Name: Ramanarayanan k 18:11:2024

22IT083

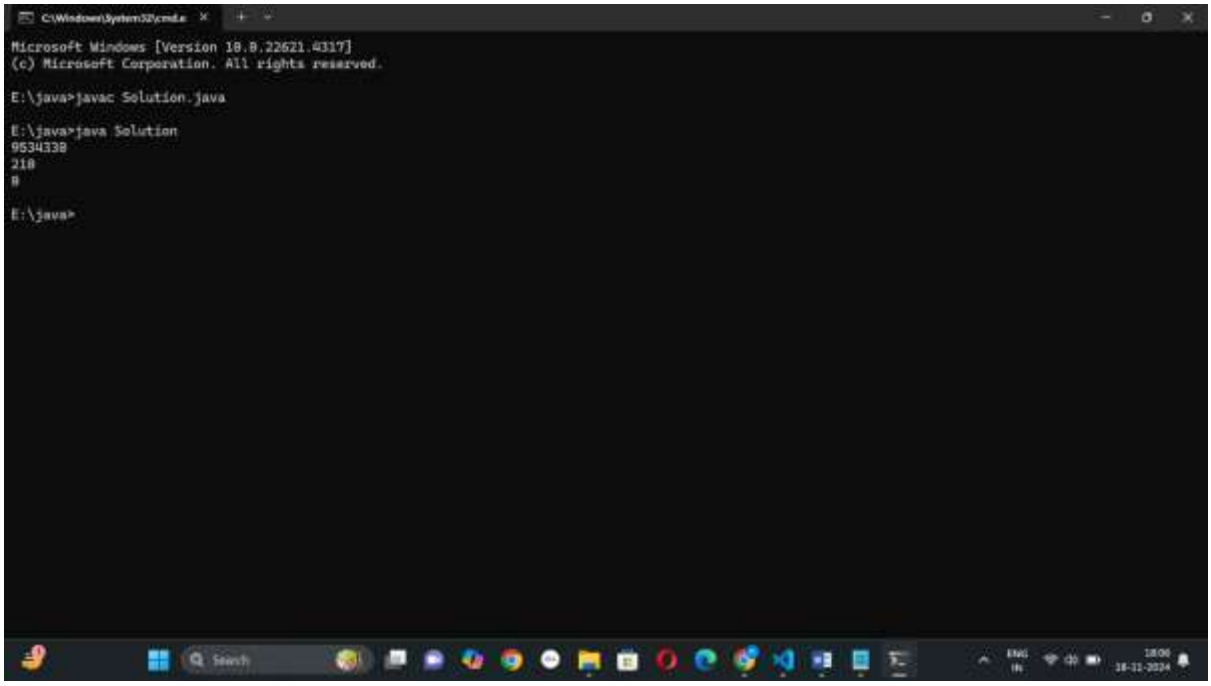
1) Form the largest number

```
import java.util.Arrays;

class Solution {
    String printLargest(String[] arr) {
        Arrays.sort(arr, (a, b) -> (b + a).compareTo(a + b));
        StringBuilder res = new StringBuilder();
        for (String s : arr) {
            res.append(s);
        }
        return res.charAt(0) == '0' ? "0" : res.toString();
    }

    public static void main(String[] args) {
        String[] arr = {"3", "30", "34", "5", "9"};
        Solution sol = new Solution();
        System.out.println(sol.printLargest(arr));

        String[] arr2 = {"10", "2"};
        System.out.println(sol.printLargest(arr2));

        String[] arr3 = {"0", "0"};
```

```
        System.out.println(sol.printLargest(arr3));
    }
}
```

OUTPUT:

TC:O(n)



