;

class node{ int data;

node \*next;

public:

node \*head, \*tail ;

node()

{head = NULL;

tail = NULL;

}

node\* insert(int);

void unions();

void intersection();

void disp(node \*);

};node \*head1, \*head2;

node \*node::insert(int a)

{head=NULL;

tail=NULL;

for(int i=0 ; i<a ; i++)

{node \*temp = new node;

temp->next = NULL;

cout<<"enter value"<<endl;

cin>>temp->data;

if(head!=NULL && tail!=NULL)

{tail->next=temp;

tail=temp;

}

else

tail=head=temp;

}

return head;

}

void node::disp(node \*t)

{if(t==NULL)

cout<<"empty!"<<endl;

while(t!=NULL)

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

t=t->next;

}

cout<<endl;

}

void node::intersection()

{ //node \*list = new node;

// node \*h;

int flag=0;

node \*temp1=head1;

node \*temp2=head2;

while(temp1!=NULL)

{ temp2=head2;

while(temp2!=NULL)

{

if(temp1->data==temp2->data)

{

cout<<temp1->data<<"\t";

flag=1;

}

temp2=temp2->next;

}

temp1=temp1->next;

}

if(flag==0)

cout<<"No intersection";

}

void node:: unions()

{

int flag=0;

node \*temp1=head1;

node \*temp2=head2;

while(temp1!=NULL)

{

cout<<temp1->data<<"\t";

temp1=temp1->next;

}

//temp1=head1;

while(temp2!=NULL)

{

temp1=head1;

flag=0;

while(temp1!=NULL)

{

if(temp1->data==temp2->data)

{

// cout<<temp2->data<<"\t";

flag=1;

}

temp1=temp1->next;

}

if(flag==0)

cout<<temp2->data;

temp2=temp2->next;

}

/\*if(flag==0)

cout<<"No union";\*/

}

void main()

{node obj;

node \*t = new node;

int n,m;

clrscr();

cout<<"no. of elements in 1st : ";cin>>n;

cout<<"no. of elements in 2nd : ";cin>>m;

head1=obj.insert(n);

head2=obj.insert(m);

cout<<"The union of the two list is\n";

obj.unions();

cout<<"\n";

cout<<"The intersection of the two list is\n";

obj.intersection();

// obj.disp(t);

getch();

//return 0;

}

**A2)**

#include<stdio.h>

#include<stdlib.h>

#include<iostream.h>

//using namespace std;

class node

{

public:

int x;

node \*l;

node \*r;

node()

{

l=NULL;

r=NULL;

}

void create(node\*\*);

void display(node\*);

void insert(node\*\*,int);

};

void node::create(node\*\*h)

{

\*h = new node;;

(\*h)->l=(\*h);

(\*h)->r=(\*h);

(\*h)->x=-1;

}

void node::display(node\*h)

{

cout<<"\n"<<h->x;

node\*t;

t=h->r;

while(t!=h)

{cout<<t->x;

t=t->r;

}

}

void node::insert(node\*\*h,int x)

{if((\*h)->x==-1)

{(\*h)->x=x;

}

else

{node \* temp = new node;

temp->r=(\*h);

temp->l=(\*h)->l;

((\*h)->l)->r=temp;

(\*h)->l=temp;

(\*h) = temp;

temp->x=x;

}

}

void main()

{

node\*h1,\*h2,\*h3,\*t1,\*t2;

node obj;

clrscr();

obj.create(&h1);

obj.create(&h2);

cout<<"Enter no. of digits for x and y\n";

int x,y;

int n;

cin>>x>>y;

int i;

cout<<"\nStarting from msb enter digits for num1 ";

for(i=0;i<x;i++)

{

cout<<"\nEnter dig for num1 ";

cin>>n;

obj.insert(&h1,n);

}

cout<<"\nStarting from msb enter digits for num1 ";

for(i=0;i<y;i++)

{

cout<<"\nEnter dig for num2 ";

cin>>n;

obj.insert(&h2,n);

}

obj.create(&h3);

int c=0;

t1=h1;

t2=h2;

//t3=h3;

for(i=0;i<x&&i<y;i++)

{int sum =0;

sum = t1->x + t2->x + c;

c = sum/10;

t1=t1->r;

t2=t2->r;

obj.insert(&h3,sum%10);

}

if(x!=y)

{if(x<y)

{for(i=x;i<y;i++)

{int sum =0;

sum = 0 + t2->x + c;

c = sum/10;

t2 = t2->r;

obj.insert(&h3,sum%10);

}}

else

{for(i=y;i<x;i++)

{int sum =0;

sum = 0 + t1->x + c;

c = sum/10;

t1=t1->r;

obj.insert(&h3,sum%10);}}}

obj.display(h3);

getch();

//return 0;

}

**A3)**

#include<iostream.h>

#include<conio.h>

class node{ int data;

node \*next;

public:

node \*head, \*tail ;

node()

{head = NULL;

tail = NULL;

}

node\* insert(int);

node\* merge();

void disp(node \*);

};node \*head1, \*head2;

node \*node::insert(int a)

{head=NULL;

tail=NULL;

for(int i=0 ; i<a ; i++)

{node \*temp = new node;

temp->next = NULL;

cout<<"enter value"<<endl;

cin>>temp->data;

if(head!=NULL && tail!=NULL)

{tail->next=temp;

tail=temp;

}

else

tail=head=temp;

}

return head;

}

void node::disp(node \*t)

{if(t==NULL)

cout<<"empty!"<<endl;

while(t!=NULL)

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

t=t->next;

}

cout<<endl;

}

node\* node::merge()

{ node \*list = new node;

node \*h = new node;

if(head1==NULL && head2==NULL)

return NULL;

if(head1==NULL)

return head2;

if(head2==NULL)

return head1;

if(head1->data > head2->data)

{list=h=head2;

head2=head2->next;

}

else

{list=h=head1;

head1=head1->next;

}

while(head1!=NULL && head2!=NULL)

{if(head1->data > head2->data)

{list->next=head2;

list=head2;

head2=head2->next;

}

else

{list->next=head1;

list=head1;

head1=head1->next;

}

}

if(head1==NULL)

list->next=head2;

else

list->next=head1;

return h;

}

void main()

{node obj;

node \*t = new node;

int n,m;

clrscr();

cout<<"no. of elements in 1st : ";cin>>n;

cout<<"no. of elements in 2nd : ";cin>>m;

head1=obj.insert(n);

head2=obj.insert(m);

t = obj.merge();

obj.disp(t);

getch();

}