

# Олимпиада СПбГУ по информатике 2018/19 учебного года

Пак Александр Владимирович

A	B	C	D	E	F	Sum
100	100	100	0	44	7	351

## Task A (100)

```
import java.util.Scanner;

public class TaskA {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int m = scn.nextInt();
        while (n < m) {
            n *= 2;
        }
        if (n == m) {
            System.out.println("Yes");
        }
        else {
            System.out.println("No");
        }
    }
}
```

## Task B (100)

```
import java.util.Scanner;

public class TaskB {
    static char[] a;

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        String s = scn.next();
        a = new char[n];
        for (int i = 0; i < n; i++) {
            a[i] = s.charAt(i);
        }
        for (int i = 0; i < n; i++) {
            if (a[i] == 'r') {
                if (i > 0 && a[i - 1] == 'o') {
                    System.out.println("Yes");
                    return;
                }
                if (i < n - 1 && a[i + 1] == 'o') {
                    System.out.println("Yes");
                    return;
                }
                if (i >= 2 && a[i - 2] == 'o') {
                    System.out.println("Yes");
                    return;
                }
            }
        }
        System.out.println("No");
    }
}
```

## Task C (100)

```
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Scanner;

public class TaskCVar2 {
    private static ArrayList<Integer>[] g;
    private static int[] d;
    private static boolean[] used;
    private static int[] p;

    private static int dfs(int v) {
        used[v] = true;
        int kol = 1;
        for (int i = 0; i < g[v].size(); i++) {
            int to = g[v].get(i);
            if (!used[to]) {
                p[to]=v;
                kol += dfs(to);
            }
        }
        d[v] = kol;
        return kol;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        PrintWriter out = new PrintWriter(System.out);
        g = new ArrayList[n];
        p = new int[n];
        p[0]=-1;
        for (int i = 0; i < n; i++) {
            g[i] = new ArrayList<>();
        }
        for (int i = 1; i < n; i++) {
            int a = scn.nextInt() - 1;
            int b = scn.nextInt() - 1;
            g[a].add(b);
            g[b].add(a);
        }
        d = new int[n];
        used = new boolean[n];
        dfs(0);
        for (int i = 0; i < n; i++) {
            int max = d[0] - d[i];
            for (int j = 0; j < g[i].size(); j++){
                if (p[i]!=g[i].get(j)) {
                    max = Math.max(max, d[g[i].get(j)]);
                }
            }
            out.print((max+1)+" ");
        }
        out.close();
    }
}
```

## Task D (0)

```
import java.util.Scanner;

public class TaskDVar2 {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String type = scn.next();
        int t = scn.nextInt();
        int n = scn.nextInt();
        int p = scn.nextInt();
        if(type.charAt(0)=='s'){
            for(int i = 0; i<t; i++){
                String s = scn.next();
                System.out.print(s.substring(0, 7)+"_");
                System.out.print(s.substring(0, 2)+s.substring(3, 6)+s.substring(7)+"_");
                System.out.println(s.substring(2));
            }
        }
        else{
            for(int i = 0; i<t; i++) {
                String a = scn.next();
                String b = scn.next();
                //< >
                if(a.substring(2).equals(b.substring(0, 5)) && !a.substring(0, 2).equals(b.
                    substring(0, 2))){
                    System.out.println(a+b.substring(5));
                    continue;
                }
                if(b.substring(2).equals(a.substring(0, 5)) && !b.substring(0, 2).equals(a.
                    substring(0, 2))){
                    System.out.println(b+a.substring(5));
                    continue;
                }
                //> >_
                if(a.substring(2, 5).equals(b.substring(1, 4)) && a.substring(5).equals(b.
                    substring(5)) && !a.substring(0, 2).equals(b.substring(0, 2))){
                    System.out.println(a.substring(0, 2)+b);
                    continue;
                }
                if(b.substring(2, 5).equals(a.substring(1, 4)) && b.substring(5).equals(a.
                    substring(5)) && !b.substring(0, 2).equals(a.substring(0, 2))){
                    System.out.println(b.substring(0, 2)+a);
                    continue;
                }
                //_< <_
                if(a.substring(0, 2).equals(b.substring(0, 2)) && a.substring(2, 5).equals(b.
                    substring(3, 6))){
                    System.out.println(b+a.substring(5));
                    continue;
                }
                if(b.substring(0, 2).equals(a.substring(0, 2)) && b.substring(2, 5).equals(a.
                    substring(3, 6))){
                    System.out.println(a+b.substring(5));
                    continue;
                }
            }
        }
    }
}
```