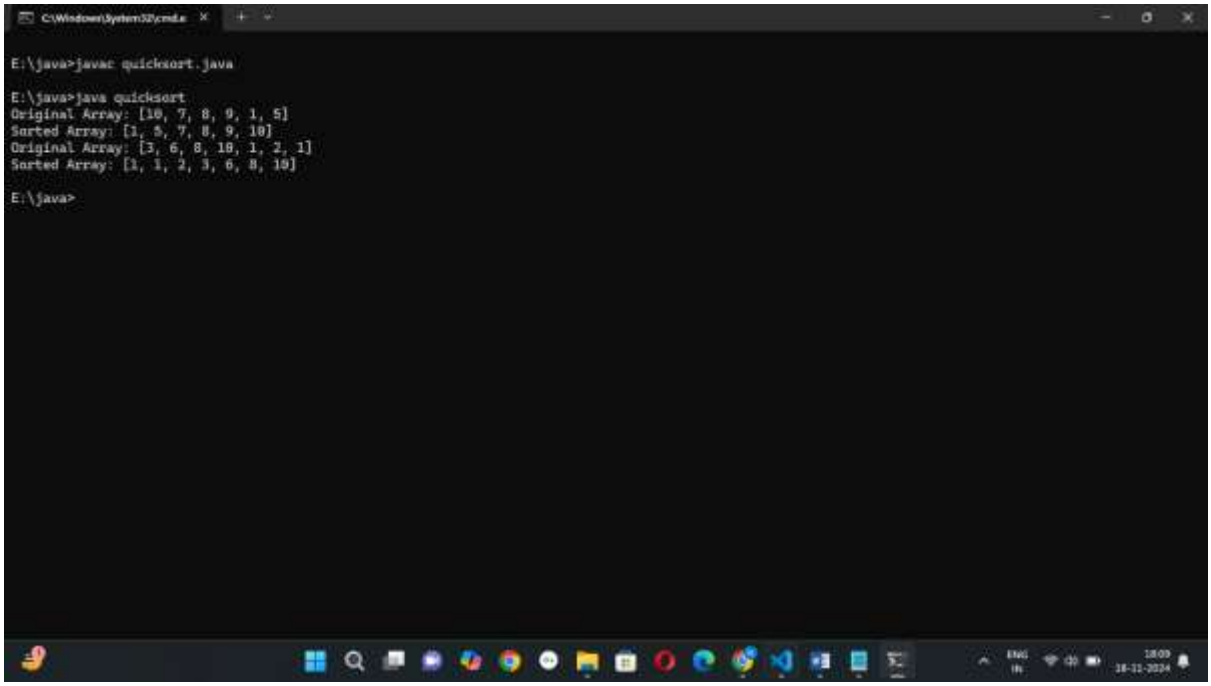
2) Ouicksort

```
import java.util.Arrays;

class quicksort {
    // Function to sort an array using quick sort algorithm.
    static void quickSort(int arr[], int low, int high) {
        if (low < high) {
```

```
        int pivat = partition(arr, low, high);
        quickSort(arr, low, pivat - 1);
        quickSort(arr, pivat + 1, high);
    }
}

static int partition(int arr[], int low, int high) {
    int pivat = arr[low];
    int i = low;
    int j = high;
    while (i < j) {
        while (i <= high && arr[i] <= pivat) {
            i++;
        }
        while (j >= low && arr[j] > pivat) {
            j--;
        }
        if (i < j) {
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
    arr[low] = arr[j];
    arr[j] = pivat;
    return j;
```

```
    }

    public static void main(String[] args) {
        int[] arr = {10, 7, 8, 9, 1, 5};
        System.out.println("Original Array: " + Arrays.toString(arr));
        quickSort(arr, 0, arr.length - 1);
        System.out.println("Sorted Array: " + Arrays.toString(arr));

        int[] arr2 = {3, 6, 8, 10, 1, 2, 1};
        System.out.println("Original Array: " + Arrays.toString(arr2));
        quickSort(arr2, 0, arr2.length - 1);
        System.out.println("Sorted Array: " + Arrays.toString(arr2));
    }
}
```

OUTPUT: TC:o(nlogn)



3) NonRepeated character first

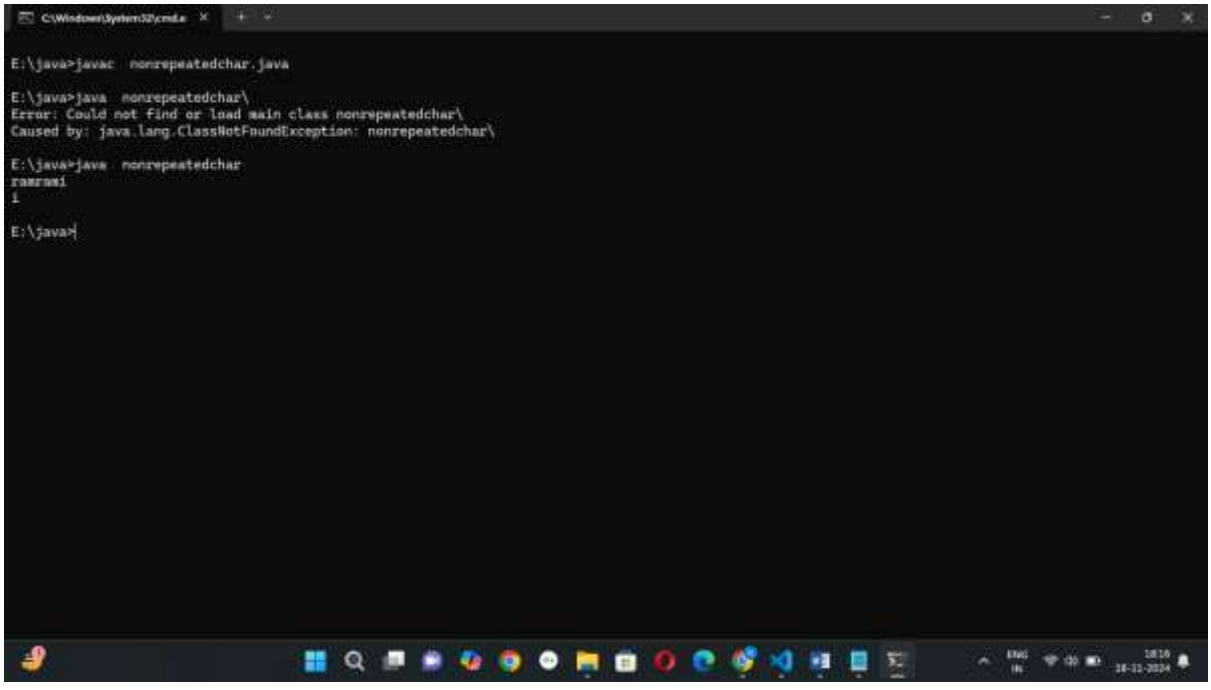
Code:

```
import java.util.*;

class nonrepeatedchar {
   static char nonRepeatingChar(String s) {
      HashMap<Character, Integer> hash = new HashMap<>();

      for (char ch : s.toCharArray()) {
         hash.put(ch, hash.getOrDefault(ch, 0) + 1);
      }

      for (char ch : s.toCharArray()) {
         if (hash.get(ch) == 1) {
            return ch;
         }
      }

      return '$';
   }

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      String s = sc.nextLine();
```

```
        char result = nonRepeatingChar(s);
        if (result != '$') {
            System.out.println(result);
        } else {
            System.out.println("$");
        }

        sc.close();
    }
}
```

