**Experiment – 10**

Q1) Write a recursive method that converts a decimal number into a binary number as a string. The method header is: public static String dec2Bin(int value)

Write a test program that prompts the user to enter a decimal number and displays its binary equivalent.

Ans:

**Program:**

// convert decimal to binary string by recursion

package Exp\_10;

import java.util.Scanner;

public class First {

    public static String dec2Bin(int value) {

        if (value == 0) {

            return "0";

        }

        if (value == 1) {

            return "1";

        }

        return dec2Bin(value / 2) + (value % 2);

    }

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a decimal number: ");

        int decimal = input.nextInt();

        if (decimal < 0) {

            System.out.println("Please enter a non-negative number.");

        } else{

            System.out.println("Binary equivalent: " + dec2Bin(decimal));

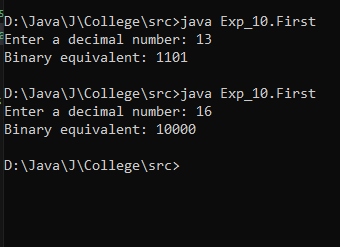
        }

        input.close();

    }

}

**Output:**



Q2) Write the following method that returns a new ArrayList. The new list contains the non-duplicate elements from the original list.

public static <E>ArrayList<E>removeDuplicates(ArrayList<E> list)

Ans:

**Program:**

package Exp\_10;

import java.util.ArrayList;

import java.util.LinkedHashSet;

public class Second {

    public static <E> ArrayList<E> removeDuplicates(ArrayList<E> list) {

        return new ArrayList<>(new LinkedHashSet<>(list));

    }

    public static void main(String[] args) {

        ArrayList<Integer> intList = new ArrayList<>();

        intList.add(1);

        intList.add(2);

        intList.add(2);

        intList.add(3);

        intList.add(1);

        intList.add(4);

        System.out.println("Original Integer List: " + intList);

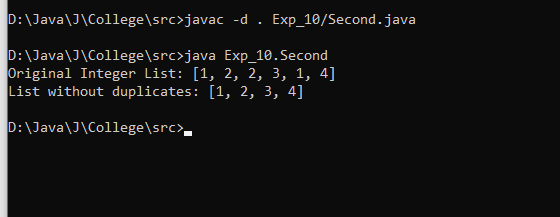
        ArrayList<Integer> noDupIntList = removeDuplicates(intList);

        System.out.println("List without duplicates: " + noDupIntList);

    }

}

**Output:**



Q3) Implement the following method using binary search.

public static <E extends Comparable<E>>

intbinarySearch(E list, E key)

Ans:

**Program:**

package Exp\_10;

import java.util.Arrays;

public class Third {

    public static int binarySearch(int[] list, int key) {

        int low = 0;

        int high = list.length - 1;

        while (low <= high) {

            int mid = (low + high) / 2;

            if (key == list[mid]) {

                return mid;

            } else if (key < list[mid]) {

                high = mid - 1;

            } else {

                low = mid + 1;

            }

        }

        return -1;

    }

    public static void main(String[] args) {

        int[] array = {2, 4, 6, 8, 10, 12, 14, 16};

        System.out.println("Array: " + Arrays.toString(array));

        int key1 = 8;

        int result1 = binarySearch(array, key1);

        System.out.println("Searching for " + key1 + ": Index = " + result1);

        int key2 = 9;

        int result2 = binarySearch(array, key2);

        System.out.println("Searching for " + key2 + ": Index = " + result2);

        int key3 = 2;

        int result3 = binarySearch(array, key3);

        System.out.println("Searching for " + key3 + ": Index = " + result3);

        int key4 = 16;

        int result4 = binarySearch(array, key4);

        System.out.println("Searching for " + key4 + ": Index = " + result4);

    }

}

**Output:**

