**Experiment – 11**

Q1) Write a program that lets the user enter numbers from a graphical user interface and displays them in a text area, as shown in Figure. Use a linked list to store the numbers. Do not store duplicate numbers. Add the buttons Sort, Shuffle, and Reverse to sort, shuffle, and reverse the list.



Ans:

**Program:**

package application;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import javafx.application.Application; import javafx.stage.Stage;

import javafx.scene.Scene;

import javafx.scene.control.Button; import javafx.scene.control.Label; import javafx.scene.control.TextArea; import javafx.scene.control.TextField; import javafx.scene.layout.AnchorPane;

public class set\_11\_01 extends Application { Node head = null;

public void start(Stage primaryStage) { try {

set\_11\_01 linklist = new set\_11\_01();

Label numlabel = new Label("Enter a number : "); numlabel.setLayoutX(20); numlabel.setLayoutY(20);

TextField inpnum = new TextField(); inpnum.setLayoutX(120); inpnum.setLayoutY(18);

TextArea textarea = new TextArea(); textarea.setLayoutX(20); textarea.setLayoutY(50); textarea.setPrefWidth(250); textarea.setPrefHeight(100);

Button addnum = new Button("Add"); addnum.setLayoutX(280); addnum.setLayoutY(18);

addnum.setOnAction(e -> {

textarea.setText(textarea.getText()+" "+inpnum.getText()); linklist.add(Integer.parseInt(inpnum.getText())); inpnum.clear();

});

Button bsort = new Button("Sort"); bsort.setLayoutX(20); bsort.setLayoutY(170); bsort.setOnAction(e -> {

linklist.sortlist(); textarea.setText(linklist.displaylist());

});

Button bshuffle = new Button("Shuffle"); bshuffle.setLayoutX(70); bshuffle.setLayoutY(170); bshuffle.setOnAction(e -> {

linklist.shufflelist(); textarea.setText(linklist.displaylist());

});

Button breverse = new Button("Reverse"); breverse.setLayoutX(130); breverse.setLayoutY(170); breverse.setOnAction(e -> {

linklist.reverselist(); textarea.setText(linklist.displaylist());

});

AnchorPane root = new AnchorPane(); root.getChildren().addAll(numlabel,inpnum,textarea,bsort,bshuffle,breverse,addnum);

Scene scene = new Scene(root); primaryStage.setScene(scene); primaryStage.show();

} catch(Exception e) {

e.printStackTrace();

}

}

class Node{

int data;

Node next;

Node(int data){

this.data = data; this.next = null;

}

}

public String displaylist() {

StringBuilder textlist = new StringBuilder(); Node p = head;

while(p!=null) {

textlist.append(p.data).append(" "); p = p.next;

}

return textlist.toString();

}

public void add(int data) {

Node newNode = new Node(data);

if(head == null) {

head = newNode; return;

}

Node p = head; while(p.next != null) {

p = p.next;

}

p.next = newNode;

}

public void sortlist() {

Node p = head; int size = 0; while(p!=null) {

p = p.next; size++;

}

p=head;

int[] arr = new int[size]; int index=0; while(p!=null) {

arr[index++] = p.data; p=p.next;

}

Arrays.sort(arr); index = 0; p=head; while(p!=null) {

p.data = arr[index++]; p=p.next;

}

}

public void shufflelist() {

ArrayList<Integer> arlist = new ArrayList<>(); Node p = head;

while(p!=null) {

arlist.add(p.data); p=p.next;

}

Collections.shuffle(arlist);

p = head;

for(int value : arlist) {

p.data = value; p = p.next;

}

}

public void reverselist() {

ArrayList<Integer> arlist = new ArrayList<>(); Node p = head;

while(p!=null) {

arlist.add(p.data); p=p.next;

}

Collections.reverse(arlist); p = head;

for(int value : arlist) {

p.data = value; p = p.next;

}

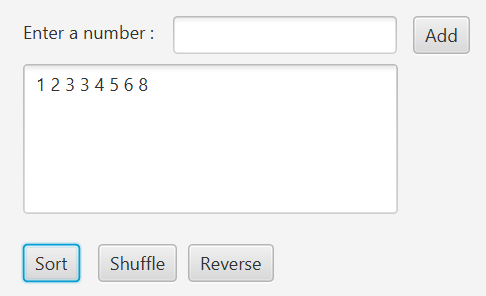
}

public static void main(String[] args) { launch(args);

}

}

**Output:**



Q2) Create two priority queues, {"George", "Jim", "John", "Blake", "Kevin", "Michael"} and {"George", "Katie", "Kevin", "Michelle", "Ryan"}, and find their union, difference, and intersection.

Ans:

**Program:**

import java.util.\*; class set\_11\_02{

public static void main(String args[]) { PriorityQueue<String> q1 = new PriorityQueue<>(); PriorityQueue<String> q2 = new PriorityQueue<>();

q1.addAll(Arrays.asList("George","Jim","Blake","Kevin","Micheal"));

q2.addAll(Arrays.asList("George","Katie","Kevin","Michelle","Ryan"));

System.out.println("Queue 1 : "+q1); System.out.println("Queue 2 : "+q2);

PriorityQueue<String> union = new PriorityQueue<>(q1); union.addAll(q2);

System.out.println("Union : "+union);

PriorityQueue<String> difference = new PriorityQueue<>(q1); difference.removeAll(q2);

System.out.println("Difference : "+difference);

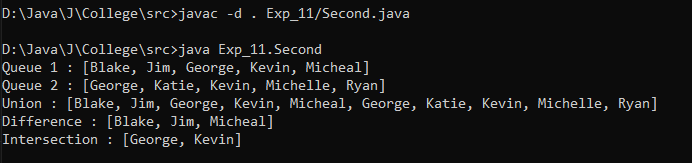
PriorityQueue<String> intersection = new PriorityQueue<>(q1); intersection.retainAll(q2);

System.out.println("Intersection : "+intersection);

}

}

**Output:**



Q3) Store pairs of 10 states and its capital in a map. Your program should prompt the user to enter a state and should display the capital for the state.

Ans:

**Program:**

package Exp\_11;

import java.util.Map;

import java.util.HashMap;

import java.util.Scanner;

public class Third {

    public static void main(String args[])

    {

        Map<String,String> stateCapital = new HashMap<>();

        stateCapital.put("Gujarat","Gandhinagar");

        stateCapital.put("Maharashtra", "Mumbai");

        stateCapital.put("Rajasthan", "Jaipur");

        stateCapital.put("Karnataka", "Bengaluru");

        stateCapital.put("Tamil Nadu", "Chennai");

        stateCapital.put("Uttar Pradesh", "Lucknow");

        stateCapital.put("Bihar", "Patna");

        stateCapital.put("Punjab", "Chandigarh");

        stateCapital.put("West Bengal", "Kolkata");

        stateCapital.put("Madhya Pradesh", "Bhopal");

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the name of state : ");

        String state = sc.nextLine();

        String capital = stateCapital.get(state);

        if(capital != null){

            System.out.println("The capital of state "+state+" is "+capital);

        }

        else {

            System.out.println("The state is not found in the list");

        }

    }

}

**Output:**

