

Informatik 1 für Regenerative Energien

Klausur vom 28. September 2015

Jörn Loviscach

Versionsstand: 27. September 2015, 22:30



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 (auch nicht wearable), kein Handy und Ähnliches.

Name	Vorname	Matrikelnummer	E-Mail-Adresse

Fingerübungen

1. Welches Bitmuster (acht Bit mit Zweierkomplement) steht für die Zahl -2 ? Welches für die Zahl -10 ? Wie rechnet man dann $-2 + (-10)$ aus? Schreiben Sie das mit Bits hin.
2. Gegeben sind die C-Variablen `unsigned char a = 0x25;` und `unsigned char b = 0xB3;`. 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?

```
bool a = true;
float b = 20.0f;
if(3.0f + b * 2.0f < 5.0f || a && b - 101 % 2 >= 0.0f)
{
    // ...
}
```

4. Die folgende C-Funktion soll feststellen, ob die übergebene Zahl (nur Zahlen ab 2 aufwärts sind erlaubt) eine Primzahl ist, also eine Zahl, die nur durch 1 und durch sich selbst ohne Rest teilbar ist. Beispiele: Für 13 soll sie `true` zurückliefern; für 42 soll sie `false` zurückliefern. Es sind aber drei Fehler in der Funktion. Finden und korrigieren Sie diese.

```
bool istPrimzahl(unsigned int x) // Hat drei Fehler!
{
    unsigned int i = x - 1;
    while(i > 1)
    {
        if(x / i == 0)
        {
            return;
        }
        i++;
    }
    return true;
}
```

5. Gegeben sei die Datenstruktur Adresse:

```
struct Adresse
{
    char Nachname[16];
    char Vorname[16];
    int Postleitzahl;
    char Ort[16];
};
typedef struct Adresse Adresse;
```

Schreiben Sie eine C-Funktion

```
int zaehleAdressen(Adresse adressen[], int anzahl,
                  char anfangsbuchstabe,
                  int PLZanfang, int PLZende),
```

die als Ergebnis eine ganze Zahl zurückgibt – nämlich wie viele der Adressen im Array `adressen` einen Nachnamen mit dem Anfangsbuchstaben `anfangsbuchstabe` und obendrein eine Postleitzahl von `PLZanfang` bis einschließlich `PLZende` haben. In `anzahl` wird der Funktion die Anzahl der Einträge des Arrays `adressen` übergeben. Die Funktionen aus `string.h` können verwendet werden.

6. Ein Programm soll so lange Zahlen vom Benutzer abfragen, bis der eine 0 eingibt. Danach soll das Programm die größte und die kleinste eingegebene Zahl ausgeben (die 0 am Ende dabei nicht einbezogen). Falls sofort eine 0 eingegeben wird, soll das Programm eine Fehlermeldung ausgeben. Lösen Sie das mit einem Flussdiagramm. Dieses darf Eingabe, Ausgabe, Zuweisung, Addition, Multiplikation, Vergleich/Verzweigung benutzen, aber keine Arrays.

Kreative Anwendung

7. Schreiben Sie eine C-Funktion `void ABzuX(char a[])`, die einen Text (nullterminierten C-String) entgegen nimmt und darin alle Zeichenfolgen AB (nur in Großschreibung) durch X ersetzt, so dass der Text jeweils um einen Buchstaben kürzer wird. Achtung: Die Länge des Texts wird nicht gesondert übergeben. Die Funktionen aus `string.h` können verwendet werden. Lesen Sie nicht jenseits des Endes des Arrays mit der Zeichenkette.
8. Schreiben Sie dieses C-Programmfragment mit `if` statt mit `switch`:

```
int a;
int b;
// ...
switch(a)
{
case 3:
    a = b;
case 4:
case 5:
    a--;
    break;
case 6:
    a++;
    break;
default:
    a += b;
}
```

9. Diese Funktion ist gegeben:

```
int f(int x)
{
    if(x <= 1)
    {
        return 5;
    }
    return x * f(x - 1);
}
```

Was steht nach dem Aufruf `int a = f(3);` in der Variablen `a`? Geben Sie möglichst auch Zwischenschritte an, damit Ihr Gedankengang nachvollziehbar ist.

10. Welche Werte stehen nach Ausführung dieser Zeilen C-Code in den Variablen a, b, c?

```
int y = 7;
int* p = &y;
int z[4] = {15};
p[0] = 42;
p = z;
int a = *p;
p++;
int b = *p;
int c = y;
```

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

```
unsigned int x = 3 + 32 * 256;
unsigned int z = x * 8;
```

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 g(int x)
{
    int a = x * x;
    for(int i = 0; i < x * x; i++)
    {
        a--;
    }
    return a * x;
}
```