**LAB 05:**

**Question 01:**

public class Main{

List h;

class List{

int d;

List n;

List pre;

List(int d){

this.d=d;

}

}

public void inssert\_end(int d){

List nno=new List(d);

nno.d=d;

nno.n=null;

nno.pre=null;

if(h==null){

h=nno; }

else{

List tem=h;

tem=h;

while(tem.n!=null)

tem=tem.n;

tem.n=nno;

nno.pre=tem;

}

}

public void addfront(int num){

List nno=new List(num);

nno.d=num;

nno.n=null;

nno.pre=null;

if(h==null){

h=nno; }

else{

h.pre=nno;

nno.n=h;

h=nno; }

}

public void insert\_at\_any\_pos(int d,int pos){

List nno=new List(d);

nno.d=d;

int c=1;

List tem=h;

List tem1=h.n;

while(c<pos-1){

tem=tem.n;

tem1=tem1.n;

c++; }

tem.n=nno;

nno.pre=tem;

nno.n=tem1;

tem1.pre=nno;

}

public void delete\_last(){

if (this.h!=null) {

if (this.h.n==null) {

this.h=null; }

else{

List tem=h;

tem=this.h;

while(tem.n.n!=null)

tem=tem.n;

List nno=tem.n;

tem.n=null;

nno=null; }

}

}

public void delete\_first(){

if(this.h!=null){

List nno=this.h;

this.h=this.h.n;

nno=null;

if(this.h!=null)

this.h.pre=null;

} }

public void delete\_any\_pos(int x){

if(x<1){

System.out.print("\n point will be >= 1."); }

else if(x==1&&h!=null){

List nno=h;

h=h.n;

nno=null;

if (h!=null)

h.pre=null;

}

else{

List tem=h;

tem=h;

for(int i=1;i<x-1;i++){

if(tem!=null){

tem=tem.n;

}

}

if (tem!=null&&tem.n!=null) {

List nno=tem.n;

tem.n=tem.n.n;

if(tem.n.n!=null){

tem.n.n.pre=tem.n;

nno=null; }

else{

System.out.print("\n The node is already null."); }

}

}

}

public void print(){

List nno=h;

nno=this.h;

if(nno!=null){

System.out.print("The List is = ");

while(nno!=null){

System.out.print(nno.d+ " ");

nno=nno.n; }

System.out.println(); }

else{

System.out.println("The list is empty."); }

}

public static void main(String[] args) {

Main t=new Main();

System.out.println("Insert at the end:.");

t.inssert\_end(1);

t.inssert\_end(2);

t.inssert\_end(3);

t.print();

System.out.println("Insert in the beginning:.");

t.addfront(7);

t.addfront(6);

t.addfront(5);

t.print();

System.out.println("Insert at any postion:.");

t.insert\_at\_any\_pos(100,3);

t.print();

System.out.println("Delete last element:.");

t.delete\_last();

t.print();

System.out.println("Delete first element:.");

t.delete\_first();

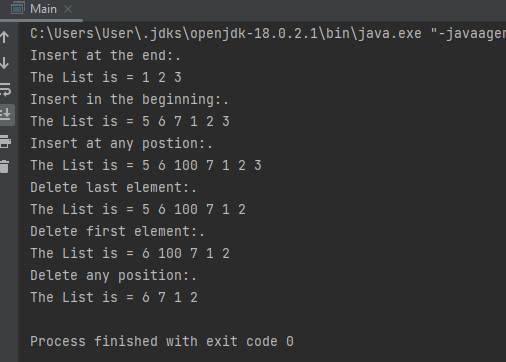
t.print();

System.out.println("Delete any position:.");

t.delete\_any\_pos(2);

t.print();

} }



**Question 02:**

package Qno2Lab5;

public class Qno2Lab5 {

public static void main(String[] args) {

clist cl = new clist();

cl.insert(1);

cl.insert(2);

cl.insert(3);

cl.insert(4);

cl.insert(5);

cl.insert(1);

cl.insert(2);

cl.insert(6);

cl.print();

clist c2 = new clist();

c2.insert(5);

c2.insert(1);

c2.insert(2);

c2.insert(6);

c2.insert(1);

c2.insert(2);

c2.insert(3);

c2.insert(4);

c2.print();

clist c3 = new clist();

// c3.compare(cl, c2);

boolean flag = c3.checkIdentical(cl, c2);

if (flag)

System.out.println("Yes");

else

System.out.println("No");

}

}

class Node

{

int data;

Node next;

}

class clist

{

Node head;

Node last;

int length;

public void insert(int a)

{

Node newnode = new Node();

newnode.data=a;

if(head==null)

{

head=newnode;

newnode.next=head;

last=head;

length=1;

}

newnode.next=head;

last.next=newnode;

last=last.next;

length++;

}

public boolean checkIdentical(clist cl, clist c2)

{

int l1 = length(cl.head);

int l2 = length(c2.head);

if (l1 != l2)

return false;

int Count = 0;

boolean flag = false;

