

Ingenieurinformatik

Name	Vorname	Matrikelnummer	Sem.-Gr.:	Hörsaal	Platz

Zulassung geprüft
vom Aufgabensteller:

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

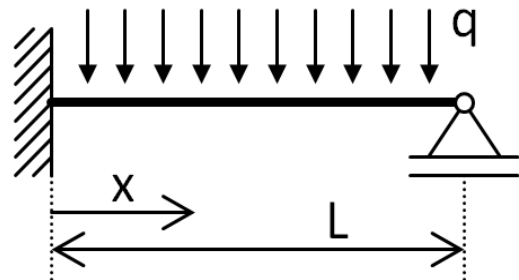
Teil 1/Aufgabe 1: 30 Minuten ohne Unterlagen, **Teil 2/Aufgaben 2-4:** 60 Minuten, beliebige eigene Unterlagen aber keine PC/Laptops, Bearbeitung mit Bleistift erlaubt.

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

Bei welchem Prüfer sind Sie angemeldet? Glück Jäger-Hezel Küpper

Aufgabe 2: (ca. 23 Punkte)

Ein Balken der Länge $L = 0,1$ m ist links fest eingespannt und rechts gelenkig gelagert. Der gesamte Balken wird mit der Streckenlast q belastet.

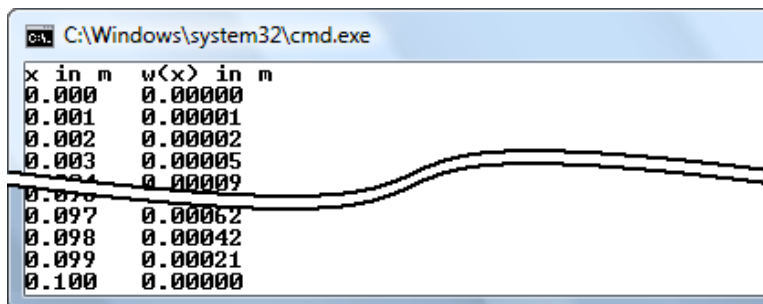


Aufgrund der Belastung biegt sich der Balken nach unten durch. Für die Durchbiegung $w(x)$ an der Stelle x gilt:

