

# Informatik 1 für Regenerative Energien

## Probeklausur 2: Lösungen

Jörn Loviscach

Versionsstand: 24. Februar 2011, 00:11



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

1.  $-15$  ist  $11110001$  und  $-1$  ist  $11111111_2$ .

```
11110001
+11111111
11111111
-----
11110000
```

2.  $a|b$  ist  $0xBF$ ,  $a\&b$  ist  $0x2$ ,  $a^b$  ist  $0xBD$ .

3.  $((a + (5 \% 3)) < b) \ || \ ((a \geq 14) \ \&\& \ (b > 7))$
- ```
      2      true      true
    44      true
  false
  true
```

4. `bool isSquare(int n)`

```
{
    for(int i = 0; i <= n; i++) // 0 statt 1
    {
        if(n == i*i) // == statt =
        {
            return true;
        }
    }
    return false; // fehlte
}
```

5. `int countSamples(Sample samples[], int number, double startTime)`

```
{
    int count = 0;
    for(int i = 0; i < number; i++)
    {
        if(samples[i].time >= startTime)
        {
```

```

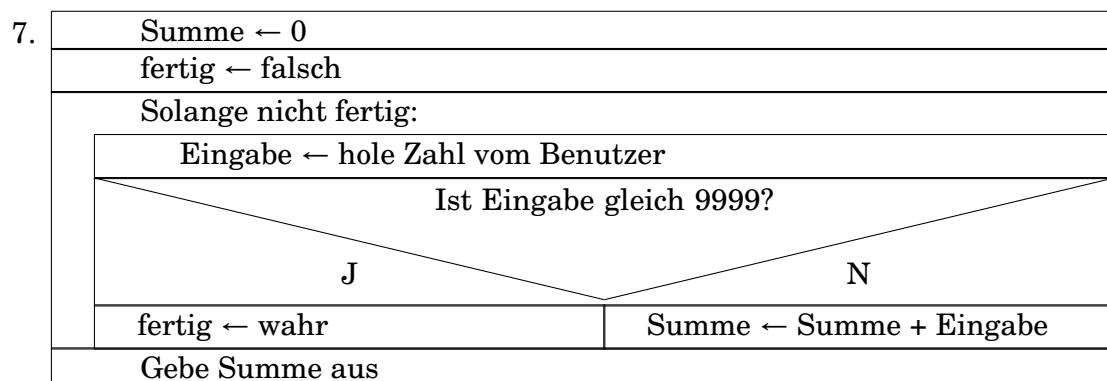
        count++;
    }
}
return count;
}

```

```

6. int a = 42;
   int b = 13;
   //...
   switch(a)
   {
   case 1:
       b = 5;
       break;
   case 2:
   case 3:
       b = 9;
       break;
   default:
       b = 17;
   }

```



```

8. bool endsWithDot(char text[])
{
    int i = 0;
    char lastCharacterSoFar = 0;
    while(text[i] != 0)
    {
        lastCharacterSoFar = text[i];
        i++;
    }
    return lastCharacterSoFar == '.';
}

```

oder

```

bool endsWithDot(char text[])
{

```

```

int length = strlen(text);
if(length != 0)
{
    return text[length-1] == '.'; // -1, weil ab 0
}
else
{
    return false;
}
}

```

und viele weitere Möglichkeiten.

9.  $42.0f + 0.00000001f \rightarrow 42.0f$  (beschränkte Auflösung)  
 $42.0f + 0.00000001f - 42.0f \rightarrow +0.0f$   
 $1.0f / (42.0f + 0.00000001f - 42.0f) \rightarrow +INF$  (Sonderzahl)

```

10. int a = 42; // 42
    int *b = &a; // 42
    b[0] = 5; // 5
    b--; // 5
    b -= 8; // 5
    b[9]++; // 6

```

11. Vier Vergleiche für das erste Element, drei fürs zweite, zwei fürs dritte, eines fürs vierte. Das fünfte Element ist dann schon am Platz. Also insgesamt zehn Vergleiche.

12. Zum Beispiel:

```

bool f(unsigned int x)
{
    return x % 2 == 0; // Ist x gerade?
}

```