

# Олимпиада СПбГУ по информатике 2018/19 учебного года

Ползик Даниэль Денисович

A	B	C	D	E	F	Sum
100	100	100	60	69	0	429

## Task A (100)

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

public class Main {

    public void run() throws IOException {
        br = new BufferedReader(new InputStreamReader(System.in));
        pw = new PrintWriter(System.out);
        int n = nextInt();
        int m = nextInt();
        if (n > m) {
            pw.print("No");
        } else if (n == m) {
            pw.print("Yes");
        } else {
            int k = m / n;
            if (n * k != m) {
                pw.print("No");
            } else {
                while (k > 1) {
                    if (k % 2 != 0) {
                        pw.print("No");
                        pw.close();
                        return;
                    }
                    k = k / 2;
                }
                pw.print("Yes");
            }
        }
        pw.close();
    }

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

    BufferedReader br;
    PrintWriter pw;
    StringTokenizer st;

    public String nextToken() throws IOException {
        while (st == null || !st.hasMoreTokens()) {
            st = new StringTokenizer(br.readLine());
        }
        return st.nextToken();
    }

    public int nextInt() throws IOException {
```

```
        }  
        return Integer.parseInt(nextToken());  
    }  
}
```

## Task B (100)

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

public class nB {

    public void run() throws IOException {
        br = new BufferedReader(new InputStreamReader(System.in));
        pw = new PrintWriter(System.out);
        String s = br.readLine();
        char[] c = br.readLine().toCharArray();
        for (int i = 0; i < c.length; i++) {
            if (c[i] == 'o') {
                if (i >= 1 && c[i - 1] == 'r') {
                    pw.print("Yes");
                    pw.close();
                    return;
                }
                if (i + 1 < c.length && c[i + 1] == 'r') {
                    pw.print("Yes");
                    pw.close();
                    return;
                }
                if (i + 2 < c.length && c[i + 2] == 'r') {
                    pw.print("Yes");
                    pw.close();
                    return;
                }
            }
        }
        pw.print("No");
        pw.close();
    }

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

    BufferedReader br;
    PrintWriter pw;
    StringTokenizer st;

    public String nextToken() throws IOException {
        while (st == null || !st.hasMoreTokens()) {
            st = new StringTokenizer(br.readLine());
        }
        return st.nextToken();
    }

    public int nextInt() throws IOException {
        return Integer.parseInt(nextToken());
    }
}
```

## Task C (100)

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

public class nC {

    public int[] col;
    public int[] par;
    public ArrayList<ArrayList<Integer>> arr;

    public int bfs(int i, int p) {
        par[i] = p;
        for (int x : arr.get(i)) {
            if (x != p) {
                col[i] = col[i] + bfs(x, i);
            }
        }
        col[i]++;
        return col[i];
    }

    public void run() throws IOException {
        br = new BufferedReader(new InputStreamReader(System.in));
        pw = new PrintWriter(System.out);
        int n = nextInt();
        if (n == 1) {
            pw.print(1);
            pw.close();
            return;
        }
        arr = new ArrayList<ArrayList<Integer>>();
        for (int i = 0; i < n; i++) {
            arr.add(new ArrayList<Integer>());
        }
        for (int i = 0; i < n - 1; i++) {
            int s = nextInt() - 1;
            int f = nextInt() - 1;
            arr.get(s).add(f);
            arr.get(f).add(s);
        }
        col = new int[n];
        par = new int[n];
        bfs(0, 0);
        for (int i = 0; i < n; i++) {
            if (arr.get(i).size() == 1) {
                pw.print(n + "\n");
            } else {
                int[] res = new int[arr.get(i).size()];
                int ind = 0;
                for (int x : arr.get(i)) {
                    if (x == par[i]) {
                        res[ind] = n - col[i];
                    } else {
                        res[ind] = col[x];
                    }
                    ind++;
                }
                int max = res[0];
                for (int j = 1; j < res.length; j++) {
                    if (res[j] > max) {
                        max = res[j];
                    }
                }
                pw.print((max + 1) + "\n");
            }
        }
    }
}
```

```

        }
        pw.close();
    }

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

    BufferedReader br;
    PrintWriter pw;
    StringTokenizer st;

    public String nextToken() throws IOException {
        while (st == null || !st.hasMoreTokens()) {
            st = new StringTokenizer(br.readLine());
        }
        return st.nextToken();
    }

