

Hardwarenahe Programmierung

Übungsaufgaben – 7. November 2019

Diese Übung enthält Punkteangaben wie in einer Klausur. Um zu „bestehen“, müssen Sie innerhalb von 45 Minuten unter Verwendung ausschließlich zugelassener Hilfsmittel 8 Punkte (von insgesamt 16) erreichen.

Aufgabe 1: Arrays mit Zahlen

Wir betrachten das folgende Programm
(Datei: [aufgabe-1.c](#)):

```
#include <stdio.h>

void f (int *s0, int *s1)
{
    while (*s0 >= 0)
    {
        int *s = s1;
        while (*s >= 0)
            if (*s0 == *s++)
                printf ("%d_", *s0);
        s0++;
    }
    printf ("\n");
}

int main (void)
{
    int a[] = { 10, 4, 3, 7, 12, 0, 1, -1 };
    int b[] = { 7, 14, 0, 8, 9, 22, 10, -1 };
    f (a, b);
    return 0;
}
```

- (a) Was bewirkt die Funktion `f`, und wie funktioniert sie? (4 Punkte)
- (b) Was passiert, wenn Sie beim Aufruf der Funktion für einen der Parameter den Wert `NULL` übergeben? Begründen Sie Ihre Antwort. (2 Punkte)
- (c) Was kann passieren, wenn Sie das Hauptprogramm wie folgt abändern ([aufgabe-1c.c](#))? Begründen Sie Ihre Antwort.

```
int main (void)
{
    int a[] = { 10, 4, 3, 7, 12, 0, 1 };
    int b[] = { 7, 14, 0, 8, 9, 22, 10 };
    f (a, b);
    return 0;
}
```

(2 Punkte)

Aufgabe 2: Datum-Bibliothek

Schreiben Sie eine Bibliothek (= Sammlung von Deklarationen und Funktionen) zur Behandlung von Datumsangaben.

Diese soll enthalten:

- einen `struct`-Datentyp `date`, der eine Datumsangabe speichert,
- eine Funktion `void date_print (date *d)`, die ein Datum ausgibt,
- eine Funktion `int date_set (date *d, int day, int month, int year)`, die ein Datum auf einen gegebenen Tag setzt und zurückgibt, ob es sich um ein gültiges Datum handelt (0 = nein, 1 = ja),
- eine Funktion `void date_next (date *d)`, die ein Datum auf den nächsten Tag vorrückt.

Schreiben Sie auch ein Programm, das die o. a. Funktionen testet.

(8 Punkte)