Angewandte Mathematik

2. Protokoll: 29.03.2011

11879 Multiple of 17

Java-Code:

```
/**
 * Angewandte Mathematik, SS11
 * Problem: 11879 Muliple of 17
 * Link:
http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=226&page=show_problem&problem=3001
 *
 * @author Brielbeck, Daniel
 * @author Weber, Georg
 * @version 1.0, 03/29/2011
 *
 * Method : Ad-Hoc
 * Status : Accepted
 * Runtime: 0.100
 */
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws Exception {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        String input = reader.readLine();
        char[] number = input.toCharArray();
        int l = number.length;
        while (!input.equals("0")) {
            int n = 0;
            for (int i = 0; i < l; i++) {
                n = n * 10 + number[i] - 48;
                n = n % 17;
            }
            if (n != 0)
                System.out.println("0");
            else
                System.out.println("1");
            input = reader.readLine();
            number = input.toCharArray();
            l = number.length;
        }
    }
}
```

Ergebnis:

8690246	11879 Multiple of 17	Accepted	JAVA	0.100	2011-03-29 12:51:00

623 500!

Java-Code:

```
/**
 * Angewandte Mathematik, SS11
 * Problem: 623 - 500!
 * Link:
http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=8&page=show_problem&problem=564
 *
 * @author Brielbeck, Daniel
 * @author Weber, Georg
 * @version 1.0, 03/29/2011
 *
 * Method : Ad-Hoc
 * Status : Accepted
 * Runtime: 1.960
 */

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.math.BigInteger;

public class Main {

    public static void main(String[] args) throws Exception {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        String input;
        int n;
        BigInteger zahl;
        while ((input = reader.readLine()) !=null) {
            input=input.split("!")[0];
            n = Integer.valueOf(input);
            zahl = new BigInteger("1");
            for (int i = n; i > 1; i--) {
                zahl = zahl.multiply(new BigInteger(String.valueOf(i)));
            }
            System.out.println(n+"!");
            System.out.println(zahl);
        }
    }
}
```

Ergebnis:

8690448	623 500!	Accepted	JAVA	1.960	2011-03-29 13:49:58

10929 You can say 11

Java-Code:

```
/**
 * Angewandte Mathematik, SS11
 * Problem: 10929 - You can say 11
 * Link:
http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=21&page=show_problem&problem=1870
 *
 * @author Brielbeck, Daniel
 * @author Weber, Georg
 * @version 1.0, 03/29/2011
 *
 * Method : Ad-Hoc
 * Status : Accepted
 * Runtime: 0.392
 */

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
  public static void main(String[] args) throws Exception{
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    String input;
    while (!(input = in.readLine()).equals("0")) {
      char[] zahlen = input.trim().toCharArray();
      if(zahlen.length>0 && zahlen.length<=1000){
        int erg=Integer.parseInt(String.valueOf(zahlen[0]));
        boolean plus = false;
        for(int i=1;i<zahlen.length;i++){
          if(plus){
            erg+=Integer.parseInt(String.valueOf(zahlen[i]));
            plus = false;
          } else {
            erg-=Integer.parseInt(String.valueOf(zahlen[i]));
            plus = true;
          }
        }
        if(erg%11==0 || erg==0) System.out.println(
              input + " is a multiple of 11.");
        else System.out.println(input + " is not a multiple of 11.");
      }
    }
  }
}
```

Ergebnis:

8690356	10929 You can say 11	Accepted	JAVA	0.392	2011-03-29 13:24:50

725 Division

Java-Code:

```
/**
 * Angewandte Mathematik, SS11
 * Problem: 725 Division
 * Link:
http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=9&page=show_problem&problem=6
66
 *
 * @author Brielbeck, Daniel
 * @author Weber, Georg
 * @version 1.0, 03/29/2011
 *
 * Method : Ad-Hoc
 * Status :
 * Runtime:
 */

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

public class Main {
  public static void main(String[] args) throws IOException{
    boolean[] check = new boolean[10];
    String[] zahlen = new String[10];
    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
    String in = reader.readLine();
    double input = Double.parseDouble(in);
    for(int i=0;i<check.length;i++) check[i]=true;
    for(int i1=0;i1<zahlen.length;i1++){
      if(check[i1]){
        zahlen[0]=String.valueOf(i1);
        check[i1]=false;
        for(int i2=0;i2<zahlen.length;i2++){
          if(check[i2]){
            zahlen[1]=String.valueOf(i2);
            check[i2]=false;
            for(int i3=0;i3<zahlen.length;i3++){
              if(check[i3]){
                zahlen[2]=String.valueOf(i3);
                check[i3]=false;
                for(int i4=0;i4<zahlen.length;i4++){
                  if(check[i4]){
                    zahlen[3]=String.valueOf(i4);
                    check[i4]=false;
                    for(int i5=0;i5<zahlen.length;i5++){
                      if(check[i5]){
                        zahlen[4]=String.valueOf(i5);
                        check[i5]=false;
                        for(int i6=0;i6<zahlen.length;i6++){
                          if(check[i6]){
                            zahlen[5]=String.valueOf(i6);
                            check[i6]=false;
                            for(int i7=0;i7<zahlen.length;i7++){
                              if(check[i7]){
                                zahlen[6]=String.valueOf(i7);
                                check[i7]=false;
                                for(int i8=0;i8<zahlen.length;i8++){
                                  if(check[i8]){
                                    zahlen[7]=String.valueOf(i8);
                                    check[i8]=false;
                                    for(int i9=0;i9<zahlen.length;i9++){
                                      if(check[i9]){
                                        zahlen[8]=String.valueOf(i9);
                                        check[i9]=false;
                                        for(int i10=0;i10<zahlen.length;i10++){
                                          if(check[i10]){
                                            zahlen[9]=String.valueOf(i10);
                                            check[i10]=false;
                                            String tmp1="";
                                            String tmp2="";
                                            double z1;
                                            double z2;
                                            for(int i0=0;i0<5;i0++) tmp1+=zahlen[i0];
                                            z1=Double.parseDouble(tmp1);
                                            for(int i0=5;i0<10;i0++) tmp2+=zahlen[i0];
                                            z2=Double.parseDouble(tmp2);
                                            if(input == (z1/z2)) System.out.print(tmp1+" / "+tmp2+" = "+in+"\n");
                                            check[i10]=true;
                                          }
                                        }
```

```
                                    check[i9]=true;
                                  }
                                }
                                check[i8]=true;
                              }
                            }
                            check[i7]=true;
                          }
                        }p
                        check[i6]=true;
                      }
                    }
                    check[i5]=true;
                  }
                }
                check[i4]=true;
              }
            }
            check[i3]=true;
          }
        }
        check[i2]=true;
      }
    }
    check[i1]=true;
  }
 }
}
```

Ergebnis:

Ergebnisse korrekt – Problem: zu lange Laufzeit