Q1) Maximum Subarray Sum:

import java.util.Arrays;

public class Main {

    public static void main(String[] args) {

        int[] arr1 = {-2,1,-3,4,-1,2,1,-5,4};

        System.out.println(maxSumSubArr1(arr1));

        System.out.println(maxSumSubArr2(arr1));

    }

   public static int maxSumSubArr1(int[] arr) {

       int cur = arr[0];

       int sum = arr[0];

       for (int i = 0; i < arr.length; i++) {

           sum = Math.max(arr[i], sum + arr[i]);

           cur = Math.max(cur, sum);

       }

       return cur;

   }

    public static int maxSumSubArr2(int[] arr) {

       int cur = arr[0];

       int sum = 0;

       for (int i : arr) {

           sum += i;

           if (sum > cur) cur = sum;

           if (sum < 0) sum = 0;

       }

       return cur;

   }

}

Q2) Remove Duplicates in Array

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        int[] arr1 = {1,1,2,3,3,3,5};

        System.out.println(Arrays.toString(removeDup(arr1)));

    }

    public static int[] removeDup(int[] arr) {

        Set<Integer> set = new LinkedHashSet<>();

        for (int i : arr) {

            set.add(i);

        }

        int[] res = new int[set.size()];

        int index = 0;

        for (int i : set) {

            res[index++] = i;

        }

        return res;

    }

}

Q3) Rotate Array

import java.util.Arrays;

public class Main {

    public static void main(String[] args) {

        int[] arr = {1,2,3,4,5};

        rotateArray(arr, 3);

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

    }

    public static void rotateArray(int[] arr, int k) {

int n = arr.length;

k = k % n;

reverseArray(arr, 0, n - 1);

reverseArray(arr, 0, k - 1);

reverseArray(arr, k, n - 1);

}

private static void reverseArray(int[] arr, int left, int right) {

while (left < right) {

int temp = arr[left];

arr[left] = arr[right];

arr[right] = temp;

left++;

right--;

}

}

}