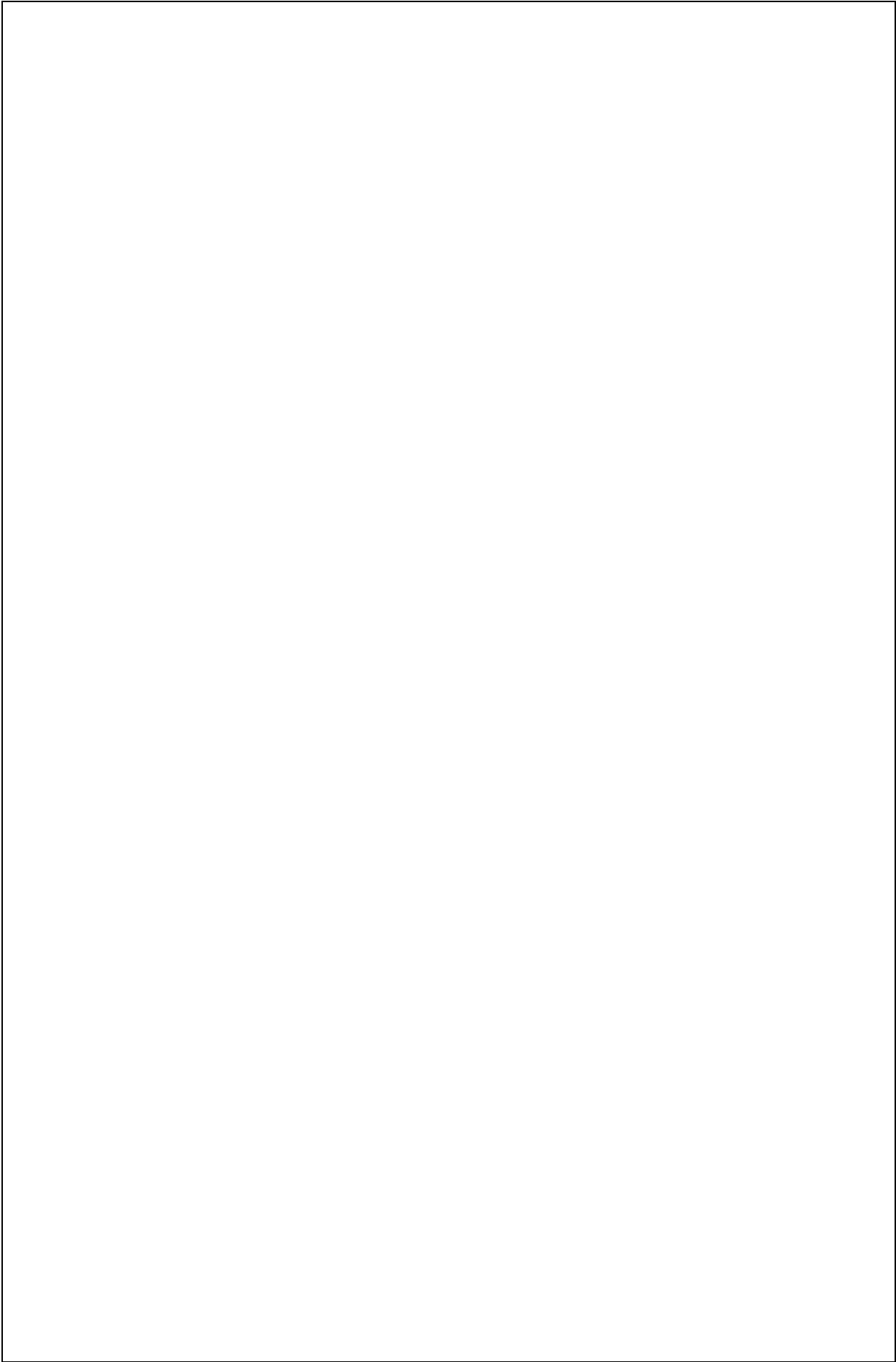
$$w(x) = \frac{10000}{24}x^4 - \frac{5000}{48}x^3 + \frac{100}{16}x^2, \quad x \text{ und } w(x) \text{ werden in Metern angegeben.}$$

Erstellen Sie ein C-Programm, welches die folgenden beiden Aufgaben löst:

- 2.1. In einer Schleife wird mit der oben angegebenen Formel die Durchbiegung $w(x)$ für alle Positionen x von $0 \dots 0,1$ m in Schritten von $\Delta x = 0,001$ m berechnet und im folgenden Format auf dem Bildschirm ausgegeben (Anzahl der Nachkommastellen beachten!):



- 2.2. Das Programm ermittelt während des Schleifendurchlaufs die Position x_{max} , an der die Durchbiegung maximal wird und gibt diese Position und die Durchbiegung $w(x_{max})$ auf dem Bildschirm aus (Anzahl der Nachkommastellen wie im Unterpunkt 2.1).



Aufgabe 3: (ca. 22 Punkte)

