

Informatik 1 für Regenerative Energien

Klausur vom 22. September 2014

Jörn Loviscach

Versionsstand: 22. September 2014, 15:57



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

Fingerübungen

1. Welches Bitmuster (acht Bit mit Zweierkomplement) steht für die Zahl -11 ? Welches für die Zahl 17 ? Wie rechnet man dann $-11 + 17$ aus? Schreiben Sie das mit Bits hin.
2. Gegeben sind die C-Variablen `unsigned char a = 0x9C;` und `unsigned char b = 0xAF;`. 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 y = 7;
bool x = true;
int z = 2;
if(z * 3 / y == 0 && ! x || y % 3 == 1)
{
    // ...
}
```

4. Die folgende C-Funktion soll feststellen, ob das übergebene Datum erlaubt ist. Beispiel: `f(21,9,2014)` soll `true` ergeben, aber `f(32,12,2014)` soll `false` ergeben. Jahreszahlen vor 1900 sollen auch `false` ergeben. Vereinfachung: Jedes vierte Jahr soll ein Schaltjahr sein. Es sind aber drei Fehler in der Funktion. Finden und korrigieren Sie diese.

```
bool pruefeDatum(int tag, int monat, int jahr)
{
    if(tag < 1 && monat < 1)
    {
        return false;
    }
    if(jahr < 1900)
    {
        return false;
    }
    // Die Zahlen 28, 30, 31 sind ohne Fehler!
    int monatslaenge[] =
        {-1, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if(jahr % 4 > 0)
    {
        monatslaenge[2] = 29;
    }
    if(monat > 12 || tag > monatslaenge)
    {
        return false;
    }
    return true;
}
```

5. Gegeben ist diese Datenstruktur:

```
struct Benutzer
{
    char benutzername[16];
    char eMailAdresse[16]; //z.B. a.mueller@abc.de
};
typedef struct Benutzer Benutzer;
```

Schreiben Sie eine C-Funktion:

```
char* findeDoppelteEMailAdresse(Benutzer b[], int anzahl)
```

Diese soll die übergebenen Benutzerdaten danach durchsuchen, ob eine E-Mail-Adresse doppelt vorkommt und dann einen der davon betroffenen Benutzernamen zurückgeben – oder `NULL`, falls keine E-Mail-Adresse doppelt vorkommt. In `anzahl` wird der Funktion die Anzahl der Einträge des Arrays `b` übergeben. Sie können `strcmp` aus `string.h` verwenden.

6. Übersetzen Sie dieses C-Programmfragment in ein Flussdiagramm:

```
int a = 0;
int b = 100;
for(int i = 0; i < b; i++)
{
    if(i == 3)
    {
        continue;
    }
    a += i;
    if(a > 50)
    {
        break;
    }
}
```

Kreative Anwendung

7. Schreiben Sie eine C-Funktion `int zahlAusHexa(char h[])`, die einen nullterminierten C-String aus Symbolen 0 bis 9 und A bis F entgegen nimmt, als Hexadezimalzahl auffasst und den Wert als Zahl zurück liefert. Beispiel: Mit der Zeichenkette "1A" aufgerufen, soll die Funktion den Wert 26 zurückliefern. Achtung: Die Länge der Zeichenkette wird nicht gesondert übergeben. Die Funktion `strlen` ist erlaubt.
8. Welche Werte gibt diese C-Funktion zurück, wenn man $f(4)$, $f(5)$ beziehungsweise $f(6)$ aufruft? (Die drei Werte angeben, in dieser Reihenfolge.)

```
int f(int x)
{
    if(!(x > 5))
    {
        if(x != 4)
        {
            return 1;
        }
        return x;
    }
    else if(x < 6)
    {
        return 2;
    }
    return 3 * x;
}
```

9. Ein C-Programm enthält die folgende Funktion:

```
void f(int* a, int i)
{
    if(a[0] < 0)
    {
        return;
    }
    else if(a[0] == 13)
    {
        a[0] = i;
    }
    f(a + 1, i + 1);
}
```

Diese Funktion wird im Programm so aufgerufen:

```
int z[] = {5, 13, 42, 7, 13, -8};
f(z, 0);
```

Welche Zahlen stehen nach dem Aufruf im Array z?

10. Welche Werte stehen nach Ausführung dieser Zeilen C-Code in den Variablen x, y, z?

```
int a = 42;
int b[4] = {13, 7, 11};
int *c = &a;
int *d = &b[1];
d++;
(*d)++;
*c = 9;
int x = a;
int y = b[2];
int z = b[3];
```

11. Welcher Wert steht nach Ausführung des folgenden C-Code auf einem 16-Bit-Rechner in der Variablen x und welcher in der Variablen z? (Geben Sie möglichst auch Zwischenschritte an, damit Ihr Gedankengang nachvollziehbar ist.)

```
int x = 255 * 256 + 253;
int z = x + 3;
```

12. Schreiben Sie diese C-Funktion so um, dass man sofort sehen kann, was sie tut. (Geben Sie möglichst auch Zwischenschritte an, damit Ihr Gedankengang nachvollziehbar ist.)

```
int f(unsigned int x)
{
    if(x > 2)
    {
        int j = -1;
        for(int i = 0; i <= x; i += 3)
        {
            j++;
        }
        return j;
    }
    else
    {
        return 0;
    }
}
```