

Informatik 1 für Regenerative Energien

Probeklausur 2

Jörn Loviscach

Versionsstand: 8. März 2011, 19:07



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.

Drei Punkte pro Aufgabe. Mindestpunktzahl zum Bestehen: 15 Punkte. Hilfsmittel: maximal vier einseitig oder zwei beidseitig beschriftete DIN-A4-Spickzettel beliebigen Inhalts, möglichst selbst verfasst oder zusammengestellt; kein Skript, keine anderen Texte, kein Taschenrechner, kein Computer, kein Handy und Ähnliches.

Name	Vorname	Matrikelnummer	E-Mail-Adresse, falls nicht in Mailingliste

Fingerübungen

1. Welches Bitmuster (acht Bit mit Zweierkomplement) steht für die Zahl -15 ? Welches für die Zahl -1 ? Wie rechnet man dann $-15 + (-1)$ aus?^{c1} Schreiben Sie das mit Bits hin.
2. Gegeben sind die C-Variablen `unsigned char a = 0xA3;` und `unsigned char b = 0x1E;`. Geben Sie in Hexadezimal an, welche Werte die drei C-Ausdrücke `a|b` und `a&b` und `a^b` bei Rechnung mit acht Bit haben.
3. Geben Sie durch Klammern an, wie die Bedingung im `if` des folgenden C-Fragments ausgewertet wird. Was sind jeweils die Teilergebnisse?

```
int a = 42;
float b = 13.0f;
if(a + 5 % 3 < b || a >= 14 && b > 7)
{
    // ...
}
```

4. Die folgende C-Funktion soll angeben, ob eine eingegebene Zahl $n \geq 0$ eine Quadratzahl ist, also gleich einer der Zahlen $0, 1, 4, 9, 16, 25, \dots$ ist. Es sind aber drei Fehler in der Funktion. Finden und korrigieren Sie diese.

^{c1}ji: -1 statt -4

```

bool isSquare(int n) // fehlerhaft
{
    for(int i = 1; i <= n; i++)
    {
        if(n = i*i)
        {
            return true;
        }
    }
}

```

5. Gegeben sei die Datenstruktur `Sample` für einen Messwert und den Zeitpunkt, zu dem er gemessen wurde:

```

struct Sample
{
    double value; // der gemessene Wert
    double time; // Zeitpunkt in Sekunden
};
typedef struct Sample Sample;

```

Schreiben Sie eine C-Funktion

```

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

```

die als Ergebnis eine ganze Zahl zurückgibt – nämlich wie viele der Messwerte im Array `samples` eine Zeitangabe größer oder gleich der `startTime` haben. In `number` wird der Funktion die Anzahl der Messwerte des Arrays `samples` übergeben.

6. Formen Sie dieses C-Programmfragment mit Hilfe von `switch` um:

```

int a = 42;
int b = 13;
//...
if(a == 1)
{
    b = 5;
}
else if(a == 2 || a == 3)
{
    b = 9;
}
else
{
    b = 17;
}

```

Kreative Anwendung

7. Ein Programm soll so lange Zahlen vom Benutzer abfragen, bis der die Zahl 9999 eingibt. Dann soll das Programm die Summe der eingegebenen Zahlen ausgeben. Lösen Sie das mit einem Struktogramm, das Eingabe, Ausgabe, Zuweisung, Addition und Schleife benutzt. (Ignorieren Sie mögliche Probleme mit Überlauf oder Rundung.)

8. Schreiben Sie eine C-Funktion, die zurückmeldet, ob ein übergebener Text (ein nullterminierter C-String) mit einem Punkt endet. Die Funktion soll diese Deklaration haben:

```
bool endsWithDot(char text[]);
```

Achtung: Die Länge des Texts wird hier nicht gesondert übergeben.

9. Die folgende Rechnung wird im Rechner mit den üblichen `float`-Zahlen ausgeführt. Notieren Sie die Ergebnisse für jede einzelne Zwischenoperation.

```
1.0f / (42.0f + 0.00000001f - 42.0f)
```

10. Schreiben Sie neben jede Zeile, welcher Wert nach Ausführung der Zeile in der Variablen `a` steht:

```
int a = 42;
int *b = &a;
b[0] = 5;
b--;
b -= 8;
b[9]++;
```

11. Das Sortierverfahren „Selection Sort“ arbeitet so: Gegeben ist eine Liste aus n zu sortierenden Elementen. Es wird zunächst von vorne bis hinten durch Vergleiche das kleine Element gesucht. Dies tauscht seinen Platz mit dem Element am Anfang der Liste. Dann wird die Teil-Liste betrachtet, die ab dem zweiten Element anfängt. Davon wird von vorne bis hinten das kleinste Element gesucht. Das tauscht seinen Platz mit dem zweiten Element der Liste usw. Wie viele Vergleiche benötigt dieser Algorithmus, um eine Liste aus $n = 5$ Einträgen zu sortieren?

12. Schreiben Sie diese C-Funktion so um, dass man sofort sehen kann, was sie tut.

```
bool f(unsigned int x)
{
    bool a = false;
    for(unsigned int i = 0; i <= x; i += 2)
    {
        if(i == x)
        {
            a = true;
        }
    }
}
```

```
    }  
  }  
  return a;  
}
```