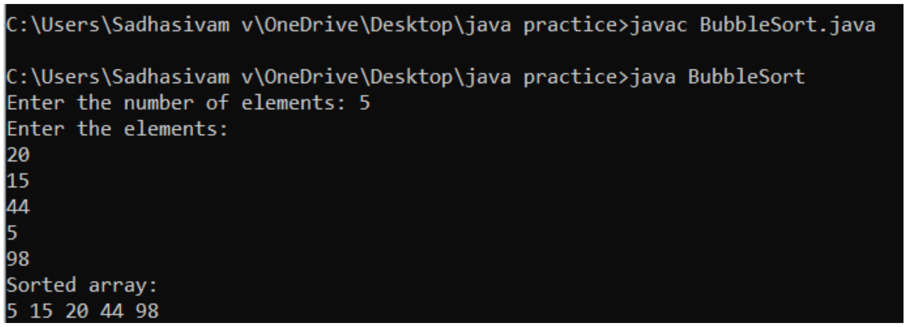
18/11/24 DSA(practice-6)

1.BubbleSort:

```
import java.util.Scanner;
public class BubbleSort {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] array = new int[n];
        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            array[i] = scanner.nextInt();
        }
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
        System.out.println("Sorted array:");
        for (int num : array) {
            System.out.print(num + " ");
        }
    }
}
```

Output:

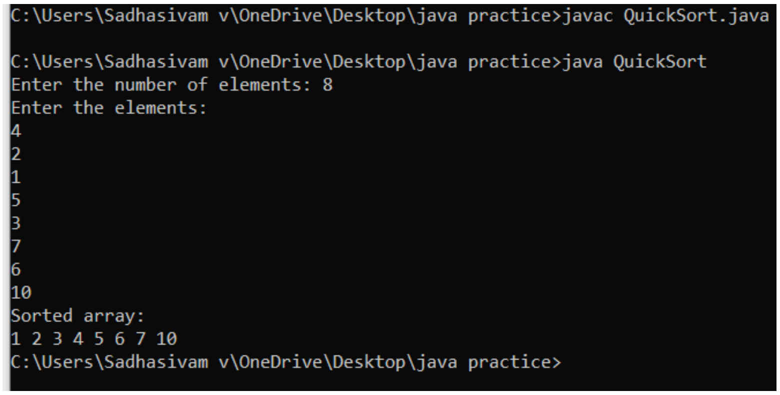


```
C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>javac BubbleSort.java

C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>java BubbleSort
Enter the number of elements: 5
Enter the elements:
20
15
44
5
98
Sorted array:
5 15 20 44 98
```

2.Quicksort:

```
import java.util.Scanner;
public class QuickSort {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] array = new int[n];
        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            array[i] = scanner.nextInt();
        }
        quickSort(array, 0, n - 1);
        System.out.println("Sorted array:");
        for (int num : array) {
            System.out.print(num + " ");
        }
    }
    public static void quickSort(int[] array, int low, int high) {
        if (low < high) {
            int pivot = partition(array, low, high);
            quickSort(array, low, pivot - 1);
            quickSort(array, pivot + 1, high);
        }
    }
    public static int partition(int[] array, int low, int high) {
        int pivot = array[high];
        int i = low;
        for (int j = low; j < high; j++) {
            if (array[j] < pivot) {
                int temp = array[i];
                array[i] = array[j];
                array[j] = temp;
                i++;
            }
        }
        int temp = array[i];
        array[i] = array[high];
        array[high] = temp;
        return i;
    }
}
```

Output

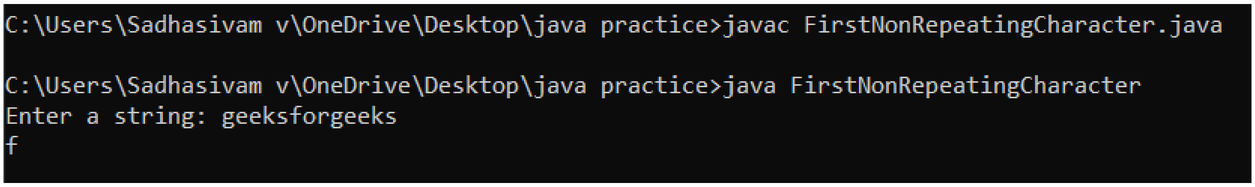


```
C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>javac QuickSort.java

C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>java QuickSort
Enter the number of elements: 8
Enter the elements:
4
2
1
5
3
7
6
10
Sorted array:
1 2 3 4 5 6 7 10
C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>
```

