Олимпиада СПбГУ по информатике 2018/19 учебного года

Кочетов Николай Павлович

A	B	C	D	E	F	Sum
100	100	100	60	12	7	379

Task A (100)

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

public class Main {

    void solve() {
        int a = readInt();
        int b = readInt();
        if (a > b)
            out.println("No");
        else if (a == b)
            out.println("Yes");
        else if (b % a == 0) {
            int count = b / a;
            count = Integer.bitCount(count);
            if (count == 1 || count == 0)
                out.println("Yes");
            else
                out.println("No");
        } else
            out.println("No");
    }

    Scanner in = new Scanner(System.in);
    PrintWriter out = new PrintWriter(System.out);

    int readInt() {
        return in.nextInt();
    }

    public static void main(String[] args) {
        new Main().run();
    }

    void run() {
        solve();
        out.close();
    }
}
```

Task B (100)

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.Scanner;

public class Main {

    void solve() throws IOException {
        fs.readLine();
        String inp = fs.readLine();
        char[] s = inp.toCharArray();
        for (int i = 0; i < s.length - 2; i++) {
            if (s[i] == 'o' && s[i + 2] == 'r') {
                out.println("Yes");
                return;
            }
        }
        for (int i = 0; i < s.length - 1; i++) {
            if (s[i] == 'r' && s[i + 1] == 'o') {
                out.println("Yes");
                return;
            }
            if (s[i] == 'o' && s[i + 1] == 'r') {
                out.println("Yes");
                return;
            }
        }
        out.println("No");
    }

    BufferedReader fs = new BufferedReader(new InputStreamReader(System.in));
    Scanner in = new Scanner(System.in);
    PrintWriter out = new PrintWriter(System.out);

    int readInt() {
        return in.nextInt();
    }

    public static void main(String[] args) throws IOException {
        new Main().run();
    }

    void run() throws IOException {
        solve();
        out.close();
    }
}
```

Task C (100)

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.Arrays;
import java.util.StringTokenizer;

public class Main {

    int[][] g;
    int[] ms;
    int n;
    int[] maxx;
    int[] from;

    void startdfs(int start) {
        Arrays.fill(ms, -1);
        dfs(start);
    }

    int dfs(int v) {
        ms[v] = 0;
        for (int i = 0; i < g[v].length; i++) {
            int vn = g[v][i];
            if (ms[vn] == -1) {
                ms[v] += dfs(vn) + 1;
            }
        }

        return ms[v];
    }

    void stupid() {
        for (int i = 0; i < n; i++) {
            if (g[i].length == 1) {
                out.print(n + " ");
                continue;
            }
            startdfs(i);
            int max = Integer.MIN_VALUE;
            for (int j = 0; j < n; j++) {
                if (j == i)
                    continue;
                if (ms[j] > max)
                    max = ms[j];
            }
            out.print((max + 2) + " ");
        }
    }

    void startSmartDfs(int start) {
        Arrays.fill(ms, -1);
        dfsSm(start, -1);
    }

    int dfsSm(int v, int par) {
        from[v] = par;
        ms[v] = 1;
        for (int i = 0; i < g[v].length; i++) {
            int vn = g[v][i];
            if (ms[vn] == -1) {
                ms[v] += dfsSm(vn, v);
                maxx[v] = Math.max(maxx[v], ms[vn]);
            }
        }

        return ms[v];
    }
```

```
void smart() {
    maxx = new int[n];
    from = new int[n];
    int list = -1;
    for (int i = 0; i < n; i++)
        if (g[i].length == 1) {
            list = i;
            break;
        }

    startSmartDfs(list);
    for (int i = 0; i < n; i++) {
        int ans = Math.max(maxx[i], n - ms[i]) + 1;
        out.print(ans + " ");
    }
}

void eztests() {
    for (int i = 1; i <= n; i++) {
        int ans = Math.max(i, (n - i + 1));
        out.print(ans + " ");
    }
}

void solve() throws IOException {
    n = readInt();
    if (n == 1) {
        out.println(1);
        return;
    }
    boolean posl = true;
    int[] x = new int[n];
    int[] y = new int[n];
    int[] count = new int[n];
    g = new int[n][];
    ms = new int[n];
    for (int i = 0; i < n - 1; i++) {
        x[i] = readInt() - 1;
        y[i] = readInt() - 1;
        if (x[i] + 1 != y[i])
            posl = false;
        count[x[i]]++;
        count[y[i]]++;
    }
    for (int i = 0; i < n; i++)
        g[i] = new int[count[i]];
    for (int i = 0; i < n - 1; i++) {
        g[x[i]][--count[x[i]]] = y[i];
        g[y[i]][--count[y[i]]] = x[i];
    }

    smart();
    /*
    if (posl)
        eztests();
    else
        stupid();
    */
}

