NEW YORK CITY COLLEGE OF TECHNOLOGY THE CITY UNIVERSITY OF NEW YORK
Department of Computer Engineering Technology *300 Jay Street, Brooklyn, NY 11201-1909*

LAB REPORT

CET 3640 – OL30

(SOFTWARE FOR COMPUTER CONTROL)

LAB#8

JAVA PROGRAM SORTING

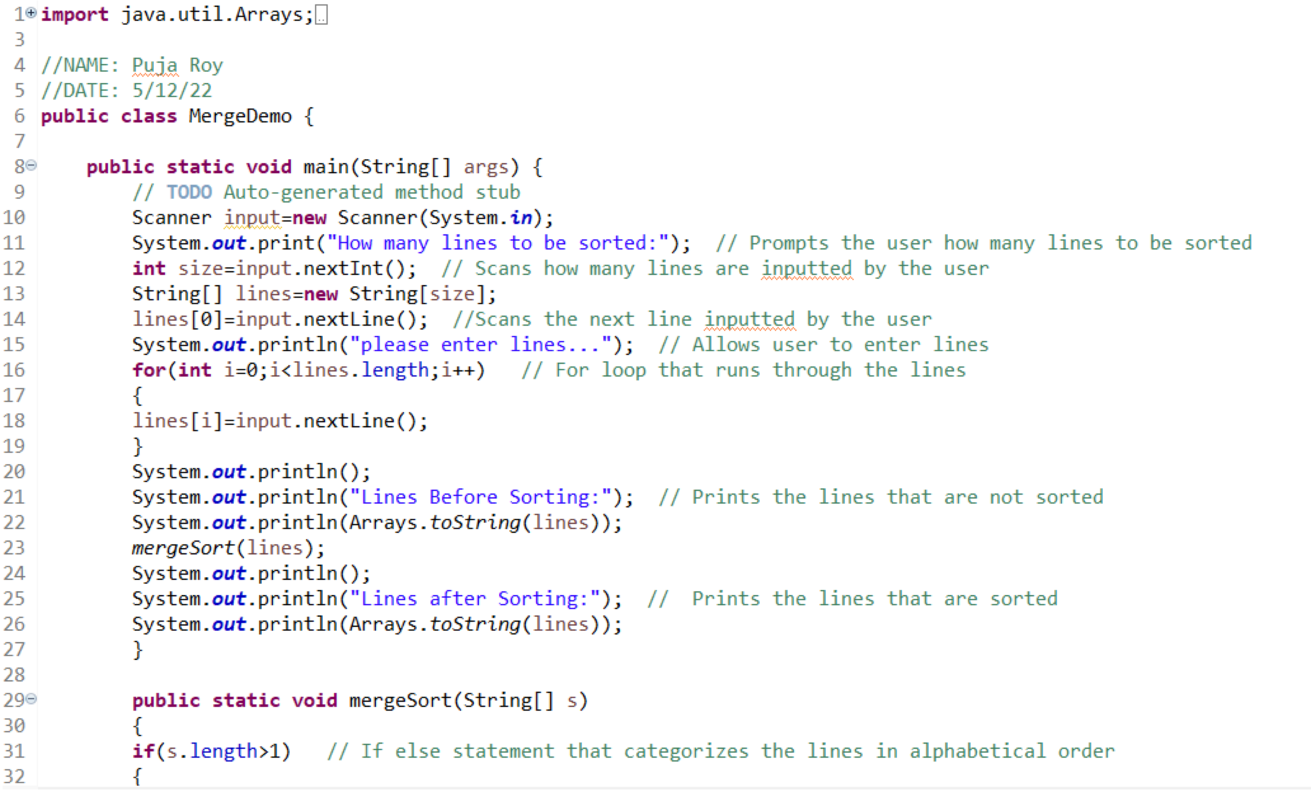
Name: Puja Roy

Date: 5/12/22

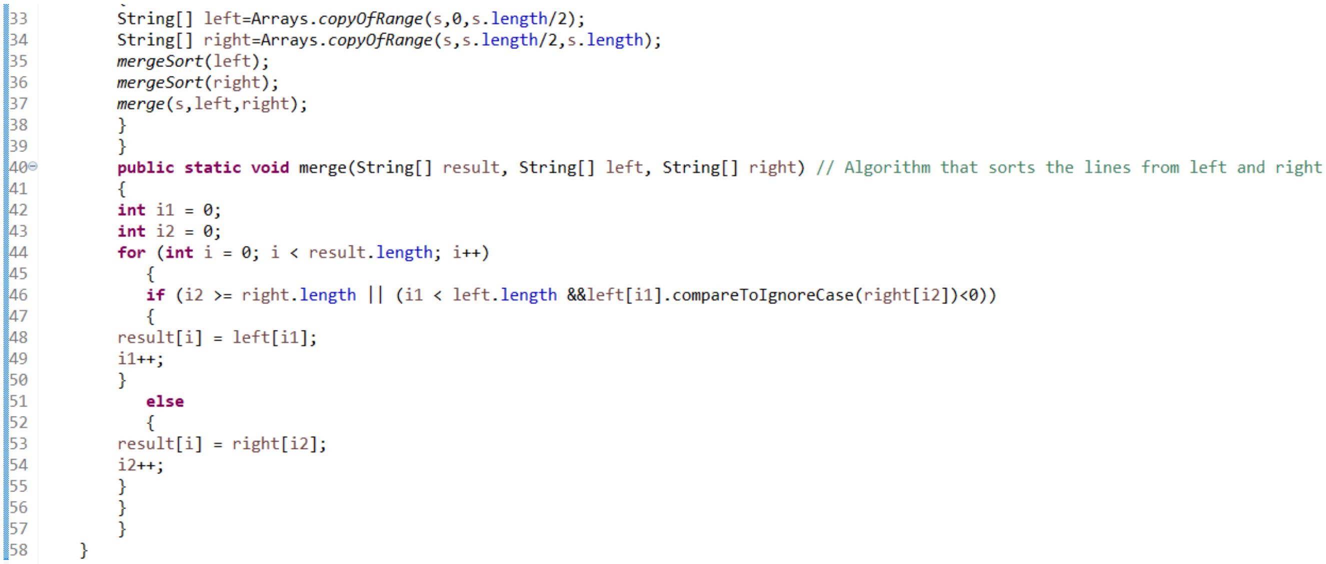
Due Date: 5/14/22

DESCRIPTION OF THE LAB:

In this lab, I wrote a java program in Eclipse that reads and scans a series of lines and then prints the output of the series of lines in alphabetical order. I also wrote a program that includes the merge sort algorithm that sorts lines of large files. First, I created a class called MergeDemo with a main file. Then, I wrote Java code that prompts the user how many lines that they want to enter for the program to sort alphabetically. The Java program includes a for loop and if else statement that runs through the lines to automatically sort the lines in alphabetical order. After the user enters the lines, the program executes a list of lines that were sorted before and then prints the lines that are sorted alphabetically.

