

# Ingenieurinformatik

Name	Vorname	Matrikelnummer	Sem.-Gr.:	Hörsaal	Platz

Zulassung geprüft  
vom Aufgabensteller:

Teil 1	Aufg. 2	Aufg. 3	Aufg. 4	Summe	Note

**Teil 1/Aufgabe 1:** 30 Minuten ohne Unterlagen, **Teil 2/Aufgaben 2-4:** 60 Minuten, beliebige eigene Unterlagen aber keine PC/Laptops, Bearbeitung mit Bleistift erlaubt.

**Die Prüfung ist nur dann gültig, wenn Sie die erforderliche Zulassungsvoraussetzung erworben haben (drei Testate im Praktikum). Dies wird vom Aufgabensteller überprüft.**

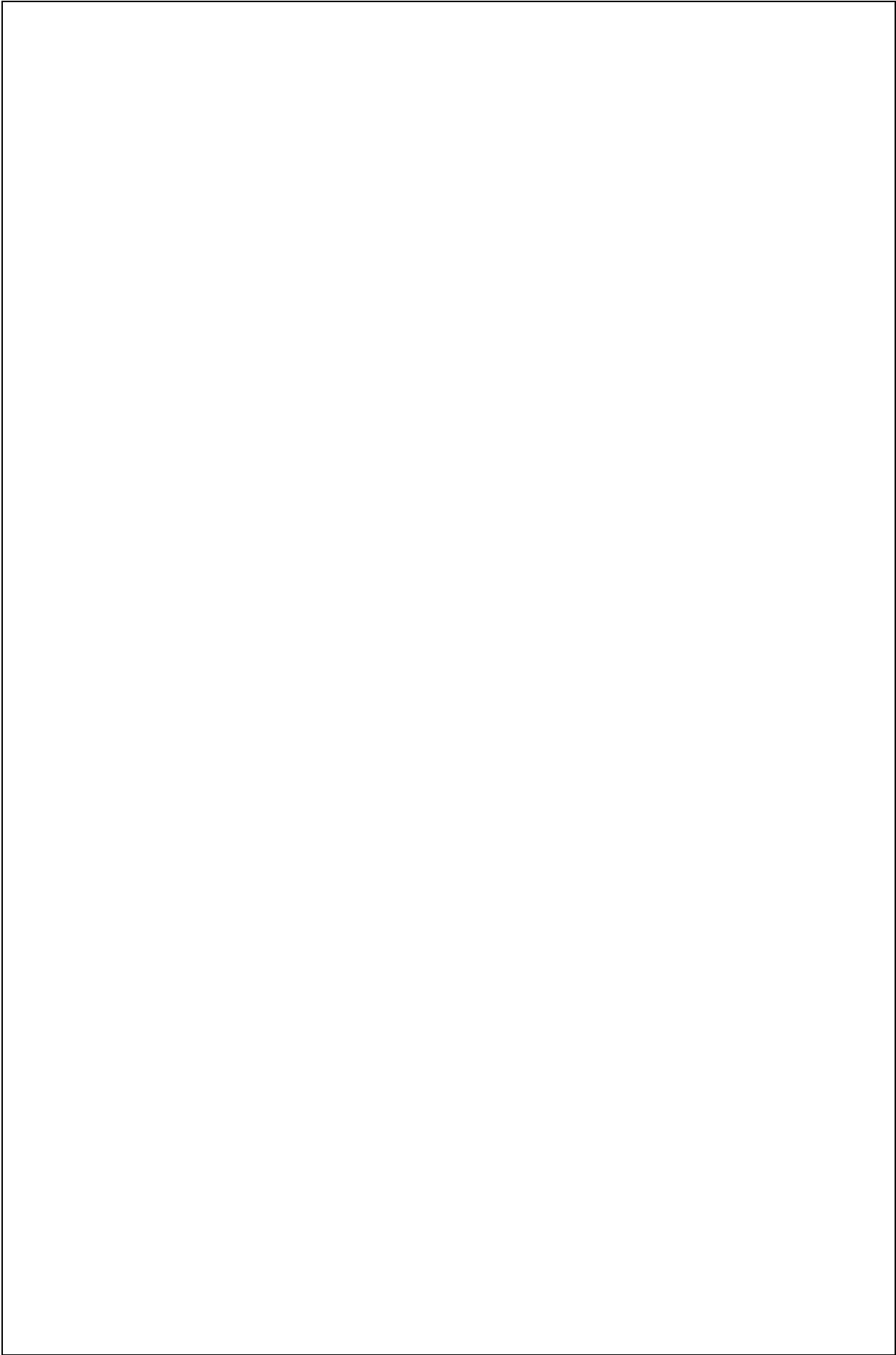
**Aufgabe 2: (ca. 24 Punkte)**

Die Funktion **rand()** aus der Bibliothek **stdlib.h** dient zur Erzeugung von Zufallszahlen. Mit einem C-Programm soll untersucht werden, ob die generierten Zufallszahlen gleichmäßig über den gesamten Zahlenbereich verteilt sind. Schreiben Sie den Quelltext des C-Programms auf die nebenstehende leere Seite!

