

Angewandte Mathematik

1. Protokoll: 22.03.2011

10424 Love Calculator

Java-Code:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws IOException {
        try {
            BufferedReader reader = new BufferedReader(new InputStreamReader(
                System.in));

            char[] name1;
            char[] name2;
            String temp1;
            String temp2;
            temp1 = reader.readLine().toUpperCase();
            temp2 = reader.readLine().toUpperCase();
            while (!temp1.isEmpty()) {

                name1 = temp1.toCharArray();
                name2 = temp2.toCharArray();

                int value1 = 0;
                int value2 = 0;
                int temp = 0;

                for (int i = 0; i < name1.length; i++) {
                    temp = name1[i] - 64;
                    if (temp >= 1 && temp < 27)
                        value1 += temp;
                }
                for (int i = 0; i < name2.length; i++) {
                    temp = name2[i] - 64;
                    if (temp >= 1 && temp < 27)
                        value2 += temp;
                }

                int t1 = quersumme(value1);
                int t2 = quersumme(value2);

                if (t1 < t2)
                {
                    System.out.format("%#.2f", (double) t1 / t2 * 100);
                    System.out.println(" %");
                }
                else
                {
                    System.out.format("%#.2f", (double) t2 / t1 * 100);
                    System.out.println(" %");
                }

                temp1 = reader.readLine().toUpperCase();
                temp2 = reader.readLine().toUpperCase();
            }
            reader.close();
        } catch (Exception e) {
        }
    }
}
```

```
private static int quersumme(int value) {
    int g = 0;
    while (value != 0) {
        g += value % 10;
        value /= 10;
    }
    if (g > 9)
        g = quersumme(g);
    return g;
}
```

Ergebnis:

8685648	10424 Love Calculator	Accepted	JAVA	0.120	2011-03-27 19:41:42
---------	-----------------------	----------	------	-------	---------------------

10370 Above Average

Java-Code:

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

public class Main {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int number_of_datasets = in.nextInt();
        for(int i=0;i<number_of_datasets;i++){
            ArrayList<Double> numbers = new ArrayList<Double>();
            int number_of_students = in.nextInt();
            double schnitt=0;
            int count=0;
            for (int k = 0; k < number_of_students; k++) {
                double temp = in.nextDouble();
                numbers.add(temp);
                schnitt += temp;
            }
            schnitt/= number_of_students;
            for (int m = 0; m < number_of_students; m++) {
                if (numbers.get(m) > schnitt) count++;
            }
            double output = (double)count/number_of_students*100;

            System.out.format("%.3f", output);
            System.out.println("%");
        }
    }
}
```

Ergebnis:

8685643	10370 Above Average	Accepted	JAVA	0.932	2011-03-27 19:39:05
---------	---------------------	----------	------	-------	---------------------

11172 Relational Operator

Java-Code:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Apl {
    public static void main(String[] args) {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        try {
            String inputLine = reader.readLine();
            int c = Integer.parseInt(inputLine);
        }
    }
}
```

```
        for(int i=0; i<c; i++){
            inputLine = reader.readLine();
            String[] x = inputLine.split(" ");
            int a = Integer.parseInt(x[0]);
            int b = Integer.parseInt(x[1]);
            if (a==b) System.out.println("=");
            else {
                if(a<b) System.out.println("<");
                else System.out.println(">");
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
```

Ergebnis:

8683966	11172 Relational Operator	Accepted	JAVA	0.108	2011-03-27 10:24:06
---------	---------------------------	----------	------	-------	---------------------

10071 Back to High School Physics

Java-Code:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws IOException
    {
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        String input;
        String[] temp;
        int v;
        int t;
        while ((input=in.readLine())!=null)
        {
            temp = input.split(" ");
            v = Integer.parseInt(temp[0]);
            t = Integer.parseInt(temp[1]);
            System.out.println(v*t*2);
        }
    }
}
```

Ergebnis:

8683938	10071 Back to High School Physics	Accepted	JAVA	1.520	2011-03-27 10:11:34
---------	-----------------------------------	----------	------	-------	---------------------

10038 Jolly Jumpers

Java-Code:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Collections;

public class Main {

    public static void main(String[] args) throws IOException {
        try {
            BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
            String input;
            String[] temp;
```

```
boolean jolly;
ArrayList<Integer> numbers = new ArrayList<Integer>();
while (!(input = in.readLine()).isEmpty()) {
    jolly = true;
    temp = input.split(" ");
    int count = temp.length - 1;
    if (Integer.valueOf(temp[0]) == 0) jolly = false;

    for (int i = 1; i < count; i++)
        numbers.add(Math.abs(Integer.valueOf(temp[i]) -
            Integer.valueOf(temp[i + 1])));

    Collections.sort(numbers);

    for (int i = 0; i < count - 1; i++)
        if (numbers.get(i) != i + 1) jolly = false;

    if (jolly) System.out.println("Jolly");
    else System.out.println("Not jolly");
    numbers.clear();
} catch (Exception e) {
}
}
```

Ergebnis:

8684616	10038 Jolly Jumpers	Accepted	JAVA	0.536	2011-03-27 14:06:59
---------	---------------------	----------	------	-------	---------------------