Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Technik & Architektur

Mikrocontroller, C-Programmierung

Selbststudium und Übungen 1

Peter Sollberger, V8.1

Am Ende dieser Semesterwoche haben Sie eine lauffähige C Entwicklungsumgebung und erste C Programme selbstständig geschrieben.
Sie kennen die elementaren Datentypen, Kontrollstrukturen und Schleifen.

1. Selbststudium

Gehen Sie nochmals die Folien durch.

2. Übungen

(zu diesen Übungen werden durch die Studierenden Musterlösungen präsentiert und abgegeben).

Übung 1.1: Entwurf Übung mit Funktionen und for - Schleife

Entwerfen Sie eine eigene Übung zum Thema Funktionen und for-Schleife, analog der Aufgabe 'Umrechnung Fahrenheit →Celsius mit Funktion'.

Sie entwerfen die Aufgabenstellung und erarbeiten die Musterlösung so wie in Kapitel 3, Weitere freiwillige Aufgaben.

Übung 1.2: Entwurf Übung mit Funktionen und while Schlaufe

Entwerfen Sie eine eigene Übung zum Thema Funktionen und while-Schleife, analog der Aufgabe 'Grösster gemeinsamer Teiler (ggT) mit while Schlaufe'.

Sie entwerfen die Aufgabenstellung und erarbeiten die Musterlösung so wie in Kapitel 3, Weitere freiwillige Aufgaben.

Übung 1.3: Integer nach Binary (iToBinary)

Schreiben Sie ein Programm, welches eine 32-Bit Integer-Zahl in Binärdarstellung ausgibt.

Der Wert 12 wird also zu `00000000000000000000000000001100`,
der Wert -13 zu `11111111111111111111111111110011`

Für die Eingabe einer Integerzahl können Sie folgendes Code-Fragment verwenden:

```
int  wert;

printf("Wert als Integer (auch negativ): ");
scanf("%d", &wert);
```

Die Variable `wert` enthält nun den Wert der Eingabe.

Hinweis: Um die Anzahl Bits einer Integer Variable zu berechnen können Sie folgende Funktion verwenden (`sizeof(…)` liefert dabei die Anzahl Bytes):

```
sizeof(unsigned int) * 8
```

Übung 1.4: Menu

Schreiben Sie das Gerüst zu einer textbasierten Menusteuerung.

Nach dem Starten des Programmes erscheint auf der Konsole eine Anzeige zur Selektion:

Durch Eingabe des entsprechenden Buchstabens kann dann die jeweilige Aktion (als Funktionsaufruf, hier z.B. die Funktionen `doA()`, `doB()` und `doC()`) aufgerufen werden.

Nach dem jeweiligen Funktionsaufruf wird der Anzeigetext erneut auf die Konsole ausgegeben und auf eine weitere Eingabe gewartet werden.

Mit der Eingabe von 'q' (oder 'Q') wird das Programm verlassen.

Folgende

```
A --> Select menu item A
B --> Select menu item B
C --> Select menu item C
Q --> Exit
Enter selection:
a
Menu A selected

A --> Select menu item A
B --> Select menu item B
C --> Select menu item C
Q --> Exit
Enter selection:
B
Menu B selected

A --> Select menu item A
B --> Select menu item B
C --> Select menu item C
Q --> Exit
Enter selection:
q
Exiting...
```

Hinweis: Die Anweisungen `scanf(..)`blockiert so lange, bis die Enter-Taste gedrückt wird.

```
char s[32]; // Erstellen eines  fixed-size Arrays für 32 Zeichen
…
// Wait for input
(void) scanf("%s", s);
```

Im fixed-size Array `s` sind dann die eingegebenen Zeichen enthalten. Mit `s[0]` erhalten Sie das erste eingegeben Zeichen.

3. Weitere freiwillige Aufgabe

Umrechnung Fahrenheit → Celsius mit Funktion

Schreiben Sie ein Programm welches Temperaturen in Fahrenheit in Grad Celsius umrechnet.

Regel: °C = (5/9) · (°F – 32)

Implementieren Sie diese Formel in einer Funktion.

Das Programm soll die Umrechnung für Werte zwischen 0 °F und 210 °F, in Schritten von 15 °F, machen. Dabei soll folgende Ausgabe erzeugt werden (linke Spalte: Temperatur in °F, rechte Spalte Temperatur in °C:

```
 0       -17.77777
15      -9.444445
30      -1.111111
45      7.222222
60      15.555556
75      23.888889
90      32.222225
105     40.555557
120     48.888893
135     57.222225
150     65.555557
165     73.888893
180     82.222229
195     90.555557
210     98.888893
```

Verwenden Sie dazu in der `main()` Funktion die richtige Schleife, rufen die Umrechnungsfunktion auf und machen die Ausgabe.

Grösster gemeinsamer Teiler (ggT) mit while Schlaufe

Im Modul PRG1 haben sie den Algorithmus von Euklid zur Bestimmung des grössten, gemeinsamen Teilers kennengelernt:

Ermittlung des grössten, gemeinsamen Teilers (ggT) zweier natürlicher Zahlen A und B:

1. *Sei A die grössere der beiden Zahlen A und B (vertauschen, falls dies nicht bereits so ist)*
2. *Setze A = A – B*
3. *Wenn A und B ungleich sind, dann fahre fort mit Schritt 1, wenn sie gleich sind, dann beende den Algorithmus: Diese Zahl ist der grösste, gemeinsame Teiler.*

Implementieren Sie diesen Algorithmus mit Hilfe einer `while()` Schlaufe.