BufferedReader fs = new BufferedReader(new InputStreamReader(System.in));
StringTokenizer tok = new StringTokenizer(" ");
PrintWriter out = new PrintWriter(System.out);

String readLine() throws IOException {
    return fs.readLine();
}

String delim = " ";

String readString() throws IOException {
    while(!tok.hasMoreTokens()) {
        tok = new StringTokenizer(readLine(), delim);
    }
```

```
        return tok.nextToken();
    }

    int readInt() throws IOException {
        return Integer.parseInt(readString());
    }

    public static void main(String[] args) throws IOException {
        new Main().run();
    }

    void run() throws IOException {
        solve();
        out.close();
    }
}
```

Task D (60)

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.StringTokenizer;

public class pass {

    int t, n, p;
    String alf = "";
    int mod = 29;
    void solve() throws IOException {
        for (char i = 'a'; i <= 'z'; i++)
            alf += i;

        String com = readString();
        t = readInt();
        n = readInt();
        p = readInt();
        if (com.equals("split")) {
            while (t --> 0)
                split();
        } else {
            while (t --> 0)
                merge();
        }
    }

    void split() throws IOException {
        if (n == 3) {
            split1();
        } else
            split2();
    }

    void merge() throws IOException {
        if (n == 3) {
            merge1();
        } else
            merge2();
    }

    void merge1() throws IOException {
        String a = readString();
        String b = readString();
        char[] ans = new char[9];
        add1(ans, a);
        add1(ans, b);
        out.println(new String(ans));
    }

    void add1(char[] ans, String aa) {
        char[] a = aa.toCharArray();
        if (a[0] == 'a')
            for (int i = 1; i < 7; i++)
                ans[i - 1] = a[i];
        else if (a[0] == 'b') {
            for (int i = 1; i < 4; i++)
                ans[i - 1] = a[i];
            for (int i = 4; i < 7; i++)
                ans[i + 2] = a[i];
        } else {
            for (int i = 1; i < 7; i++) {
                ans[i + 2] = a[i];
            }
        }
    }

    void split1() throws IOException {
```

```
    String pass = readString();
    String
            a = "a" + pass.substring(0, 6),
            b = "b" + pass.substring(0, 3) + pass.substring(6, 9),
            c = "c" + pass.substring(3, 9);
    out.println(a + " " + b + " " + c);
}


String[] k2 = {
        "000000***",
        "000000***",
        "000***000",
        "***000000",
        "***000000"
};

void merge2() throws IOException {
    int count = (n + 1) / 2;
    char[][] inp = new char[count][];
    for (int i = 0; i < count; i++)
        inp[i] = readString().toCharArray();
    char[] ans = new char[9];
    for (char[] c : inp)
        add2(ans, c);
    out.println(new String(ans));
}

void add2(char[] ans, char[] a) {
    int ai = 1;
    int kk = a[0] - 'a';
    for (int i = 0; i < 9; i++) {
        if (ind(kk, i)) {
            ans[i] = a[ai++];
        }
    }
}

boolean ind(int x, int y) {
    return k2[x].charAt(y) == '0';
}

void split2() throws IOException {
    String pass = readString();
    for (int i = 0; i < n; i++)
        out.print(gen2(i, pass) + " ");
    out.println();
}

String gen2(int kk, String pass) {
    String ans = (char)('a' + kk) + "";
    if (kk == 0 || kk == 1) {
        ans += pass.substring(0, 6);
    } else if (kk == 2) {
        ans += pass.substring(0, 3) + pass.substring(6, 9);
    } else {
        ans += pass.substring(3, 9);
    }
    return ans;
}

BufferedReader fs = new BufferedReader(new InputStreamReader(System.in));
StringTokenizer tok = new StringTokenizer(" ");
PrintWriter out = new PrintWriter(System.out);

String readLine() throws IOException {
    return fs.readLine();
}

String delim = " ";

String readString() throws IOException {
    while(!tok.hasMoreTokens()) {
        tok = new StringTokenizer(readLine(), delim);
```

```
        }
        return tok.nextToken();
    }

    int readInt() throws IOException {
        return Integer.parseInt(readString());
    }

    public static void main(String[] args) throws IOException {
        new pass().run();
    }

    void run() throws IOException {
        solve();
        out.close();
    }
}
```