3.FirstNonRepeatingCharacter:

```
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Scanner;
public class FirstNonRepeatingCharacter {
    public static char firstNonRepeating(String s) {
        Map<Character, Integer> charCount = new LinkedHashMap<>();
        for (char ch : s.toCharArray()) {
            charCount.put(ch, charCount.getOrDefault(ch, 0) + 1);
        }
        for (char ch : s.toCharArray()) {
            if (charCount.get(ch) == 1) {
                return ch;
            }
        }
        return '$';
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = scanner.nextLine();
        char result = firstNonRepeating(s);
        if (result == '$') {
            System.out.println(-1);
        } else {
            System.out.println(result);
        }

        scanner.close();
    }
}
```

Output:

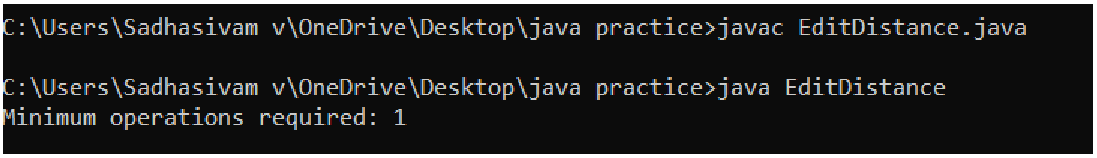


```
C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>javac FirstNonRepeatingCharacter.java

C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>java FirstNonRepeatingCharacter
Enter a string: geeksforgeeks
f
```

4.EditDistance:

```
public class EditDistance {
    public static int minEditDistance(String s1, String s2) {
        int m = s1.length();
        int n = s2.length();
        int[][] dp = new int[m + 1][n + 1];
        for (int i = 0; i <= m; i++) {
            for (int j = 0; j <= n; j++) {
                if (i == 0) {
                    dp[i][j] = j;
                }
                else if (j == 0) {
                    dp[i][j] = i;
                }
                else if (s1.charAt(i - 1) == s2.charAt(j - 1)) {
                    dp[i][j] = dp[i - 1][j - 1];
                }
                else {
                    dp[i][j] = 1 + Math.min(dp[i - 1][j - 1],
                            Math.min(dp[i - 1][j],
                                    dp[i][j - 1]));
                }
            }
        }

        return dp[m][n];
    }
    public static void main(String[] args) {
        String s1 = "geek";
        String s2 = "gesek";
        System.out.println("Minimum operations required: " + minEditDistance(s1, s2));
    }
}
```

Output:

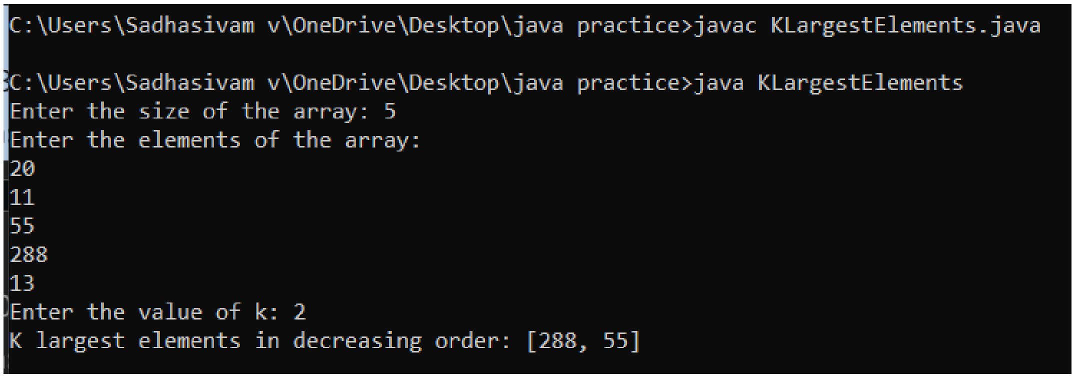


```
C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>javac EditDistance.java

C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>java EditDistance
Minimum operations required: 1
```

