import java.util.\*;

//write and execute a java program to insert a new node after a given node(based on node value) in a ddl

class Node {

    int data;

    Node prev;

    Node next;

    Node(int d) {

        data = d;

        prev = null;

        next = null;

    }

}

public class question1 {

    static Node head = null;

    public static void insertAtEnd(int value) {

        Node newNode = new Node(value);

        if (head == null) {

            head = newNode;

            return;

        }

        Node current = head;

        while (current.next != null) {

            current = current.next;

        }

        current.next = newNode;

        newNode.prev = current;

    }

    public static void insertAfterValue(int targetValue, int newValue) {

        Node current = head;

        while (current != null && current.data != targetValue) {

            current = current.next;

        }

        if (current == null) {

            System.out.println("Value " + targetValue + " not found.");

            return;

        }

        Node newNode = new Node(newValue);

        newNode.next = current.next;

        newNode.prev = current;

        if (current.next != null) {

            current.next.prev = newNode;

        }

        current.next = newNode;

    }

    public static void printList() {

        Node current = head;

        System.out.print("Doubly Linked List: ");

        while (current != null) {

            System.out.print(current.data + " <-> ");

            current = current.next;

        }

        System.out.println("null");

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Initial DLL

        insertAtEnd(11);

        insertAtEnd(12);

        insertAtEnd(14);

        insertAtEnd(15);

        printList();

        System.out.print("Enter the value after which to insert: ");

        int target = scanner.nextInt();

        System.out.print("Enter the new value to insert: ");

        int newValue = scanner.nextInt();

        insertAfterValue(target, newValue);

        printList();

        scanner.close();

    }

}

//2 write and execute a java program to insert a new node before a given node in a DLL (doubly linked list)

//3write and execute a java program to sort a dll in ascending order using bubble sort

//4 write a execute a java program to delete all nodes with a given value in a dll

import java.util.Scanner;

public class question2 {

    // Node class for DLL

    static class Node {

        int data;

        Node prev;

        Node next;

        Node(int d) {

            data = d;

            prev = null;

            next = null;

        }

    }

    // Head of the DLL

    Node head = null;

    // Method to insert at end

    void insertAtEnd(int value) {

        Node newNode = new Node(value);

        if (head == null) {

            head = newNode;

            return;

        }

        Node current = head;

        while (current.next != null) {

            current = current.next;

        }

        current.next = newNode;

        newNode.prev = current;

    }

    // Method to insert before a given value

    void insertBefore(int target, int newValue) {

        if (head == null) {

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

            return;

        }

        Node current = head;

        // If inserting before head

        if (current.data == target) {

            Node newNode = new Node(newValue);

            newNode.next = head;

            head.prev = newNode;

            head = newNode;

            return;

        }

        // Traverse to find the target

        while (current != null && current.data != target) {

            current = current.next;

        }

        if (current == null) {

            System.out.println("Target value not found.");

            return;

        }

        Node newNode = new Node(newValue);

        Node prevNode = current.prev;

        newNode.prev = prevNode;

        newNode.next = current;

        prevNode.next = newNode;

        current.prev = newNode;

    }

    void printList() {

        Node current = head;

        System.out.print("DLL: ");

        while (current != null) {

            System.out.print(current.data + " <-> ");

            current = current.next;

        }

        System.out.println("null");

    }

    public static void main(String[] args) {

        question2 dll = new question2();

        // Predefined list

        dll.insertAtEnd(10);

        dll.insertAtEnd(20);

        dll.insertAtEnd(30);

        dll.insertAtEnd(40);

        System.out.println("Original List:");

        dll.printList();

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter value to insert before: ");

        int target = sc.nextInt();

        System.out.print("Enter new value to insert: ");

        int newVal = sc.nextInt();

        dll.insertBefore(target, newVal);

        System.out.println("After Insertion:");

        dll.printList();

    }

}

//3write and execute a java program to sort a dll in ascending order using bubble sort

public class question3 {

    // Node class for DLL

    static class Node {

        int data;

        Node prev;

        Node next;

        Node(int data) {

            this.data = data;

            this.prev = null;

            this.next = null;

        }

    }

    // Head of the DLL

    Node head = null;

    // Insert at the end

    void insertAtEnd(int data) {

        Node newNode = new Node(data);

        if (head == null) {

            head = newNode;

            return;

        }

        Node current = head;

        while (current.next != null) {

            current = current.next;

        }

        current.next = newNode;

        newNode.prev = current;

    }

    // Bubble Sort on DLL

    void bubbleSort() {

        if (head == null) return;

        boolean swapped;

        Node current;

        do {

            swapped = false;

            current = head;

            while (current.next != null) {

                if (current.data > current.next.data) {

                    // Swap data

                    int temp = current.data;

                    current.data = current.next.data;

                    current.next.data = temp;

                    swapped = true;

                }

                current = current.next;

            }

        } while (swapped);

    }

    // Print the list

    void printList() {

        Node current = head;

        System.out.print("DLL: ");

        while (current != null) {

            System.out.print(current.data + " <-> ");

            current = current.next;

        }

        System.out.println("null");

    }

    // Main function

    public static void main(String[] args) {

        question3 dll = new question3();

        // Add elements

        dll.insertAtEnd(45);

        dll.insertAtEnd(10);

        dll.insertAtEnd(32);

        dll.insertAtEnd(8);

        dll.insertAtEnd(17);

        System.out.println("Before Sorting:");

        dll.printList();

        dll.bubbleSort();

        System.out.println("After Sorting in Ascending Order:");

        dll.printList();

    }

}