Node h1 = cl.head;

Node h2 = c2.head;

while (true) {

if (h1.data == h2.data) {

h1 = h1.next;

Count++;

if (Count == l1)

return true;

}

else {

h1 = cl.head;

Count = 0;

if (flag)

return false;

}

if (h2.next == c2.head)

flag = true;

h2 = h2.next;

}

}

private static int length(Node head2) {

// TODO Auto-generated method stub

return 0;

}

public void print()

{

Node temp=head;

do

{

System.out.print(temp.data+" ");

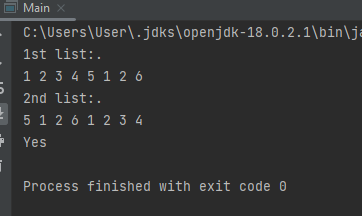
temp=temp.next;

}while(temp!=last.next);

System.out.println();

}

}



**Question 03:**

package Qno3Lab5;

import java.util.Scanner;

public class Qno3Lab5 {

public static void main(String[] args) {

Dlink dl = new Dlink();

dl.insert(1);

dl.insert(2);

dl.insert(3);

dl.insert(4);

dl.insert(5);

dl.insert(6);

dl.print();

Scanner sc = new Scanner(System.in);

System.out.println("Enter the value of X");

int x = sc.nextInt();

System.out.println("Enter the value of Y");

int y = sc.nextInt();

dl.swepone(x, y);

dl.print();

}

}

class Node

{

int data;

Node next;

Node previous;

}

class Dlink

{

static Node head;

public void insert(int a)

{

Node newnode = new Node();

newnode.data=a;

if(head==null)

{

head=newnode;

}

else

{

Node temp=head;

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=newnode;

newnode.previous=temp;

}

}

public void swepone(int x , int y)

{

Node temp=head;

int count=1;

while(temp!=null)

{

if(temp.data==x&&temp.next.data==y)

{

count++;

// break;

}

if(temp.data==y&&temp.previous.data==x)

{

System.out.println("Node x is the previous node of node y in the list");

count=1;

break;

}

else if(temp.data==y&&temp.next.data==x)

{

System.out.println("Node x is the next node after node y in the list");

count=1;

break;

}

else if(temp.data==x&&temp.data==y)

{

System.out.println("x and y are the same node (no need to swap)");

count++;

break;

}

else

{

count=1;

}

temp=temp.next;

}

if(count==1)

{

System.out.println("Swep");

swapNodes( x, y);

}

else

{

System.out.println("Not swep");

}

}

public static void swapNodes(int x, int y)

{

if (x == y)

return;

Node prevX = null;

Node currX = head;

while (currX != null && currX.data != x) {

prevX = currX;

currX = currX.next;

currX.previous=currX;

}

Node prevY = null;

Node currY = head;

while (currY != null && currY.data != y) {

prevY = currY;

currY = currY.next;

currY.previous=currY;

}

if (currX == null || currY == null)

return;

if (prevX != null)

{

prevX.next = currY;

currY.previous=prevX;

}

else

{

head = currY;

}

if (prevY != null)

{

prevY.next = currX;

currX.previous=prevY;

}

else

{

head = currX;

}

Node temp = currX.next;

currX.next = currY.next;

currY.next = temp;

temp.previous=currY;

}

public void print()

{

Node temp=head;

while(temp!=null)

{

System.out.print(temp.data+" ");

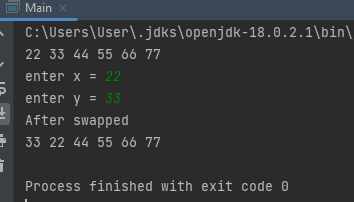
temp=temp.next;

}

System.out.println();

}

}



**Question 04:**

package Qno4Lab5;

import Qno4Lab5.Node;

public class Qno4Lab5 {

public static void main(String[] args) {

cdlist cdl = new cdlist();

cdl.insert(1);

cdl.insert(9);

cdl.insert(12);

cdl.insert(7);

cdl.print();

cdl.any();

}

}

class Node

{

Node next;

Node previous;

int data;

}

class cdlist

{

Node head;

Node last;

int length;

public void insert(int a)

{

Node newnode = new Node();

newnode.data=a;

if(head==null)

{

head=newnode;

newnode.next=head;

newnode.previous=head;

last=head;

length=1;

}

newnode.next=head;

newnode.previous=head;

last.next=newnode;

last=last.next;

length++;

}

public void any()

{

Node temp=head;

Node ptr=null;

int count=0;

int flag=0;

do

{

ptr=temp;

while(count!=3)

{

System.out.print(ptr.data+" ");

ptr=ptr.next;

count++;

}

System.out.print(" ");

temp=temp.next;

temp.previous=temp;

count=0;

flag++;

}while(flag!=5);

}

public void print()

{

Node temp=head;

do

{

System.out.print(temp.data+" ");

temp=temp.next;

}while(temp!=last.next);

System.out.println();

}

}

