

Angewandte Informatik – Klausur – 10. Juli 2013

Prof. Dr. Peter Gerwinski, Wintersemester 2012/13

Name:	
Matrikel-Nr.:	

Zeit: 120 Minuten

Zulässige Hilfsmittel:

- Schreibgerät
- Beliebige Unterlagen in Papierform und/oder auf Datenträgern
- Elektronische Rechner (Notebook, Taschenrechner o. ä.)
ohne Zugang zu Datennetzen jeglicher Art

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

Die reguläre Maximalpunktzahl beträgt 42 Punkte.
Bei besonderen Leistungen sind Zusatzpunkte möglich.
Mit 20 erreichten Punkten gilt die Klausur als bestanden.

Aufgabe 1: Strings

Strings werden in der Programmiersprache C durch Zeiger auf **char**-Variable realisiert.

Wir betrachten die folgende Funktion:

```
int fun_1 (char *s1, char *s2)
{
    int result = 1;
    for (int i = 0; s1[i] && s2[i]; i++)
        if (s1[i] != s2[i])
            result = 0;
    return result;
}
```

(a) Was bewirkt die Funktion? (3 Punkte)

(b) Welchen Sinn hat die Bedingung „s1[i] && s2[i]“ in der **for**-Schleife? (2 Punkte)

(b) Was würde sich ändern, wenn die Bedingung „s1[i] && s2[i]“ in der **for**-Schleife zu „s1[i]“ verkürzt würde? (3 Punkte)

- (c) Von welcher Ordnung (Landau-Symbol) ist die Funktion `fun_1()` hinsichtlich der Anzahl ihrer Zugriffe auf die Zeichen in den Strings – und warum? (2 Punkte)

- (d) Schreiben Sie eine eigene Funktion, die dieselbe Aufgabe erledigt wie `fun_1()`, nur effizienter, und geben Sie die Ordnung (Landau-Symbol) der von Ihnen geschriebenen Funktion an. (5 Punkte)

Aufgabe 2: Trickprogrammierung

Wir betrachten das folgende Programm **hi.c**:

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

int main (void)
{
    uint64_t x = 5221205178585213256;
    char *s = &x;
    printf ("%s\n", s);
    return 0;
}
```

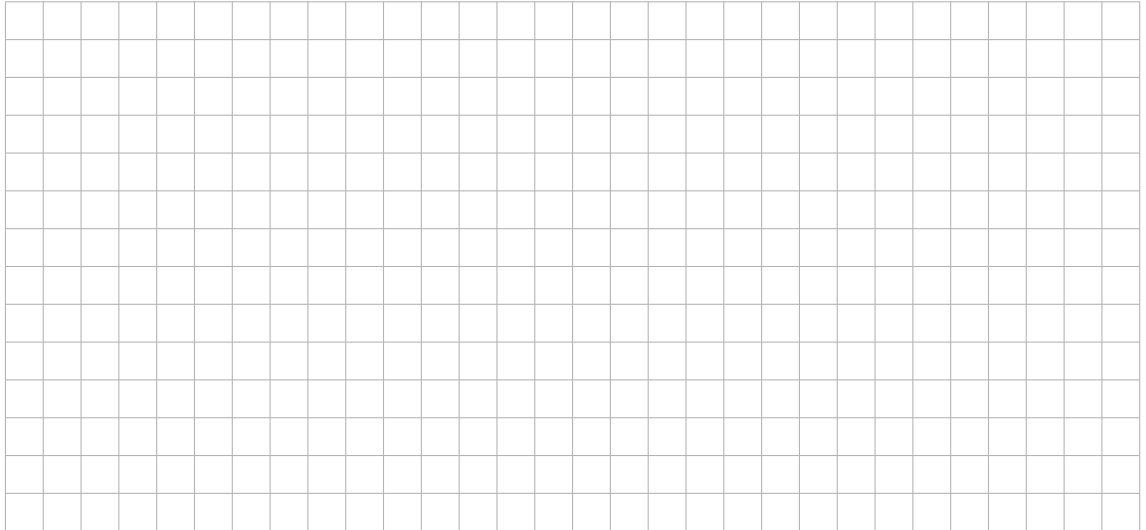
Das Programm wird compiliert und auf einem 64-Bit-Little-Endian-Computer ausgeführt:

