

Aufgabe 1

```

def div(dividend, divisor):
{P} ≡ {dividend ≥ 0 ∧ divisor > 0} ≡ { (dividend = 0*divisor+dividend) ∧ dividend ≥ 0 ∧ divisor > 0 }
    result = 0
{R1} ≡ { (dividend = result*divisor+dividend) ∧ divisor > 0 ∧ dividend ≥ 0 }
    rest = dividend
{INV} ≡ { (dividend = result*divisor+rest) ∧ divisor > 0 ∧ rest ≥ 0 }
    while rest >= divisor:
        {INV ∧ B} ≡ { (dividend = result*divisor+rest) ∧ divisor > 0 ∧ rest ≥ 0 ∧ rest ≥ divisor ∧ τ = rest }
        {R2} ≡ { (dividend = result*divisor+divisor+rest-divisor) ∧ divisor > 0 ∧ rest ≥ divisor ∧ τ = rest }
            rest = rest - divisor
        {R3} ≡ { (dividend = (result+1)*divisor+rest) ∧ divisor > 0 ∧ rest ≥ 0 ∧ τ = rest-divisor }
        {R4} ≡ { (dividend = result*divisor+divisor+rest) ∧ divisor > 0 ∧ rest ≥ 0 ∧ τ < rest }
            result = result + 1
        {INV} ≡ { (dividend = result*divisor+rest) ∧ divisor > 0 ∧ rest ≥ 0 ∧ τ < rest }
{INV ∧ ¬B} ≡ { (dividend = result*divisor+rest) ∧ divisor > 0 ∧ rest ≥ 0 ∧ divisor > rest }
{Q} ≡ { (dividend=result*divisor+rest) ∧ rest ≥ 0 ∧ divisor > rest }
    return (result, rest)

```

{P} zu {Q} mit Sequenzregel, aus:

{P} zu {R1} mit Zuweisungsaxiom

{R1} zu {INV} mit Zuweisungsaxiom

{INV} zu {INV ∧ ¬B} mit While-Regel, aus:

mit $\tau(\text{rest}, \text{divisor}, \text{result}) := \text{rest}$, folgt wegen $\text{rest} \geq 0$ auch $\tau \geq 0$

{INV ∧ B} zu {R2} mit Konsequenzregel II

{R2} zu {R3} mit Zuweisungsaxiom

{R3} zu {R4} mit Konsequenzregel II

{R4} zu {INV} mit Zuweisungsaxiom

Mit erweiterter While-Regel folgt totale Korrektheit

{INV ∧ ¬B} zu {Q} mit Konsequenzregel II

durch rekursive Anwendung der Sequenzregel folgt totale Korrektheit

Aufgabe 2

```
public class IEEE_Test
{
    public static void main(String[] args)
    {
        // Rechnen mit Floating Points
        double x = 1.0;
        double y = 0.0;
        double z = Double.POSITIVE_INFINITY;
        System.out.println(y/x);
        // => 0.0
        // 0/1=0, der Typ bleibt Double
        System.out.println((x/y+x/y));
        // => Infinity
        // Unendlich kann addiert werden und bleibt unendlich
        System.out.println((x/y-x/y));
        // => NaN (= not a number)
        // Auf Unendlich ist keine Differenz definiert.
        System.out.println(y/y);
        // => NaN
        // Division ist nicht auf Unendlichkeiten definiert.
        System.out.println(z/z);
        // => NaN
        // Division ist nicht auf Unendlichkeiten definiert.

        // Rechnen mit Integers
        int m = 3;
        int n = 1;
        System.out.println(x/m);
        // => 0.3333333333333333
        // Integer werden bei nicht ganzzahligem Ergebnis einer Division zu Doubles.
    }
}
```

```

System.out.println(n/m);
// => 0
// Integer bleiben bei ganzzahligem Ergebnis einer Division Integer.

// Maxima und Minima von Zahlen
int max = 2147483647;
int min = -2147483648;
System.out.println(max+1);
// => -2147483648
System.out.println(min-1);
// => 2147483647
System.out.println((max+1)==(min));
// => true
// 32 bit Integer reichen von  $-2^{31}$  bis  $2^{31}-1$ . Wenn sie aus dem Wertebereich laufen,
// fangen sie oben/unten wieder an, daher ist  $\text{max}+1 == \text{min}$ .
System.out.println(Double.MAX_VALUE*2);
// => Infinity
// Der Maximalwert von Doubles ist Infinity. 2 mal unendlich bleibt unendlich.

// Mathematische Funktionen
System.out.println(Math.sqrt(-x));
// => NaN
// Java kann nicht in Komplexe Zahlen umrechnen.
System.out.println(Math.log(-x));
// => NaN
// Der Logarithmus von negativen Zahlen ist nicht definiert.

// Bit Operationen
System.out.println(3|4);
// => 7
// | ist ein Bitweise oder.  $3_{10}|4_{10} = 011_2|100_2 = 111_2 = 7_{10}$ .
System.out.println(~5);
// => -6
// ~ ist eine Bitweise negation  $\sim a = -a-1$ .

