

Hardwarenahe Software-Entwicklung – Klausur – 20. Juli 2012

Prof. Dr. Peter Gerwinski, Sommersemester 2012

Name:	
Matrikel-Nr.:	

Zeit: 120 Minuten

Zulässige Hilfsmittel:

- Schreibgerät
- Beliebige Unterlagen in Papierform
- Beliebige Unterlagen in elektronischer Form
(Skripte, Dokumentation, Beispiel-Programme, Software)
- Von der Hochschule zur Verfügung gestellte Rechner

Nur die o. a. zulässigen Hilfsmittel dürfen sich während der Klausur im Arbeitsbereich befinden. Mobiltelefone, Geräte mit mobilem Internet-Zugang u. ä. sind auszuschalten und in der Tasche zu verstauen.

Bewertung:

Aufgabe 1 (a)	/2	Aufgabe 2 (a)	/3
(b)	/5	(b)	/6
(c)	/2	(c)	/2
(d)	/4	(d)	/2
(e)	/3	Aufgabe 3	/10
(f)	/3	Summe:	/42

Bei besonderen Leistungen sind Zusatzpunkte möglich. Mit 20 erreichten Punkten gilt die Klausur als bestanden.

Aufgabe 1: Liste von Namen

Wir betrachten das folgende C-Programm:

```
#include <stdio.h>
#include <string.h>
```

```
char *name[10];
int names = 0;

void insert_name (char *new_name)
{
    int i = 0;
    while (i < names && strcmp (new_name, name[i]) < 0)
        i++;
    int k = names;
    while (k > i)
    {
        name[k] = name[k - 1];
        k--;
    }
    name[i] = new_name;
    names++;
}
```

```
int main (void)
{
    insert_name ("Otto");
    insert_name ("Lisa");
    insert_name ("Anna");
    insert_name ("Heinrich");
    insert_name ("Zacharias");
    insert_name ("Siegfried");
    insert_name ("Dieter");
    insert_name ("Hans");
    insert_name ("Berta");
    insert_name ("Maria");

    for (int i = 0; i < names; i++)
        printf ("%s\n", name[i]);

    return 0;
}
```

(a) Was bewirkt das Programm? (2 Punkte)

(b) Erklären Sie, wie das Programm funktioniert. (5 Punkte)

- (c) Durch welche minimale Änderung kann man die Reihenfolge, in der das Programm die Namen ausgibt, genau umdrehen? (2 Punkte)

- (d) Das Programm enthält einen gravierenden Fehler. Was kann passieren, wenn man im Hauptprogramm einen weiteren Aufruf von `insert_name()` hinzufügt (z. B. `insert_name("Ulrich");`), und warum? Skizzieren Sie, wie man das Programm ändern muß, um dies zu verhindern. (4 Punkte)

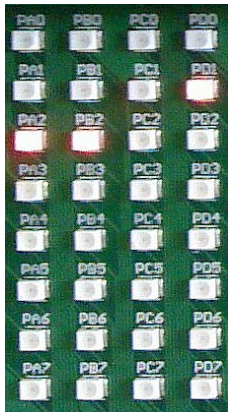
- (e) Welches Laufzeitverhalten $\mathcal{O}(?)$ hat das Programm hinsichtlich der Anzahl der Vergleiche (Aufrufe der Funktion `strcmp()`) in Abhängigkeit von der Anzahl n der Namen und warum? (3 Punkte)

- (f) Beschreiben Sie (in Worten oder als C-Quelltext) eine mögliche Optimierung für die Funktion `insert_name()`, die das Programm hinsichtlich der Vergleiche (Aufrufe von `strcmp()`) das Laufzeitverhalten $\mathcal{O}(n \log n)$ annehmen läßt. (3 Punkte)

Aufgabe 2: Mikro-Controller

An die vier Ports eines ATmega16-Mikro-Controllers sind Leuchtdioden angeschlossen:

- von links nach rechts
an die Ports A, B, C und D,
- von oben nach unten
an die Bits Nr. 0 bis 7.



Wir betrachten das folgende C-Programm für diesen Mikro-Controller:

```
#include <avr/io.h>
#include <avr/interrupt.h>

int counter = 0;

ISR (TIMER0_COMP_vect)
{
    PORTA = 1 << ((counter++ >> 6) & 7);
}

int main (void)
{
    cli ();
    TCCR0 = (1 << CS01) | (1 << CS00);
    TIMSK = 1 << OCIE0;
    sei ();
    DDRA = 0xff;
    while (1);
    return 0;
}
```

Das Programm bewirkt ein periodisches Laufflicht in der linken Spalte von oben nach unten. Eine Animation davon wurde Ihnen zu Beginn der Klausur vorgeführt. Sie finden diese auch in der Datei [aufgabe-2.gif](#).

- (a) Wieso bewirkt das Programm überhaupt etwas, wenn doch das Hauptprogramm nach dem Initialisieren lediglich eine Endlosschleife ausführt, in der *nichts* passiert? (3 Punkte)

- (b) Erklären Sie, wie die Anweisung

```
PORTA = 1 << ((counter++ >> 6) & 7);
```

das LED-Blinkmuster hervorruft. (6 Punkte)

Hinweis: Zerlegen Sie die eine lange Anweisung in mehrere kürzere.

Wenn nötig, verwenden Sie zusätzliche Variable für Zwischenergebnisse.

(c) Was bedeutet „ISR (TIMER0_COMP_vect)“? (2 Punkte)

(d) Wieso leuchten die Leuchtdioden PB2 und PD1? (2 Punkte)

Aufgabe 3: Collatz-Vermutung

Wir betrachten Zahlenfolgen folgenden Typs:

- Wir beginnen mit irgendeiner natürlichen Zahl $n > 0$.
- Wenn die Zahl gerade ist, lautet die nächste Zahl: $n / 2$
- Wenn die Zahl ungerade ist, lautet die nächste Zahl: $3 \cdot n + 1$

Die Collatz-Vermutung lautet:

Jede auf diese Weise konstruierte Zahlenfolge erreicht schließlich die Zahl 1.
(Diese Vermutung konnte bis heute weder bewiesen noch widerlegt werden.)

Beispiel: Für $n = 24$ ergibt sich die Zahlenfolge: 24, 12, 6, 3, 10, 5, 16, 8, 4, 2, 1.

Schreiben Sie eine C-Funktion, die eine natürliche Zahl $n > 0$ als Parameter entgegennimmt und ihre Collatz-Zahlenfolge ausgibt. (10 Punkte)

