**// DOUBLY LINKED LIST**

import java.util.Scanner;

class DoublyLinkedList

{

private Node head;

class Node

{

private int data;

private Node left;

private Node right;

public Node(int data)

{

this.data = data;

this.left = null;

this.right = null;

}

}

public void insertAtEnd(int data)

{

Node temp = new Node(data);

if(head == null)

{

head = temp;

}

else

{

Node ptr = head;

while(ptr.right != null)

{

ptr = ptr.right;

}

ptr.right = temp;

temp.left = ptr;

}

System.out.println(data+ " is inserted Successfully");

}

public void deleteAtFront()

{

if(head == null)

{

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

}

else

{

int data = head.data;

head = head.right;

head.left = null;

System.out.println(data+" is deleted from the list");

}

}

public void display()

{

Node temp = head;

if(head == null)

{

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

}

else

{

while(temp != null)

{

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

temp = temp.right;

}

System.out.println();

}

}

}

class Test{

public static void main(String args[]){

DoublyLinkedList dl = new DoublyLinkedList();

int ch = 0;

while(ch != 4){

System.out.println("1.Insert\n2.Delete\n3.Display\n4.Exit");

Scanner sc = new Scanner(System.in);

ch = sc.nextInt();

sc.nextLine();

switch(ch){

case 1: System.out.println("Enter the element");

int elt = sc.nextInt();

sc.nextLine();

dl.insertAtEnd(elt);

break;

case 2:dl.deleteAtFront();

break;

case 3:dl.display();

break;

case 4: break;

default: System.out.println("Invalid Choice");

}

}

}

}