Zum Testen Ihres Algorithmus mit Eingaben können Sie folgendes Code-Fragment verwenden:

```
int ersteZahl, zweiteZahl;

printf("Erste Zahl: ");
scanf("%d", &ersteZahl);
printf("Zweite Zahl: ");
scanf("%d", &zweiteZahl);
```

Die Variablen `ersteZahl` und `zweiteZahl` enthalten nun die Werte Ihrer Eingabe.

Wörter Zählen

Schreiben Sie ein Programm, das Wörter zählt und das Ergebnis ausgibt.

Dabei lesen Sie solange einzelne Zeichen ein (mit der Funktion getchar()), bis sie ein EOF Zeichen erhalten.

Als Trennzeichen zwischen den Worten gelten ' ' (Blanks), '\t' (Tabs) und '\n' (Zeilenumbrüche).

Zum Testen rufen Sie das Programm auf der Kommandozeile auf und verwenden die Standardeingabe aus einer Datei (z.B. CountWords.exe < CountWords.c)

Hinweis 1: Das EOF Zeichen geben Sie über die Tastatur mit der Tastenkombination 'Ctrl' + 'D' ein.

Hinweis 2: Verwenden Sie eine Zustandsvariable (mit den zwei Werten IN und OUT), welche Sie zum Merken verwenden, ob Sie in einem Wort sind oder nicht. Je nach eingelesenem Zeichen ändert Sich der Zustand. Wenn der Zustand auf IN wechselt, erhöhen Sie den Wörterzähler.

Eine Musterlösung zu diesen Aufgaben finden Sie auf der nächsten Seite. Probieren Sie aber selber zuerst, eine Lösung zu erarbeiten.
Sie wissen ja: Verstehen ist nicht das gleiche wie selber schreiben können.

Umrechnung Fahrenheit → Celsius mit Funktion Musterlösung (HCS08):

```
/* Copyright 2015 Hochschule Luzern - Technik & Architektur */

#include <hidef.h>    // for EnableInterrupts macro
#include "platform.h" // include peripheral declarations
#include "clock.h"
#include "sci.h"
#include <stdio.h>
#include <float.h>    // Remember to adjust linker library setting!

/**
 * Umwandlung von Fahrenheit in Celsius.
 */

#define RANGE_LOWER 0
#define RANGE_UPPER 210
#define RANGE_STEP  15

/**
 * Converts the passed temperature in fahrenheit to a value in celsius.
 * @param  fahrenheit Temperature in Fahrenheit
 * @return corresponding temperature in celsius
 */
float convertToCelsius(float fahrenheit);

void main(void) {
        float f, c;
        char s[32];

        initClock();        // Init 24 MHz clock and disable watchdog
        sci2Init(9600); // Initialise the serial interface for stdio in- and output
        EnableInterrupts;

        for (;;) {
                for (f = RANGE_LOWER; f <= RANGE_UPPER; f += RANGE_STEP) {
                        c = convertToCelsius(f);
                        (void) printf("%d\t%f\n", (int) f, c);
                }

                (void) printf("\nPress enter to continue");
                (void) scanf("%s", s);
                (void) printf("\n\n");
        }
}

float convertToCelsius(float fahrenheit) {
        float c;

        c = (5.0f / 9.0f) * (fahrenheit - 32.0f);

        return c;
}
```

Grösster gemeinsamer Teiler (ggT) mit while Schlaufe Musterlösung (HCS08):

```
/* Copyright 2015 Hochschule Luzern - Technik & Architektur */

#include <hidef.h>    // for EnableInterrupts macro
#include "platform.h" // include peripheral declarations
#include "clock.h"
#include "sci.h"
#include <stdio.h>

/**
 * Bestimmen des ggT von zwei Zahlen.
 * @author Peter Sollberger <peter.sollberger at hslu.ch>
 */

/**
 * Berechnet den grössten gemeinsamen Teiler der beiden Zahlen a und b
 * mit dem Verfahren von Euklid.
 */
int ggT(int a, int b);

/**
 * Einlesen zweier Zahlen, ggt berechnen und Resultat ausgeben.
 */
void main(void) {
    int ersteZahl, zweiteZahl, t;

        initClock();        // Init 24 MHz clock and disable watchdog
        sci2Init(9600); // Initialise the serial interface for stdio in- and output
        EnableInterrupts;

        for (;;) {
                (void) printf("Erste Zahl: ");
                (void) scanf("%d", &ersteZahl);
                (void) printf("\nZweite Zahl: ");
                (void) scanf("%d", &zweiteZahl);

                t = ggT(ersteZahl, zweiteZahl);

                (void) printf("\nggT von %d und %d ist %d\n\n", ersteZahl,
zweiteZahl, t);
        }
}

int ggT(int a, int b) {
        while (a != b) {
                if (a > b) {
                        a -= b;
                } else {
                        b -= a;
                }
        }
        return a;
}
```

CountWords Musterlösung (x86):

```
/* Copyright 2010 Hochschule Luzern - Technik & Architektur */

/**
 * Zählt die Anzahl Wörter.
 * Als Trennzeichen werden ' ' (Blanks), '\t' (Tabs) und '\n' (Zeilenumbrüche)
 * genommen.
 * @author: Peter Sollberger
 */

#include <stdio.h>

#define IN  1
#define OUT 2

int main(int argc, char** argv) {
    int c;
    int wordCount = 0;
    int state = OUT;

    while ((c = getchar()) != EOF) {
        if ((c == ' ') || (c == '\t') || (c == '\n')) {
            state = OUT;
        } else if (state == OUT) {
            wordCount++;
            state = IN;
        }
    }
    printf("Anzahl Woerter: %d\n", wordCount);

    return 0;
}
```