```
$ gcc -Wall -O hi.c -o hi
hi.c: In function 'main':
hi.c:7:13: warning: initialization from incompatible pointer type [...]
$ ./hi
Hi
```

- (a) Erklären Sie die Warnung beim Compilieren. (1 Punkt)

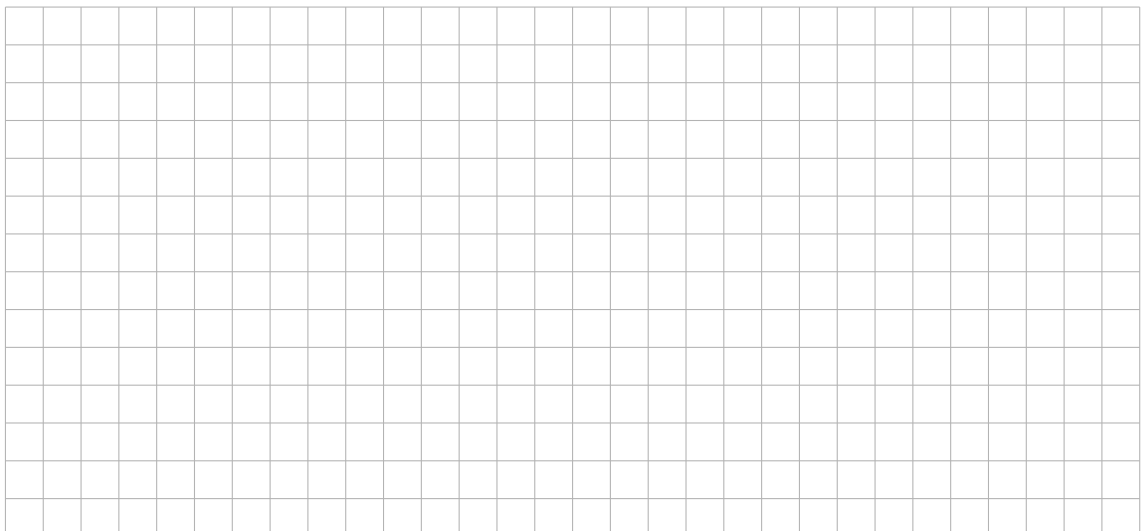
- (b) Erklären Sie die Ausgabe des Programms. (3 Punkte)

- (a) Warum sieht man lediglich ein leeres Fenster? Welchen Befehl muß man ergänzen, um diesen Fehler zu beheben? (4 Punkte)

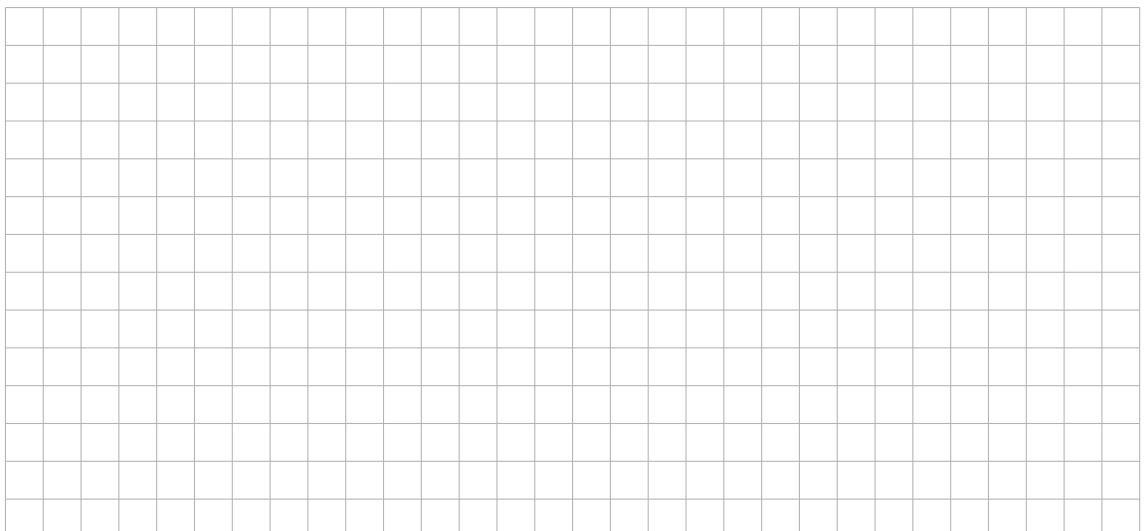


Nach der Fehlerbehebung in Aufgabenteil (a) zeigt das Programm einen sich drehenden Würfel.
(Sie können sich diesen während der Klausur jederzeit vorführen lassen.)

- (b) Erklären Sie das Drehverhalten des Würfels. (4 Punkte)



