

# Informatik 1 für Regenerative Energien

## Probeklausur 1: Lösungen

Jörn Loviscach

Versionsstand: 15. März 2011, 15:06



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. 3 ist 00000011<sub>2</sub> und -5 ist 11111011<sub>2</sub>.

$$\begin{array}{r} 00000011 \\ +11111011 \\ \hline 11111110 \end{array}$$

2. ~a ist 0xED, a&b ist 0x00, a^b ist 0x3F.

3. (( (a + (3 \* 4)) > 13) && (! b)) || ((a < 60) && (a != 30))
- |                 |                |                |                |
|-----------------|----------------|----------------|----------------|
| _____12_____    | false          | _____true_____ | _____true_____ |
| _____54_____    |                |                | _____true_____ |
| _____true_____  |                |                |                |
| _____false_____ |                |                |                |
|                 | _____true_____ |                |                |

4. int factorial(int n)

```
{  
    int result = 1; // 1, nicht 0  
    for(int i = 2; i <= n; i++) // <= n statt < n  
    {  
        result *= i;  
    }  
    return result; // hat gefehlt  
}
```

5. void printAdults(int number, Person persons[])

```
{  
    for(int i = 0; i < number; i++)  
    {  
        if(persons[i].age >= 18)  
        {  
            puts(persons[i].name);  
        }  
    }  
}
```

```

    }
}
```

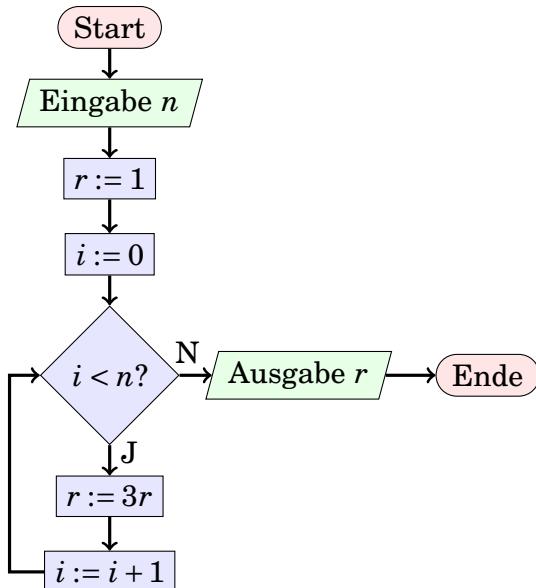
6. Person myFriends[] = // [] fehlte

```

{
    {"Berta", 42}, // Komma statt Semikolon
    {"Karlhei", 43} // nur acht Zeichen inkl. Null am Ende
};
```

7. **c1**

c1jl: J, N hinzugefügt



8. if (myFriends[0].name[4] == 'a')  
{  
 // ...  
}  
if (myFriends[1].name[1] == 'a')  
{  
 // ...  
}  
if (myFriends[0].name[0] - 'B' + 'a' == 'a')  
{  
 // ...
}

und viele weitere Möglichkeiten.

9. Hier läuft int \* int über. Mögliche Abhilfe:

```

long product(int a, int b)
{
    return a * (long)b;
}
```

10. bool hasNext(void)

```

{
    return count > 0;
}

int pop(void)
{
    count--;
    return stack[count];
}

```

11.  $13 + 3n + \frac{1}{n^2+2}$  ist asymptotisch  $O(n)$  und  $O(n^2)$  und  $O(n^3)$ . (Begründung war nicht gefragt: Das ist  $O(n)$ , weil  $13 + 3n + \frac{1}{n^2+2} \leq 4n$  für  $n \geq 14$ . Jede Funktion in  $O(n)$  ist automatisch in  $O(n^2)$ ; jede Funktion in  $O(n^2)$  ist automatisch in  $O(n^3)$ .)

12. Zum Beispiel:

```

int f(int x)
{
    if(x < 0)
    {
        return -x;
    }
    return x;
}

```

Oder (aber eine Lösung reicht):

```

int f(int x)
{
    int result = x;
    if(x < 0)
    {
        result = -result;
    }
    return result;
}

```