1. Generieren Sie zunächst 500 ganzzahlige Zufallszahlen im Bereich von 0 bis einschließlich 9.
2. Das Programm merkt sich – zum Beispiel in einem geeigneten Vektor mit zehn Elementen – wie oft jede der zehn möglichen Zufallszahlen aufgetreten ist.
3. Schließlich wird das Ergebnis – wie in der Abbildung gezeigt – auf dem Bildschirm ausgegeben: Es wird zu jeder der zehn möglichen Zufallszahlen die gefundene Anzahl auf zwei Arten ausgegeben. Einmal als Zahlenwert und einmal in Form eines „Balkens“ aus Doppelkreuzen (Raute-Symbol, #).

```

C:\Windows\system32\cmd.exe
Zahl 0, Anzahl 44: #####
Zahl 1, Anzahl 50: #####
Zahl 2, Anzahl 56: #####
Zahl 3, Anzahl 52: #####
Zahl 4, Anzahl 49: #####
Zahl 5, Anzahl 50: #####
Zahl 6, Anzahl 49: #####
Zahl 7, Anzahl 48: #####
Zahl 8, Anzahl 54: #####
Zahl 9, Anzahl 48: #####
    
```



### Aufgabe 3: (ca. 24 Punkte)

Das folgende Programm simuliert den Anlauf eines Elektromotors und zeigt den zeitlichen Verlauf der Motordrehzahl in einem Diagramm.

```
#define _USE_MATH_DEFINES

#include <math.h>
#include <stdio.h>
#include "chart.h" /**/

#define DT = 0.0001 /* Simulations-Zeitschritt */

int main(void);
{
    int i;
    double u, w, t, t7000;

    w = 0;
    t = 0;
    t7000 = 0;

    for(i = 0, i <= 500; ++i)
    {
        /* Aktuelle Betriebsspannung */
        u = 42.0 * t / 0.01;
        if(u > 42.0) { u = 42.0;

        /* Zeit, Winkelgeschwindigkeit berechnen */
        t += DT;
        w += DT * 3567.2 * (u - w * 0.05236);

        /* Auswertung */
        chart_lineto(t, w, CHART_BLUE); /**/
        if(w / 2 / M_PI < 7000.0 / 60.0) t7000 = t;
    }

    if(t7000 < t)
        printf("7000/min nach %.3ds.\n", t7000);
    else
        printf("7000/min nicht erreicht!\n");

    chart_show(CHART_DEFAULT, "DC-Motor"); /**/
    return 0;
}
```

3.1. Korrigieren Sie die fünf Fehler, die sich in den abgebildeten Quelltext eingeschlichen haben. Hinweis: In den mit `/**/` markierten Zeilen befinden sich keine Fehler.

3.2. Wie oft wird die for-Schleife durchlaufen? Welche Werte haben die Variablen `i` und `u`, nachdem die for-Schleife verlassen wurde?

Anzahl der Durchläufe =

`i` =

`u` =

3.3. Zeichnen Sie für das (korrigierte) Hauptprogramm ein Struktogramm, welches den genauen Ablauf der Funktion `main` beschreibt.

Symb. Konstante:	DT = 0,0001
int-Variable:	i
double-Variablen:	u, w, t, t7000

**Aufgabe 4: (ca. 19 Punkte)****Name:** \_\_\_\_\_

- 4.1. Eine globale Matrix ist als **double mat[10][10]**; definiert und bereits mit Werten belegt. Schreiben Sie eine Funktion **min\_max**, welche die beiden Elemente mit dem größten und dem kleinsten Betrag (!) ermittelt. Von diesen beiden Elementen sollen die Beträge (!) mithilfe von Zeigern zurückgegeben werden.

- 4.2. Wie lautet die Ausgabe des folgenden C-Programms?

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

int main(void)
{
    int n;
    n = count("abcabcabc", 100, 'a'); printf("%d\n", n);
    n = count("abcabcabc", 5, 'a'); printf("%d\n", n);
    n = count("abcabcabc", 1, 'a'); printf("%d\n", n);
    n = count("abcabcabc", 0, 'a'); printf("%d\n", n);
    n = count("abcabcabc", 100, '#'); printf("%d\n", n);
    return 0;
}

int count(char *str, int max, char search)
{
    int pos, result = 0;
    for(pos = 0; str[pos] != 0; ++pos)
    {
        if(pos >= max) break;
        if(str[pos] == search) ++result;
    }
    return result;
}
```

**Ausgabe:**

- 4.3. Wie lautet die Ausgabe des folgenden C-Programms? (Nicht nur die Zahlenwerte sondern die vollständige Ausgabe der printf-Befehle hinschreiben!)

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

int main(void)
{
    double vekt1[DIM] = { 1.0, 1.5, 2.0 };
    double vekt2[DIM] = { 1.0, 0.0, 1.0 };
    double vprod[DIM], sprod = 0;
    int i;

    for(i = 0; i < 3; ++i)
        sprod += vekt1[i] * vekt2[i];
    printf("sprod = %.1f\n", sprod);

    vprod[0] = vekt1[1] * vekt2[2] - vekt1[2] * vekt2[1];
    vprod[1] = vekt1[2] * vekt2[0] - vekt1[0] * vekt2[2];
    vprod[2] = vekt1[0] * vekt2[1] - vekt1[1] * vekt2[0];

    for(i = 0; i < 3; ++i)
        printf("vprod[%d] = %.1f\n", i, vprod[i]);
    return 0;
}
```

**Ausgabe:**

Wie nennt man in der linearen Algebra die beiden Vektor-Operationen, die von dem abgebildeten C-Programm ausgeführt werden?