Ingenieurinformatik

C-Programmierung

- ❒ **Bachelor-Studiengang, neue SPO**
- ❒ **Bachelor-Studiengang, alte SPO (Kombinationsprüfung)**
- ❒ **Diplomstudiengang**

Name	Vorname	Matrikelnummer	Sem.-Gr.:	Hörsaal	Platz

Zulassung geprüft:

_____

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Summe

Die Prüfung ist nur dann gültig, wenn Sie die erforderliche Zulassungsvoraussetzung erworben haben (erfolgreiche Teilnahme am Praktikum). Dies wird vom Aufgabensteller überprüft.

Aufgabensteller: **Dr. Reichl, Dr. Küpper und Kollegen**

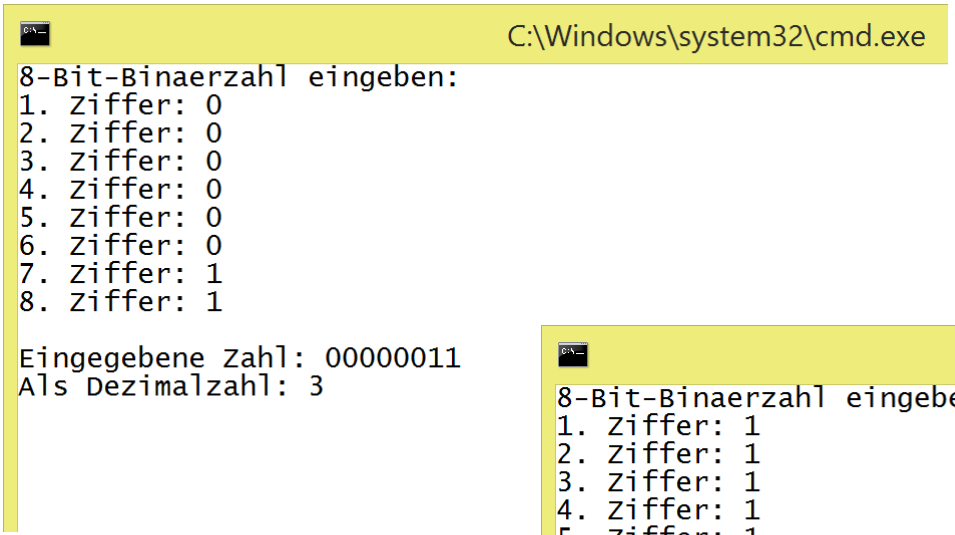
Bearbeitungszeit: **60 Minuten**

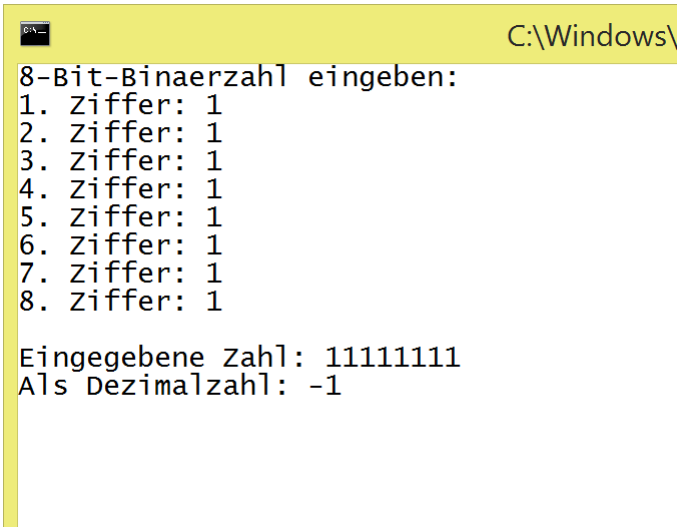
Hilfsmittel: **Taschenrechner nicht zugelassen,**
PC/Notebook nicht zugelassen,
sonstige eigene Hilfsmittel sind erlaubt,
Bearbeitung mit Bleistift ist erlaubt.

