Q1) Static vs Dynamic Array

A) Static array has fixed size and dynamic array can grow or shrink its size.

While creating Static array we need to initialize the size of it.

Int[] arr = new int[5];

Whereas we don’t need to initialize any size in dynamic array.

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

Q2) String reverse

public class Main {

    public static void main(String[] args) {

        String s = "Hello";

        System.out.println(s);

        String res = rev(s);

        System.out.println(res);

        System.out.println(new StringBuilder(s).reverse().toString());

    }

    public static String rev(String s) {

        int l = 0;

        int r = s.length() - 1;

        char[] ch = s.toCharArray();

        while (l <= r) {

            char temp = ch[l];

            ch[l] = ch[r];

            ch[r] = temp;

            l++;

            r--;

        }

        return new String(ch);

    }

}

Q3) Merge Sort

import java.util.Arrays;

public class Main {

    public static void main(String[] args) {

        int[] arr1 = {1,3,5};

        int[] arr2 = {2,4,6,8};

        int[] res = mSort(arr1, arr2);

        System.out.println(Arrays.toString(res));

    }

    public static int[] mSort(int[] arr1, int[] arr2) {

        int[] res = new int[arr1.length + arr2.length];

        int i = 0, j = 0, k = 0;

        while (i < arr1.length && j < arr2.length) {

            if (arr1[i] < arr2[j]) {

                res[k] = arr1[i];

                i++;

            } else {

                res[k] = arr2[j];

                j++;

            }

            k++;

        }

        while (i < arr1.length) {

            res[k] = arr1[i];

            i++;

            k++;

        }

        while (j < arr2.length) {

            res[k] = arr2[j];

            j++;

            k++;

        }

        return res;

    }

}