- (c) Welche Befehle muß man ergänzen, um eine gleichförmige Drehung zu erhalten? (4 Punkte)



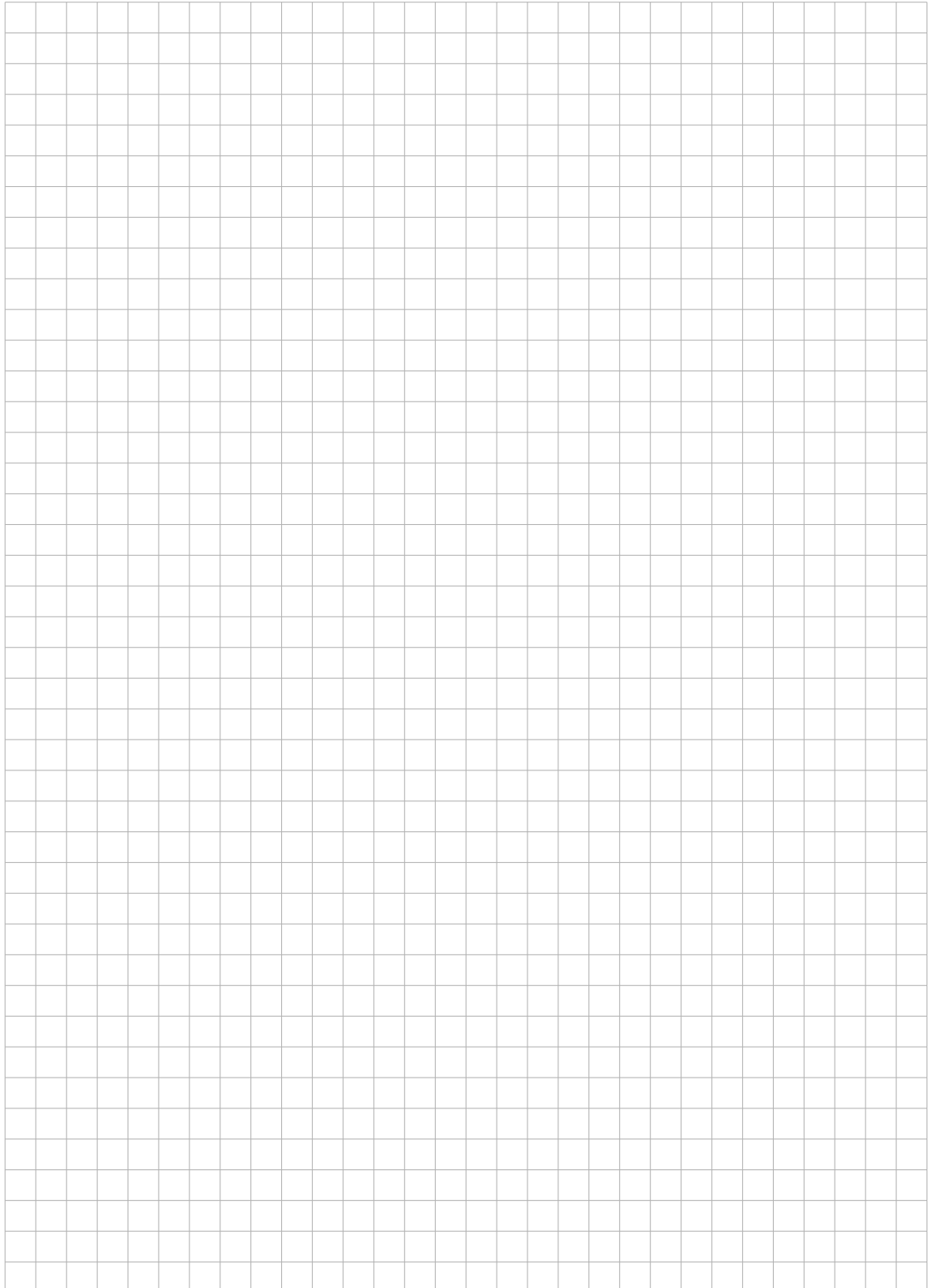
Aufgabe 4: Ausgabe von Hexadezimalzahlen

Schreiben Sie eine Funktion `void print_hex (uint32_t x)`, die eine gegebene vorzeichenlose 32-Bit-Ganzzahl `x` als Hexadezimalzahl ausgibt.

Verwenden Sie dafür *nicht* `printf()` mit der Formatspezifikation `%x` als fertige Lösung, sondern programmieren Sie die nötige Ausgabe selbst.

Die Verwendung von `printf()` mit anderen Formatspezifikationen wie z. B. `%d` oder `%c` oder `%s` ist hingegen zulässig.

(8 Punkte)

A large grid of graph paper, consisting of 20 columns and 30 rows of small squares, intended for the student to write their code solution.