Aufgabe 1: (ca. 26 Punkte)

Schreiben Sie ein C-Programm, welches den dezimalen Wert einer 8-Bit-Binärzahl in Zweierkomplementdarstellung ermittelt und auf dem Bildschirm ausgibt:

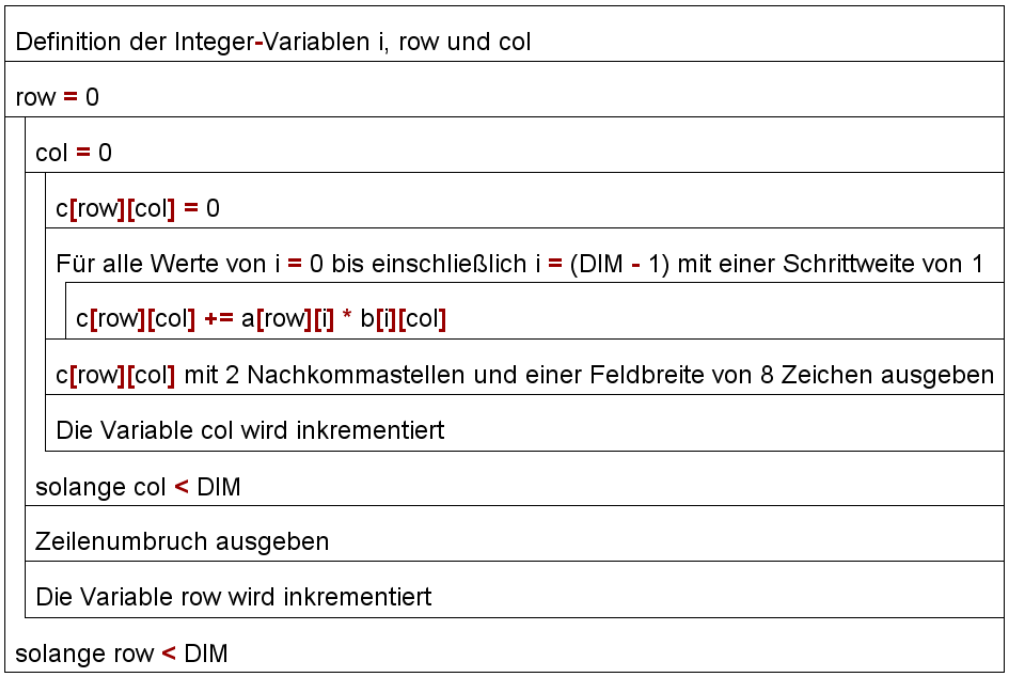
- Die Ausgabe Ihres C-Programms auf dem Bildschirm soll so aussehen, wie in den Bildschirmfotos unten auf dieser Seite gezeigt.
- Der Benutzer gibt eine 8-Bit-Binärzahl Ziffer für Ziffer ein, beginnend mit dem Wert für das höchstwertige Bit. Die einzelnen Ziffern werden in einem int-Vektor geeigneter Länge gespeichert: Im Vektor steht die Ziffer mit der niedrigsten Wertigkeit an Position 0, die mit der nächsthöheren an Position 1 usw.
- Die Eingaben des Anwenders müssen nicht auf Korrektheit geprüft werden.
- Die eingegebene Binärzahl wird zunächst zur Kontrolle einzeilig ausgegeben.
- Anschließend wird der dezimale Wert der eingegebenen Binärzahl ermittelt und auf dem Bildschirm ausgegeben.
- Beachten Sie dabei, dass das höchstwertige Bit der Binärzahl das Vorzeichen angibt:
  - Um den dezimalen Wert von positiven Zahlen zu ermitteln, können die Wertigkeiten der einzelnen Bits einfach aufaddiert werden (1er-, 2er-, 4er, 8er-Stelle usw.).
  - Um den dezimalen Wert von negativen Zahlen zu ermitteln, muss das Zweierkomplement der eingegebenen Binärzahl gebildet werden (also alle Bits invertieren, 1 addieren).
  - Tipp: Es ist einfacher zu programmieren, wenn bei negativen Zahlen zunächst alle Bits invertiert werden, anschließend der dezimale Wert der Zahl berechnet wird und erst danach zum dezimalen Wert noch 1 addiert und mit -1 multipliziert wird.





