lab 3:

#include <iostream>

using namespace std;

// Node class

class Node {

int data;

Node \*next;

public:

Node() {next=0;}

void SetData(int aData){data = aData;}

void SetNext(Node\* aNext){next = aNext;}

int Data(){return data;}

Node \*Next(){return next;}

};

// List class

class List {

Node \*head;

public:

List() {head = NULL;}

void Print()

{

Node \*temp;

temp = head;

if (head == NULL)

{cout<<"Empty"<<endl;

}

else

{

while(temp->Next() != NULL)

{cout<<temp->Data()<<" ";

temp = temp->Next();}

cout<<temp->Data();}

cout<<endl;

}

void Append(int data)

{Node \*a;

a = new Node;

a->SetNext(NULL);

a->SetData(data);

if (head == NULL)

{head = a;}

else if (head != NULL)

{Node \*temp;

temp = head;

while(temp->Next() != NULL)

{temp = temp->Next();}

temp->SetNext(a);

}}

void Insert(int data, int dataAfter );

void Swap(int d1, int d2)

{Node \*temp, \*temp1, \*temp2, \*temp3, \*tempn;

temp1 = head;

temp3 = head;

if(head == NULL)

{cout<<"List is empty"<<endl;}

else

{while(temp1->Data() != d1)

{temp = temp1;

temp1 = temp1->Next();}

while(temp3->Data() != d2)

{temp2 = temp3;

temp3 = temp3->Next();}

tempn = temp;

temp->SetNext(temp2->Next());

temp2->SetNext(tempn->Next());

tempn = temp1;

temp1->SetNext(temp3->Next());

temp3->SetNext(tempn->Next());

}}

void Delete(int data)

{Node \*temp, \*temp2;

temp = head;

int i=0;

if (head == NULL)

{cout<<"List is empty"<<endl;}

else

{while (temp->Data() != data)

{temp2 = temp;

temp = temp->Next();

i++;}

if(i == 0)

{head = temp->Next();

temp->SetData(0);

temp->SetNext(NULL);}

else

{temp2->SetNext(temp->Next());

temp->SetData(0);

temp->SetNext(NULL);}

}}

};

int main()

{

List list;

list.Append(100);

list.Print();

list.Append(200);

list.Print();

list.Append(300);

list.Print();

list.Append(400);

list.Print();

list.Append(500);

list.Print();

list.Append(600);

list.Print();

// list.Delete(100);

// list.Print();

// list.Delete(300);

// list.Print();

//list.Delete(500);

// list.Print();

// list.Delete(600);

// list.Print();

list.Swap(200, 500);

list.Print();

system("pause");

return 0:}