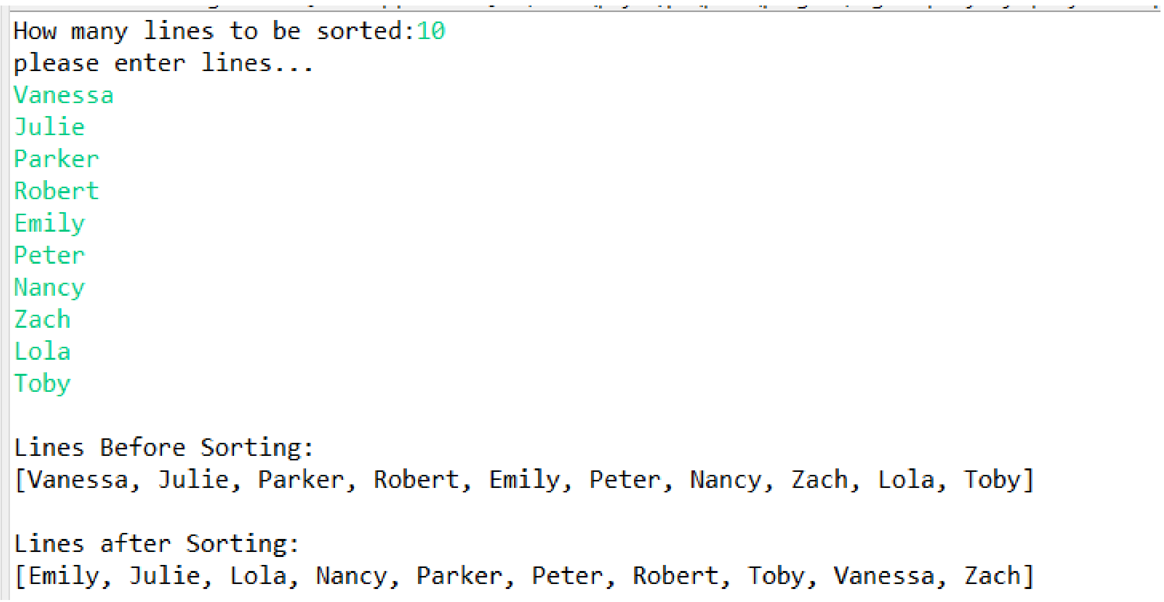


```
import java.util.Arrays;

//NAME: Puja Roy
//DATE: 5/12/22
public class MergeDemo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner input=new Scanner(System.in);
        System.out.print("How many lines to be sorted:");  // Prompts the user how many lines to be sorted
        int size=input.nextInt();  // Scans how many lines are inputted by the user
        String[] lines=new String[size];
        lines[0]=input.nextLine();  //Scans the next line inputted by the user
        System.out.println("please enter lines...");  // Allows user to enter lines
        for(int i=0;i<lines.length;i++)   // For loop that runs through the lines
        {
        lines[i]=input.nextLine();
        }
        System.out.println();
        System.out.println("Lines Before Sorting:");  // Prints the lines that are not sorted
        System.out.println(Arrays.toString(lines));
        mergeSort(lines);
        System.out.println();
        System.out.println("Lines after Sorting:");  //  Prints the lines that are sorted
        System.out.println(Arrays.toString(lines));
        }

        public static void mergeSort(String[] s)
        {
        if(s.length>1)   // If else statement that categorizes the lines in alphabetical order
        {
```



```
        String[] left=Arrays.copyOfRange(s,0,s.length/2);
        String[] right=Arrays.copyOfRange(s,s.length/2,s.length);
        mergeSort(left);
        mergeSort(right);
        merge(s,left,right);
        }
        }
        public static void merge(String[] result, String[] left, String[] right) // Algorithm that sorts the lines from left and right
        {
        int i1 = 0;
        int i2 = 0;
        for (int i = 0; i < result.length; i++)
           {
           if (i2 >= right.length || (i1 < left.length &&left[i1].compareToIgnoreCase(right[i2])<0))
           {
        result[i] = left[i1];
        i1++;
        }
           else
           {
        result[i] = right[i2];
        i2++;
        }
        }
        }
    }
```

Output:



```
How many lines to be sorted:10
please enter lines...
Vanessa
Julie
Parker
Robert
Emily
Peter
Nancy
Zach
Lola
Toby

Lines Before Sorting:
[Vanessa, Julie, Parker, Robert, Emily, Peter, Nancy, Zach, Lola, Toby]

Lines after Sorting:
[Emily, Julie, Lola, Nancy, Parker, Peter, Robert, Toby, Vanessa, Zach]
```