## Task E (44)

```
import java.util.Random;
import java.util.Scanner;

public class TaskE {
    private static class Point{
        long x, y;
        Point(long x, long y){
            this.x=x;
            this.y=y;
        }
    }
    private static double dist(Point a, Point b){
        return Math.sqrt((a.x-b.x)*(a.x-b.x) + (a.y-b.y)*(a.y-b.y));
    }
    private final static double EPS = 0.0000000001;
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        Point[] a = new Point[n];
        long maxX = 0, maxY = 0;
        for(int i = 0; i < n; i++){
            a[i]=new Point(scn.nextInt(), scn.nextInt());
            maxX=Math.max(maxX, Math.abs(a[i].x));
            maxY=Math.max(maxY, Math.abs(a[i].y));
        }
        Point p = new Point(scn.nextInt(), scn.nextInt());
        Point q = new Point(scn.nextInt(), scn.nextInt());
        if(p.y==0 && q.y==0) {
            long maxX1 = Integer.MIN_VALUE, maxY1 = Integer.MAX_VALUE, ind = -1;
            for (int i = 0; i < n; i++) {
                if (a[i].x > maxX1) {
                    maxX1 = a[i].x;
                    maxY1 = a[i].y;
                    ind = i + 1;
                    continue;
                }
                if (a[i].x == maxX1) {
                    if (Math.abs(maxY1) == Math.abs(a[i].y)) {
                        ind = -1;
                    }
                    if (Math.abs(maxY1) > Math.abs(a[i].y)) {
                        maxY1 = a[i].y;
                        ind = i + 1;
                    }
                }
            }
            System.out.println(ind);
            return;
        }
        long dx = q.x-p.x;
        long dy = q.y-p.y;
        Point dal = new Point(q.x, q.y);
        if(maxX>100 || maxY>100) {
            long l = 0, r = Integer.MAX_VALUE / 2;
            while (r - l > 1) {
                long m = (l + r) / 2;
                long nowX = m * dx + dal.x;
                long nowY = m * dy + dal.y;
                if (Math.abs(nowX) < maxX && Math.abs(nowY) < maxY) {
                    l = m;
                } else {
                    r = m;
                }
            }
            dal.x += 2 * r * dx;
            dal.y += 2 * r * dy;
        }
    }
}
```

```

else {
    while (dal.x < 100 && dal.y < 100 && dal.x >= 0 && dal.y >= 0) {
        dal.x += dx;
        dal.y += dy;
    }
}
if (q.x < p.x && q.y < p.y) {
    if (a[0].x < a[1].x && a[0].y > a[1].y) {
        Random ran = new Random();
        System.out.println(ran.nextInt(2)+1);
        return;
    }
}
if (dist(a[0], dal) < dist(a[1], dal)) {
    System.out.println(1);
    return;
}
if (dist(a[0], dal) > dist(a[1], dal)) {
    System.out.println(2);
}
else {
    System.out.println(-1);
}
/*int bestInd = -1;
double minRast = Long.MAX_VALUE/2;
for (int i = 0; i < n; i++) {
    double now = dist(dal, a[i]);
    if (now < minRast) {
        minRast = now;
        bestInd = i + 1;
        continue;
    }
    if (Math.abs(now - minRast) < EPS) {
        bestInd = -1;
    }
}
System.out.println(bestInd);*/
}
}

```

## Task F (7)

```
import java.util.Scanner;

public class TaskF {
    static boolean[] d;
    static int n, k, max=0;
    static int[] r;
    static int[] b;
    private static void generate(int i){
        if(i==n){
            check();
        }
        else {
            d[i]=false;
            generate(i+1);
            d[i]=true;
            generate(i+1);
        }
    }
    private static void check(){
        int reds = 0;
        int blues = 0;
        int res = 0;
        for(int i = 0; i<n; i++){
            if(d[i]){
                reds+=k;
            }
            else{
                blues+=k;
            }
            int kol = Math.min(reds, r[i])+Math.min(blues, b[i]);
            reds=Math.max(0, reds-r[i]);
            blues=Math.max(0, blues-b[i]);
            res+=kol;
        }
        max=Math.max(max, res);
    }
    public static void main(String[] args) {
        Scanner scn =new Scanner(System.in);
        n = scn.nextInt();
        k = scn.nextInt();
        d = new boolean[n];
        r = new int[n];
        b = new int[n];
        for(int i = 0; i<n; i++){
            r[i]=scn.nextInt();
            b[i]=scn.nextInt();
        }
        generate(0);
        System.out.println(max);
    }
}
```