

Informatik 1 für Regenerative Energien

Klausur vom 21. September 2011: Lösungen

Jörn Loviscach

Versionsstand: 27. September 2011, 19:53



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. -11 ist 11110101_2 und 14 ist 00001110_2 .

$$\begin{array}{r} 11110101 \\ +00001110 \\ \hline 111111 \\ \hline 00000011 \end{array}$$

2. $\sim a$ ist $0xC5$, $a|b$ ist $0x7E$, a^b ist $0x4C$.

3. $a \ || \ (((b \% 3) + 5.0f) > 1.0f) \ \&\& \ (! a)$
- ```
 0_ true_
 5.0f_
 true_
 true_
 true_
```

4. `bool isPrime(unsigned int x) // Rückgabetyyp bool statt void`
- ```
{
    for(unsigned int i = 2; i < x; i++) // Zuweisung statt Vergleich
    {
        if(x%i == 0) // teilbar <=> Rest 0, nicht Rest 1
        {
            return false;
        }
    }
    return true;
}
```

5. `int addAmounts(Item items[], int number, char initial)`
- ```
{
 int sum = 0;
 for(int i = 0; i < number; i++)
 {
 if(items[i].name[0] == initial)
```

```

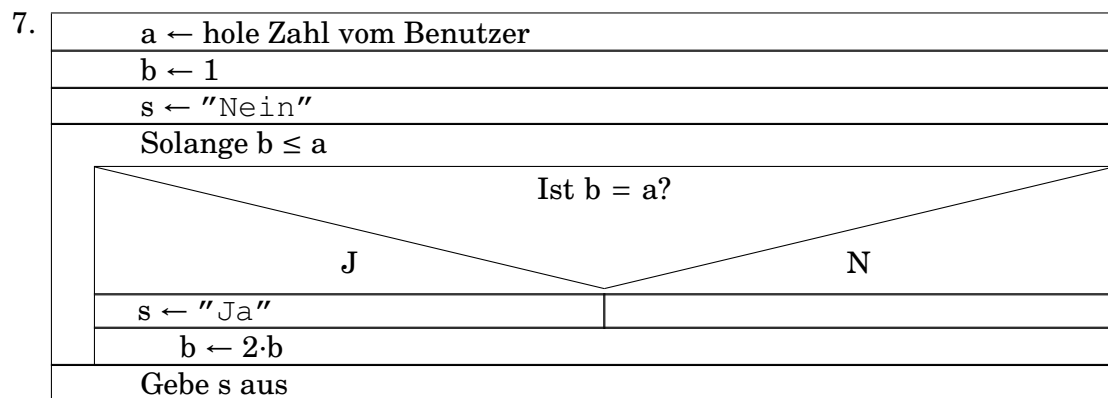
 {
 sum += items[i].amount;
 }
}
return sum;
}

```

```

6. int x = 42;
 int y = 0;
 // ...
 if(x == 1 || x == 2)
 {
 y = x;
 y += 10; // Nach dem case stand kein break!
 }
 else if(x == 3)
 {
 y += 10;
 }
 else
 {
 y = 11;
 }

```



```

8. bool findAB(char a[])
{
 int n = strlen(a);
 for(int i = 0; i < n-1; i++)
 {
 if(a[i] == 'A' && a[i+1] == 'B')
 {
 return true;
 }
 }
 return false;
}

```

9. *a ist 5. a zählt, wie oft die Funktion aufgerufen wird. Der Aufruf  $f(3)$  führt zum Aufruf von  $f(2)$  und  $f(1)$ , der Aufruf von  $f(2)$  wiederum zum Aufruf von  $f(1)$  und  $f(0)$ .*
10. *a ist 42, b ist 100, c ist 41.*
11. *Der Wert 0 (weil das einzige Einser-Bit von 16 durch die Multiplikation nach links herausgeschoben wird).*
12. *Wenn  $x$  kleiner als null ist, wird 1 addiert und später im return wieder abgezogen. Wenn  $x$  größer oder gleich null ist, wird 1 abgezogen und später im return wieder addiert.*

```
int f(int x)
{
 return x;
}
```