1.// Maximum and Minimum Element in an Array

import java.util.\*;

public class prog1{

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the size of array: ");

        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.println("Enter element: ");

        for(int i = 0; i < n; i++){

            arr[i] = sc.nextInt();

        }

        int min = Integer.MAX\_VALUE;

        int max = Integer.MIN\_VALUE;

        for(int i = 0; i < n; i++){

            if(arr[i] < min){

                min = arr[i];

            }

            else{

                max = arr[i];

            }

        }

        System.out.println("Minimum element: "+min);

        System.out.println("Maximum element: "+max);

    }

}

2.//Reverse the Array

import java.util.\*;

public class prog2 {

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the size of array: ");

        int n = sc.nextInt();

        int arr[] = new int[n];

        int reverse\_arr[] = new int[n];

        System.out.println("Enter element: ");

        for(int i = 0; i < n; i++){

            arr[i] = sc.nextInt();

        }

        for(int i = 0; i < n; i++){

            reverse\_arr[i] = 0;

        }

        int p = 0;

        for(int i = n-1; i >= 0; i--){

            reverse\_arr[p] = arr[i];

            p++;

        }

        for(int i = 0; i < n; i++){

            System.out.print(reverse\_arr[i]+" ");

        }

    }

}

3.// Maximum-Subarray sum

import java.util.\*;

public class prog3 {

        public static int maxSubArray(int[] nums) {

            int n = nums.length;

            int sum = 0;

            int max = Integer.MIN\_VALUE;

            for(int i = 0; i < n; i++){

                sum+= nums[i];

                max = Math.max(sum,max);

                if(sum<0) sum = 0;

            }

            return max;

        }

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        int arr[] = {5,4,-1,7,8};

        System.out.print(maxSubArray(arr));

    }

}

4.//Contains Duplicate

import java.util.Arrays;

import java.util.\*;

public class prog4 {

    public static boolean contain\_duplicate(int nums[]){

        Arrays.sort(nums);

        for(int i = 0; i < nums.length - 1; i++){

            if(nums[i]==nums[i+1]){

                return true;

            }

        }

        return false;

    }

    public static void main(String args[]){

        int nums[] = {1,2,3,1};

        System.out.print(contain\_duplicate(nums));

    }

}

5. // Chocolate Distribution Problem O(nlogn)

import java.util.\*;

import java.util.Arrays;

public class prog5 {

    public static int chocolateProblem(int arr[], int k){

        Arrays.sort(arr);

        int min = Integer.MAX\_VALUE;

        for(int i = 0; i+k-1 < arr.length; i++){

            int diff = arr[i+k-1] - arr[i];

            if(diff < min){

                min = diff;

            }

        }

        return min;

    }

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the size of array: ");

        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.println("Enter the number of students: ");

        int m = sc.nextInt();

        System.out.println("Enter element: ");

        for(int i = 0; i < n; i++){

            arr[i] = sc.nextInt();

        }

        System.out.print(chocolateProblem(arr, m));

    }

}

6. // Search in Rotated Sorted Array

import java.util.\*;

public class prog6 {

    public static int rotatedArray(int arr[], int key){

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

            if(key == arr[i]){

            return i;

        }

    }

    return -1;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the size of array: ");

        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.println("Enter the target to search: ");

        int m = sc.nextInt();

        System.out.println("Enter element: ");

        for(int i = 0; i < n; i++){

            arr[i] = sc.nextInt();

        }

        System.out.println(rotatedArray(arr, m));

    }

}

7. //Next Permutation

import java.util.\*;

import java.util.Arrays;

public class prog7 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the size of array: ");

        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.println("Enter elements: ");

        for (int i = 0; i < n; i++) {

            arr[i] = sc.nextInt();

        }

        int ind = -1;

        for (int i = n - 1; i > 0; i--) {

            if (arr[i - 1] < arr[i]) {

                ind = i - 1;

                break;

            }

        }

        if (ind != -1) {

            int temp = arr[ind];

            arr[ind] = arr[ind + 1];

            arr[ind + 1] = temp;

            int start = ind + 1;

            int end = n - 1;

            while (start < end) {

                int t = arr[start];

                arr[start] = arr[end];

                arr[end] = t;

                start++;

                end--;

            }

        } else {

            Arrays.sort(arr);

        }

        for (int i = 0; i < n; i++) {

            System.out.print(arr[i] + " ");

        }

    }

}

8. //Best Time to Buy and Sell Stock

import java.util.\*;

class prog8 {

    public static int maxProfit(int[] prices) {

        int max = Integer.MAX\_VALUE;

        int min = prices[0];

        int profit = 0;

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

        int cost = prices[i] - min;

        profit = Math.max(profit,cost);

        min = Math.min(min,prices[i]);

    }

    return profit;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int prices[] = {7,1,5,3,6,4};

        System.out.println(maxProfit(prices));

    }

}