Output:

TC:O(n)



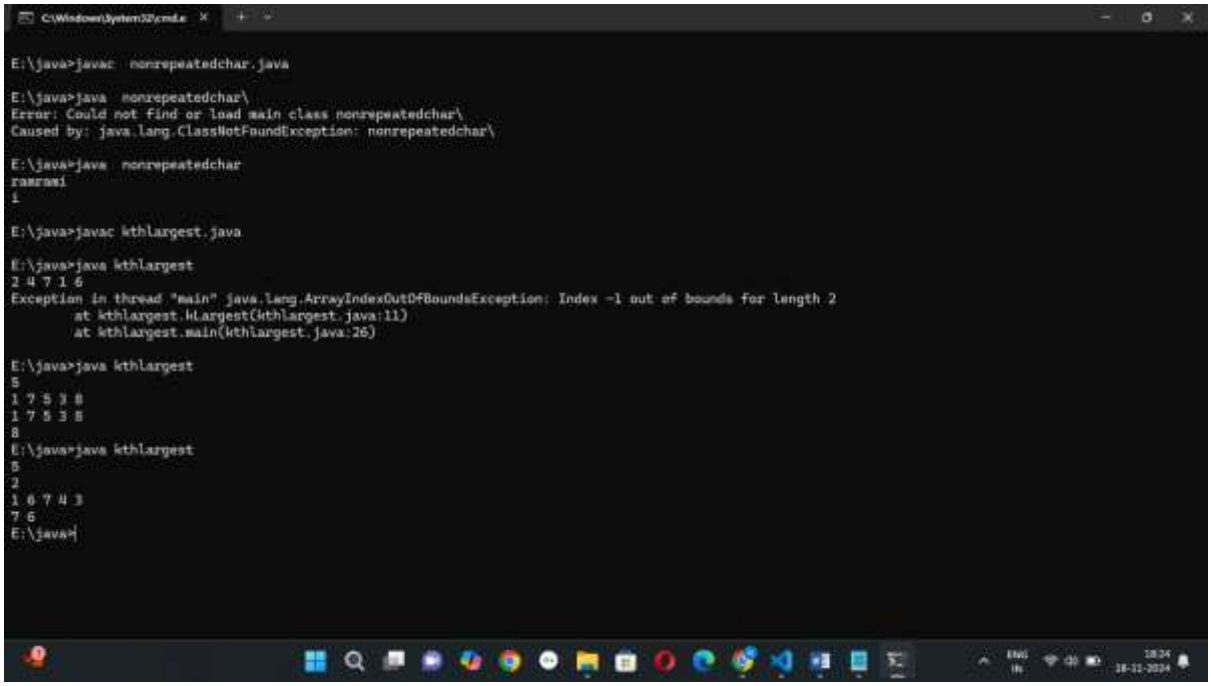
4) Kth largest number

```
import java.util.*;

class kthlargest{


   static List<Integer> kLargest(int arr[], int k) {
      int n = arr.length;
      List<Integer> ls = new ArrayList<>(k);
      Arrays.sort(arr);
      for (int i = n - 1; i >= n - k; i--) {
         ls.add(arr[i]);
      }
      return ls;
   }

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      int n = sc.nextInt();
      int k = sc.nextInt();
      int[] arr = new int[n];
      for (int i = 0; i < n; i++) {
         arr[i] = sc.nextInt();
      }
```

```
        List<Integer> result = kLargest(arr, k);
        for (int num : result) {
            System.out.print(num + " ");
        }

        sc.close();
    }
}
```

Output:

TC:O(n)



5) Bubblesort array

```
import java.util.*;

class bubblesort{
```

```
public static void bubbleSort(int arr[]) {
    int n = arr.length;
    for (int i = n - 1; i > 0; i--) {
        for (int j = 0; j <= i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j + 1];
                arr[j + 1] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    int n = sc.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = sc.nextInt();
    }

    bubbleSort(arr);

    for (int num : arr) {
```

```
            System.out.print(num + " ");
        }


        sc.close();
    }
}
```

Output:

TC:O(n2)