```

```
System.out.println(2^2^3);  
// => 3  
// ^ ist ein Bitweise exklusives oder. 10^10^11 = 00^11 = 11
```

```
// Bool Operationen
```

```
System.out.println(true || 12%3 == 1);  
// => true  
// true or false = true
```

```
System.out.println(1<2 && 1<0);  
// => false  
// true and false = false
```

```
System.out.println(true? 8 : 7);  
// => 8
```

```
// Der Ternary Operator 'condition? a : b' ist a, wenn condition true ist, sonst ist er b.
```

```
}
```

```
}
```

Aufgabe 3

```
import java.util.Random;

public class SomeCalculations
{
    //Aufgabenteil a)
    public static int querSumme(int n)
    {
        if( n<=9 )
        {
            return n;
        }
        return n%10 + querSumme(n/10);
    }

    public static boolean multipleOf3(int n)
    {
        if( (querSumme(n)%3 == 0) && (n%3 == 0) )
        {
            return true;
        }
        return false;
    }

    //Aufgabenteil b)
    public static int weekday(int year, int month, int day)
    {
        int y0 = year - (14-month)/12;
        int x = y0 + y0/4 - y0/100 + y0/400;
        int m0 = month + 12*((14-month)/12)-2;
        return (day + x + 31*m0/12)%7;
    }
}
```

```
}
```

```
//Aufgabenteil c)
```

```
public static int gluecksspieler(int bargeld)
{
    Random rand = new Random();
    while( bargeld > 0 )
    {
        if( rand.nextInt(2) == 0 )
        {
            bargeld ++;
        }
        else
        {
            bargeld --;
        }
        System.out.println(bargeld);
    }
    System.out.println("alles verloren!");
    return 0;
}
```

```
//Tests
```

```
public static void main(String[] args)
{
    //Aufgabenteil a)
    System.out.println(querSumme(12));
    System.out.println(querSumme(124));
    System.out.println(multipleOf3(12));
    System.out.println(multipleOf3(124));

    //Aufgabenteil b)
    System.out.println("Hib ein Jahr ein:");
    int year = Keyboard.readInt();
}
```

```

// Jahre haben keinen eingeschränkten Wertebereich
System.out.println("Gib einen Monat ein:");
int month = Keyboard.readInt();
while( month > 12 || month < 1 )
{
    System.out.println("Eingabe nicht im Wertebereich zwischen 1 und 12, versuch es nochmal:");
    month = Keyboard.readInt();
}
System.out.println("Gib einen Tag ein:");
int day = Keyboard.readInt();
while( day > 31 || day < 1 )
{
    System.out.println("Eingabe nicht im Wertebereich zwischen 1 und 31, versuch es nochmal:");
    day = Keyboard.readInt();
}
System.out.println("Der Wochentag ist:");
String dayName = "";
switch( weekday(year,month,day) )
{
    case 0: dayName = "So"; break;
    case 1: dayName = "Mo"; break;
    case 2: dayName = "Di"; break;
    case 3: dayName = "Mi"; break;
    case 4: dayName = "Do"; break;
    case 5: dayName = "Fr"; break;
    case 6: dayName = "Sa"; break;
}
System.out.println(dayName);

//Aufgabenteil c)
gluecksspieler(100);

```

```

}

```

```

}

```