Task E (12)

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.StringTokenizer;

import static java.lang.Math.*;

public class I_Love_Geoma {

    class Point {
        double x, y;

        Point(double xx, double yy) {
            x = xx;
            y = yy;
        }
    }

    Point readPoint() throws IOException {
        return new Point(readLong(), readLong());
    }

    class kxb {
        double k, b;

        kxb(double kk, double bb) {
            k = kk;
            b = bb;
        }
        kxb(Point a, Point c) {
            k = (c.y - a.y) / (c.x - a.x);
            b = a.y - a.x * k;
        }
    }

    Point A, B;
    kxb lu;
    int n;
    Point[] p;

    void solve() throws IOException {
        n = readInt();
        p = new Point[n];
        for (int i = 0; i < n; i++)
            p[i] = readPoint();

        A = readPoint();
        B = readPoint();
        lu = new kxb(A, B);
        if (n == 2)
            solve1();
        else
            zagl();
    }

    void zagl() {
        if (n == 3)
            out.println("2");
    }

    void solve1() {
        Point f = p[0];
        Point s = p[1];
        if (f.x > s.x) {
            f = p[1];
            s = p[0];
        }
        Point zen = new Point((f.x + s.x) / 2, (s.y + f.y) / 2);
```

```
        double k = (s.x - f.x) / (s.y - f.y);
        double b = zen.y - zen.x * k;
        if (A.y - B.y == 0) {
            if (f.x - s.x == 0) {
                if (zen.y == A.y)
                    out.println(-1);
                else if (zen.y < A.y) {
                    out.println(p[0].y > p[1].y ? 1 : 2);
                } else
                    out.println(p[0].y < p[1].y ? 1 : 2);
            } else {
                if (p[0].x < p[1].x) {
                    out.println(A.x > B.x ? 1 : 2);
                } else
                    out.println(A.x < B.x ? 1 : 2);
            }
        }
    }

    BufferedReader fs = new BufferedReader(new InputStreamReader(System.in));
    StringTokenizer tok = new StringTokenizer("␣");
    PrintWriter out = new PrintWriter(System.out);

    String readLine() throws IOException {
        return fs.readLine();
    }

    String delim = "␣";

    String readString() throws IOException {
        while(!tok.hasMoreTokens()) {
            tok = new StringTokenizer(readLine(), delim);
        }
        return tok.nextToken();
    }

    int readInt() throws IOException {
        return Integer.parseInt(readString());
    }

    long readLong() throws IOException {
        return Long.parseLong(readString());
    }

    public static void main(String[] args) throws IOException {
        new I_Love_Geoma().run();
    }

    void run() throws IOException {
        solve();
        out.close();
    }
}
```

Task F (7)

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.Arrays;
import java.util.StringTokenizer;

import static java.lang.Math.*;

public class Main {

    int n;
    long k;
    long[] r, b;

    void bitkek() {
        long mx = Long.MIN_VALUE;
        int maxBits = 1 << (n + 1);
        for (int bit = 0; bit <= maxBits; bit++) {
            long now = 0;
            long[] pon = {0, 0};
            int ed = 1;
            for (int i = 0; i < n; i++) {
                pon[(ed & bit) == 0 ? 0 : 1] += k;
                now += min(r[i], pon[0]);
                now += min(b[i], pon[1]);
                pon[0] -= min(r[i], pon[0]);
                pon[1] -= min(b[i], pon[1]);
                ed = ed << 1;
            }
            mx = max(mx, now);
        }
        out.println(mx);
    }

    void solve() throws IOException {
        n = readInt();
        k = readLong();
        r = new long[n];
        b = new long[n];
        for (int i = 0; i < n; i++) {
            r[i] = readLong();
            b[i] = readLong();
        }
        bitkek();
    }

    BufferedReader fs = new BufferedReader(new InputStreamReader(System.in));
    StringTokenizer tok = new StringTokenizer("␣");
    PrintWriter out = new PrintWriter(System.out);

    String readLine() throws IOException {
        return fs.readLine();
    }

    String delim = "␣";

    String readString() throws IOException {
        while(!tok.hasMoreTokens()) {
            tok = new StringTokenizer(readLine(), delim);
        }
        return tok.nextToken();
    }

    int readInt() throws IOException {
        return Integer.parseInt(readString());
    }

    long readLong() throws IOException {
```

```
        return Long.parseLong(readString());
    }

    public static void main(String[] args) throws IOException {
        new Main().run();
    }

    void run() throws IOException {
        solve();
        out.close();
    }
}
```