Aufgabe 2: (ca. 22 Punkte)

Das folgende Struktogramm zeigt den Aufbau eines Programms zur Multiplikation von zwei global definierten **quadratischen Matrizen a und b**. Die symbolische Konstante DIM legt die Dimension der Matrizen fest. Während der Berechnung werden Zwischenergebnisse ausgegeben.



2.1. Vervollständigen Sie den C-Quelltext der Funktion **main** auf der folgenden Seite so, dass er genau dem abgebildeten Struktogramm entspricht.

2.2. Wie lautet die Ausgabe des Programms auf dem Bildschirm, wenn die beiden quadratischen Matrizen a und b mit den im Quelltext (auf der folgenden Seite) gezeigten Werten belegt sind?

```c
#include <stdio.h>
#define DIM 3

double a[DIM][DIM] = { { 1.1,  2.2,  3.3 },
                       { 4.4,  5.5,  6.6 },
                       { 7.7,  8.8,  9.9 } };

double b[DIM][DIM] = { { 1.0,  0.0,  0.0 },
                       { 0.0,  2.0,  0.0 },
                       { 0.0,  0.0, -1.0 } };

double c[DIM][DIM];

int main(void)
{









    return 0;
}
```

Aufgabe 3: (ca. 19 Punkte)

3.1. Das folgende C-Programm generiert 100 ganzzahlige Zufallszahlen im Bereich 0 … 9. Auf dem Bildschirm wird anschließend ausgegeben, wie oft dabei jede Zufallszahl aufgetreten ist. Korrigieren Sie die fünf Fehler, die sich in den Quelltext eingeschlichen haben.

```
#include <studio.h>
#include <stdlib.h>

int main(void)
{
    int i, zuf;
    int anzahl[10] =
    {
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    };

    for(i = 0; i < 100; ++i)
    {
        /* Zufallszahl im Bereich 0...9 erzeugen */
        zuf = rand % 10;
        anzahl[zuf] += 1;
    }

    for(i = 0; i <= 10; i++)
    {
        printf("Zufallszahl %d: Anzahl = %d\n", i, anzahl);
    }

    return 0;
```

3.2. Wie lautet die Ausgabe des folgenden Programms?

```
#include <stdio.h>

int main(void)
{
    int i = 1;

    while(i > 0)
    {
        ++i;
    }

    printf("%d", i);
    return 0;
}
```

Ausgabe des Programms:

3.3. Wie lautet die Ausgabe des folgenden Programms?

```c
#include <stdio.h>
int count(char *str);

int main(void)
{
    printf("%d\n", count("abc 123 XYZ"));
    printf("%d\n", count("abc*123*XYZ"));
    printf("%d\n", count("abc+123+XYZ"));
    printf("%d\n", count("abc+123*XYZ"));
    printf("%d\n", count("abc123XYZ"));
    return 0;
}

int count(char *str)
{
    int result;
    for(result = 0; *str != 0; ++str)
        if(*str == '*' || *str == '+') ++result;
    return result;
}
```

Ausgabe des Programms:

3.4. Fügen Sie die Deklaration (Prototyp) und die Definition der Funktion „random“ zum Quelltext hinzu: Die Funktion „random“ berechnet zwei double-Zufallszahlen im Bereich von +0,25…+0,75 (Grenzen eingeschlossen) und gibt beide Zufallszahlen per Zeiger ans Hauptprogramm zurück.

```c
#include <stdlib.h>
#include <stdio.h>

/* Deklaration der Funktion "random"... */


int main(void)
{
    int i;
    double z1, z2;
    for(i = 1; i < 10; ++i)
    {
        random(&z1, &z2);
        printf("%.2f, %.2f\n", z1, z2);
    }
    return 0;
}

/* Definition der Funktion "random"... */
```

(Platz für Notizen und Nebenrechnungen)