DAY 16

/\*

class Node {

int data;

Node next, prev;

Node(int x) {

data = x;

next = prev = null;

}

}

class DoubleLinkedList {

Node head;

Node createNode(int x) {

Node np = new Node(x);

return np;

}

void insertEnd(int x) {

Node nptr = createNode(x);

if(head == null) {

head = nptr;

}

else {

Node temp = head;

while(temp.next != null) {

temp = temp.next;

}

temp.next = nptr;

nptr.prev = temp;

}

}

void insertBeg(int x) {

Node nptr = createNode(x);

if(head == null) {

head = nptr;

}

else {

nptr.next = head;

head.prev = nptr;

head = nptr;

}

}

void deleteEnd() {

if(head == null) {

System.out.println("No node to delete");

return;

}

if(head.next == null) {

System.out.println("Deleted item: " + head.data);

head = null;

return;

}

Node temp = head;

while(temp.next.next != null) {

temp = temp.next;

}

System.out.println("Deleted item: " + temp.next.data);

temp.next = null;

}

void deleteBeg() {

if(head == null) {

System.out.println("No node to delete");

return;

}

System.out.println("Deleted item: " + head.data);

head = head.next;

head.prev = null;

}

void display() {

if(head == null) {

System.out.println("List Empty");

return;

}

System.out.println("List elements are: ");

Node temp = head;

while(temp != null) {

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

temp = temp.next;

}

System.out.println();

}

void displayReverse() {

if(head == null) {

System.out.println("List Empty");

return;

}

System.out.println("List elements are: ");

Node temp = head;

while(temp.next != null) {

temp = temp.next;

}

while(temp != null) {

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

temp = temp.prev;

}

System.out.println();

}

public static void main(String[] args) {

DoubleLinkedList li = new DoubleLinkedList();

li.head = null;

li.insertEnd(10);

li.display();

li.insertEnd(20);

li.display();

li.insertEnd(30);

li.display();

li.insertBeg(5);

li.display();

li.displayReverse();

li.deleteEnd();

li.display();

li.deleteBeg();

li.display();

}

}

\*/

class Node {

int data;

Node next;

}

class CircularLinkedList {

Node tail;

Node createNode(int x) {

Node np = new Node();

np.data = x;

np.next = np;

return np;

}

void insertEnd(int x) {

Node nptr = createNode(x);

if(tail == null) {

tail = nptr;

}

else {

nptr.next = tail.next;

tail.next = nptr;

tail = nptr;

}

}

void insertBeg(int x) {

Node nptr = createNode(x);

if(tail == null) {

tail = nptr;

}

else {

nptr.next = tail.next;

tail.next = nptr;

}

}

void deleteEnd() {

if(tail == null) {

System.out.println("No node to delete");

return;

}

if(tail.next == tail) {

System.out.println("Deleted item: " + tail.data);

tail = null;

return;

}

Node temp = tail.next;

while(temp.next != tail) {

temp = temp.next;

}

System.out.println("Deleted item: " + tail.data);

temp.next = tail.next;

tail = temp;

}

void deleteBeg() {

if(tail == null) {

System.out.println("No node to delete");

return;

}

System.out.println("Deleted item: " + tail.next.data);

tail.next = tail.next.next;

}

void display() {

if(tail == null) {

System.out.println("List Empty");

return;

}

System.out.println("List elements are: ");

Node temp = tail.next;

while(temp != tail) {

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

temp = temp.next;

}

System.out.println(tail.data);

}

public static void main(String[] args) {

CircularLinkedList li = new CircularLinkedList();

li.tail = null;

li.insertEnd(10);

li.display();

li.insertEnd(20);

li.display();

li.insertEnd(30);

li.display();

li.insertBeg(5);

li.display();

li.deleteEnd();

li.display();

li.deleteBeg();

li.display();

}

}