5.KLargestElements:

```
import java.util.*;
public class KLargestElements {
    public static List<Integer> kLargest(int[] arr, int k) {
        PriorityQueue<Integer> minHeap = new PriorityQueue<>();
        for (int num : arr) {
            minHeap.add(num);
            if (minHeap.size() > k) {
                minHeap.poll();
            }
        }
        List<Integer> result = new ArrayList<>(minHeap);
        result.sort(Collections.reverseOrder());
        return result;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        System.out.print("Enter the value of k: ");
        int k = scanner.nextInt();
        if (k > n || k <= 0) {
            System.out.println("Invalid value of k. It should be between 1 and " + n);
            return;
        }
        List<Integer> result = kLargest(arr, k);
        System.out.println("K largest elements in decreasing order: " + result);
        scanner.close();
    }
}
```

Output:

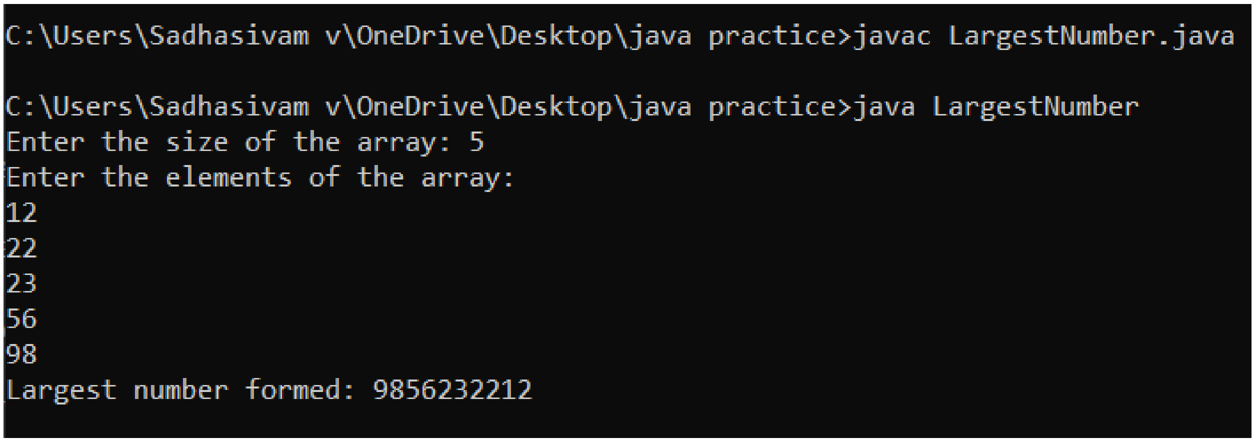


```
C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>javac KLargestElements.java

C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>java KLargestElements
Enter the size of the array: 5
Enter the elements of the array:
20
11
55
288
13
Enter the value of k: 2
K largest elements in decreasing order: [288, 55]
```

6.LargestNumber:

```
import java.util.*;
public class LargestNumber {
    public static String largestNumber(int[] arr) {
        String[] strArr = Arrays.stream(arr)
                                .mapToObj(String::valueOf)
                                .toArray(String[]::new);
        Arrays.sort(strArr, (a, b) -> (b + a).compareTo(a + b));
        if (strArr[0].equals("0")) {
            return "0";
        }
        return String.join("", strArr);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        String result = largestNumber(arr);
        System.out.println("Largest number formed: " + result);

        scanner.close();
    }
}
```

Output:



```
C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>javac LargestNumber.java

C:\Users\Sadhasivam v\OneDrive\Desktop\java practice>java LargestNumber
Enter the size of the array: 5
Enter the elements of the array:
12
22
23
56
98
Largest number formed: 9856232212
```