    public int nextInt() throws IOException {
        return Integer.parseInt(nextToken());
    }
}

```

## Task D (60)

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

public class nD {

    public void run() throws IOException {
        br = new BufferedReader(new InputStreamReader(System.in));
        pw = new PrintWriter(System.out);

        String s = br.readLine();
        int t = nextInt();
        int n = nextInt();
        int p = nextInt();
        if (s.equals("split")) {
            if (n == 3) {
                for (int i = 0; i < t; i++) {
                    String pass = br.readLine();
                    pw.println("a" + pass.substring(3, 9) + "_" + "b" + pass.substring(0, 3) +
                        pass.substring(6, 9) + "_" + "c" + pass.substring(0, 6) + "_");
                }
            } else if (n == 5) {
                for (int i = 0; i < t; i++) {
                    String pass = br.readLine();
                    pw.println("a" + pass.substring(3, 9) + "_" + "a" + pass.substring(3, 9) + "_"
                        + "b" + pass.substring(0, 3) + pass.substring(6, 9) + "_" + "c" + pass.
                        substring(0, 6) + "_" + "c" + pass.substring(0, 6));
                }
            } else {
                for (int i = 0; i < t; i++) {
                    String[] pass = br.readLine().split("_");
                    char[] res = new char[9];
                    for (int k = 0; k < pass.length; k++) {
                        if (pass[k].charAt(0) == 'a') {
                            for (int j = 1; j < 7; j++) {
                                res[j + 2] = pass[k].charAt(j);
                            }
                        }
                        if (pass[k].charAt(0) == 'b') {
                            for (int j = 1; j < 4; j++) {
                                res[j - 1] = pass[k].charAt(j);
                            }
                        }
                        if (pass[k].charAt(0) == 'c') {
                            for (int j = 1; j < 7; j++) {
                                res[j - 1] = pass[k].charAt(j);
                            }
                        }
                    }
                    pw.println(res);
                }
            }
        }
        pw.close();
    }

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

    BufferedReader br;
    PrintWriter pw;
    StringTokenizer st;
```

```

public String nextToken() throws IOException {
    while (st == null || !st.hasMoreTokens()) {
        st = new StringTokenizer(br.readLine());
    }
    return st.nextToken();
}

public int nextInt() throws IOException {
    return Integer.parseInt(nextToken());
}

public long nextLong() throws IOException {
    return Long.parseLong(nextToken());
}
}

//      aswords bpassrds cpasswo
//      aaaaaaa buhaaaa cuhaaaa
//      aaaaaaa baaaaaa caaaaaa
//      ainword bplaord cplainw

```

## Task E (69)

```
n = int(input())
x = [0] * n
y = [0] * n
i = 0
while i < n:
    s = raw_input().split("_")
    x[i] = int(s[0])
    y[i] = int(s[1])
    i += 1
s = raw_input().split("_")
xP = int(s[0])
yP = int(s[1])
s = raw_input().split("_")
xQ = int(s[0])
yQ = int(s[1])
a = yQ - yP
b = xP - xQ
c = xQ * yP - xP * yQ

ind = -1
maxL = 0.0
maxH = 0.0
f = True
summ = 0
i = 0
while i < n:
    h = abs((a * x[i] + b * y[i] + c)) / pow(a * a + b * b, 0.5)
    l = pow((xP - x[i]) * (xP - x[i]) + (yP - y[i]) * (yP - y[i]) - h * h, 0.5)
    if ((xP - xQ) * (xP - xQ) + (yP - yQ) * (yP - yQ)) + ((xP - x[i]) * (xP - x[i]) + (yP - y[i]) * (yP - y[i])) < ((x[i] - xQ) * (x[i] - xQ) + (y[i] - yQ) * (y[i] - yQ))):
        l = -1
    if f:
        maxL = l
        maxH = h
        summ = 1
        ind = i
        f = False
    else:
        if abs(l - maxL) < 0.00005 and abs(h - maxH) < 0.00005:
            summ = 2
        else:
            if abs(l - maxL) < 0.00005 and -h + maxH >= 0.00005:
                maxL = l
                maxH = h
                summ = 1
                ind = i
            else:
                if l - maxL >= 0.000001:
                    maxL = l
                    maxH = h
                    summ = 1
                    ind = i
    i += 1
if summ == 2:
    print(-1)
else:
    print((ind + 1))
```



## Task F (0)

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

public class n {

    public void run() throws IOException {
        br = new BufferedReader(new InputStreamReader(System.in));
        pw = new PrintWriter(System.out);
        int d = nextInt();
        int k = nextInt();
        long[] r = new long[d];
        long[] b = new long[d];
        long rS = 0;
        long bS = 0;
        for(int i = 0; i < d; i++){
            r[i] = nextLong();
            rS += r[i];
            b[i] = nextLong();
            bS += b[i];
        }
        long R = 0;
        long B = 0;
        long sum = 0;
        for(int i = 0; i < d; i++){
            if(rS - R > bS - B){
                R += k;
            } else {
                B += k;
            }
            if(R > r[i]){
                R = R - r[i];
                sum += r[i];
            } else {
                sum += R;
                R = 0;
            }
            if(B > b[i]){
                B = B - b[i];
                sum += b[i];
            } else {
                sum += B;
                B = 0;
            }
        }
        pw.print(sum);
        pw.close();
    }

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

    BufferedReader br;
    PrintWriter pw;
    StringTokenizer st;

    public String nextToken() throws IOException {
        while (st == null || !st.hasMoreTokens()) {
            st = new StringTokenizer(br.readLine());
        }
        return st.nextToken();
    }

    public int nextInt() throws IOException {
        return Integer.parseInt(nextToken());
    }
}
```

```
    public long nextLong() throws IOException {  
        return Long.parseLong(nextToken());  
    }  
}
```