**As Easy As A+B**

package com.my;

import java.util.Arrays;

import java.util.Random;

import java.util.Scanner;

public class AsEasyAsAB {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int n = input.nextInt();

        for(int i = 0;i<n;i++){

            int arrNum = input.nextInt();

            int[] arr = new int[arrNum];

            for(int j=0;j<arrNum;j++){

                arr[j]=input.nextInt();

            }

            quickSort(arr,0,arr.length-1);

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

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

            }

            System.out.println();

        }

    }

    //返回基准元素的位置

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

        int partition = left;

        int partitionNum = arr[partition];

        int l = left;

        int r = right;

        while (l!=r){

            while ((l<r)&&(arr[r]>partitionNum)){

                r--;

            }

            while ((l<r)&&(arr[l]<=partitionNum)){

                l++;

            }

            if(l<r){

                int tmp = arr[l];

                arr[l] = arr[r];

                arr[r] =tmp;

            }

        }

       arr[partition]=arr[l];

       arr[l] = partitionNum;

       return l;

    }

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

        if(left>=right){

            return;

        }

        int p = selectpartition(arr,left,right);

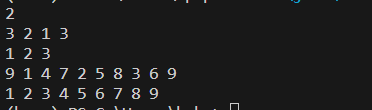
        quickSort(arr,left,p-1);

        quickSort(arr,p+1,right);

    }

}

运行结果



**Calculate e**

package com.my;

import java.util.Arrays;

public class CalculateE {

    public static void main(String[] args) {

        System.out.println("n e");

        System.out.println("- -----------");

        int e0=1;

        int e1=2;

        double[] result = new double[8];

        result[0] = 2.5;

        for(int i=3;i<=9;i++){

            result[i-2] = result[i-3] + cal(i);

        }

        System.out.println("0 "+e0);

        System.out.println("1 "+e1 );

        System.out.println("2 "+result[0]);

        for(int i=1;i<8;i++){

            System.out.print((i+2)+" ");

            String out = String.format("%.9f",result[i]);

            System.out.println(out);

        }

    }

    private static double cal(int num){

        int result = 1;

        for(int i=1;i<=num;i++){

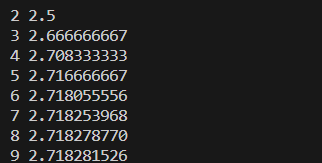
            result = result \* i;

        }

        return 1.0/(double)result;

    }

}



# Elevator

package com.my;

import java.util.Scanner;

public class Elevator {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        while(true){

            int n = input.nextInt();

            if(n == 0){

                break;

            }

            int[] arr = new int[n];

            for(int i=0;i<n;i++){

                arr[i] = input.nextInt();

            }

            int start = 0;

            int time = 0;

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

                if(start<arr[i]){

                    time = time + 5 + (arr[i]-start)\*6;

                    start = arr[i];

                }

                else {

                    time = time + 5 + (start-arr[i])\*4;

                    start = arr[i];

                }

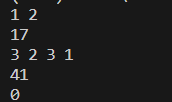
            }

            System.out.println(time);

        }

    }

}



**Max Sum**

package com.my;

import java.util.Arrays;

import java.util.Scanner;

public class MaxSum {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int n = input.nextInt();

        for(int i=0;i<n;i++){

            int num = input.nextInt();

            int[] arr = new int[num];

            for(int j=0;j<num;j++){

                arr[j] = input.nextInt();

            }

            int[] dp = new int[num];

            dp[0] = arr[0];

            int start = 1;

            int end = 0;

            for(int j = 1;j<arr.length;j++){

                if(dp[j-1]+arr[j]>=arr[j]){

                    dp[j] = dp[j-1] + arr[j];

                }

                else {

                    dp[j] = arr[j];

                    start = j+1;

                }

            }

            int out = 0;

            for(int j=0;j<dp.length;j++){

                if(dp[j]>out){

                    end = j+1;

                    out = dp[j];

                }

            }

            System.out.println("Case "+(i+1)+":");

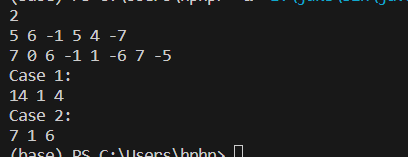
            System.out.print(out);

            System.out.println(" "+start+" "+end);

        }

    }

}



**Number Sequence**

package com.my;

import java.util.Scanner;

public class NumberSequence {

    private static int[] logN= new int[100000000];

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        while(true){

            int A = input.nextInt();

            int B = input.nextInt();

            int n = input.nextInt();

            if(A==0 && B==0 && n==0){

                break;

            }

            System.out.println(findResult(A,B,n));

        }

    }

    private static int findResult(int a, int b, int n) {

        int f3 = (a\*1+b\*1)%7;

        if(n==1||n==2){

            return 1;

        }

        if(n==3){

            return f3;

        }

        int[][] m = new int[2][2];

        m[0][0]=a;

        m[0][1]=b;

        m[1][0]=1;

        m[1][1]=0;

        int[][] fastMatrix = fastMatrix(m,n-2,7);

        return (fastMatrix[0][0] + fastMatrix[0][1])%7;

    }

    private static int[][] calMatrix(int[][] a,int[][] b,int mod){

        int r = a.length;

        int l = b[0].length;

        int kl = a[0].length;

        int[][] res = new int[r][l];

        for(int i=0;i<r;i++){

            for(int j = 0;j<l;j++){

                for(int k=0;k<kl;k++){

                    res[i][j] = (res[i][j] + a[i][k]\*b[k][j])%mod;

                }

            }

        }

        return res;

    }

    private static int[][] fastMatrix(int[][] a,int p,int mod){

        int[][] res = new int[a.length][a.length];

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

            res[i][i] = 1;

        }

        while (p != 0){

            if((p&1)==1){

                res = calMatrix(res,a,mod);

            }

            a=calMatrix(a,a,mod);

            p=p/2;

        }

        return res;

    }

}