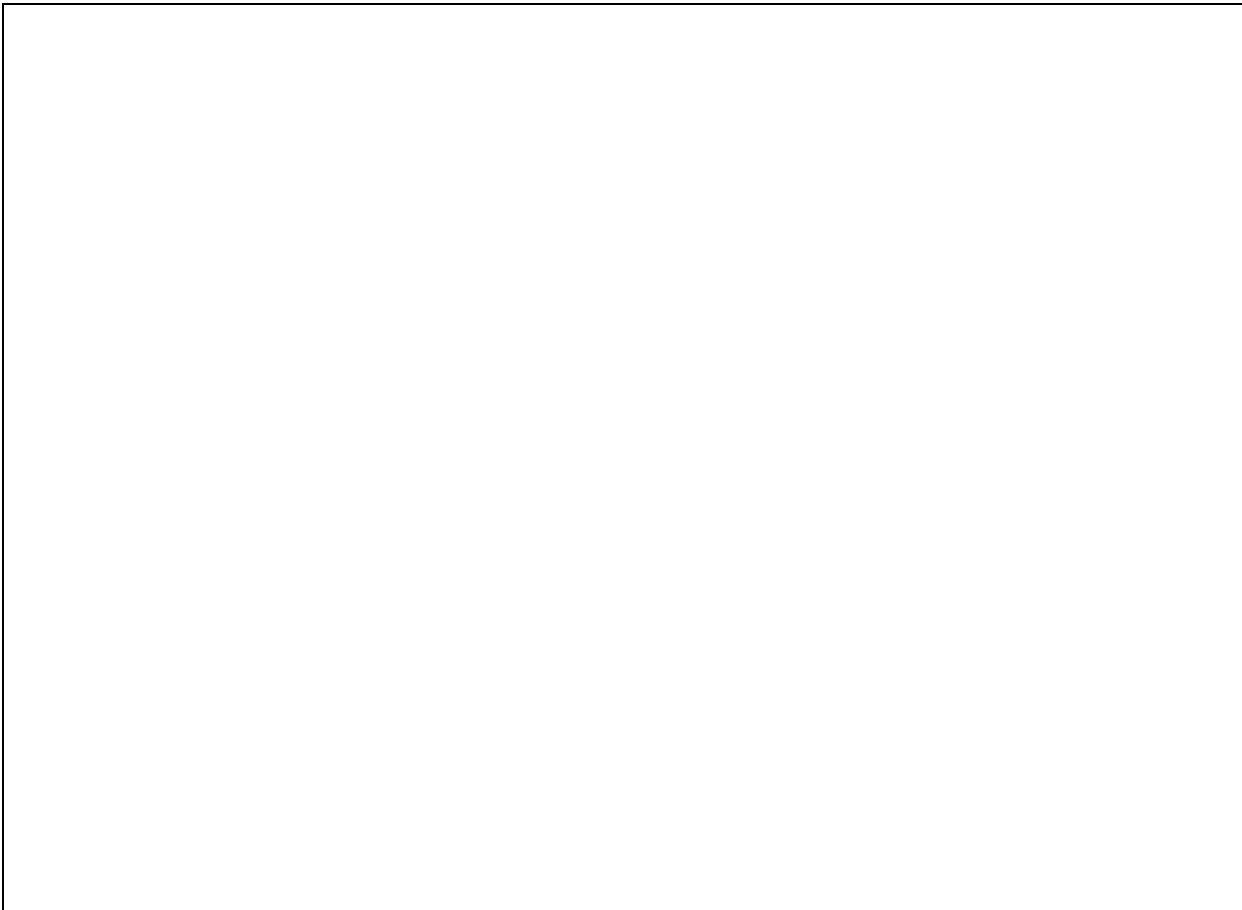
Das folgende Programm läuft auf einem KFZ-Steuergerät; es dient zum Auslösen eines Airbags.

Die Funktionen Geschw(), und Beschl1() ermitteln die aktuelle Geschwindigkeit und die Beschleunigung des Fahrzeugs. **Die Definitionen dieser Funktionen sind nicht dargestellt; sie können als gegeben vorausgesetzt werden! Entsprechendes gilt für die Funktion AirbagZuenden().**

```
double Geschw(void);      /* Rückgabe in km/h      */
double Beschl1(void);     /* Rückgabe in m/s2    */
double Beschl2(void);     /* Rückgabe in m/s2    */
double Beschl3(void);     /* Rückgabe in m/s2    */
void AirbagZuenden(void); /* Airbag wird ausgelöst */

int main(void)
{
    double v, a;
    while(1 == 1)
    {
        v = Geschw();
        a = Beschl1();
        if(v >= 30.0)
        {
            if(a <= -40.0) /* Bremsbeschleunigung: negative Werte */
                AirbagZuenden();
        }
    }
    return 0;
}
```

3.1. Zeichnen Sie ein Struktogramm der Funktion main().



3.2. Wie oft wird die while-Schleife im abgebildeten Programm durchlaufen?

3.3. Welche Bedingungen müssen erfüllt sein, damit der Airbag ausgelöst wird?

3.4. Bei Praxistests zeigt sich, dass der Airbag gelegentlich zündet, ohne dass das KFZ in einen Unfall verwickelt ist. Um solche Fehlzündungen zu vermeiden, wird das Programm erweitert:

Das Programm fragt nun die Beschleunigungswerte von **drei unterschiedlichen Sensoren** ab. Dazu dienen die Funktionen `Beschl1()`, `Beschl2()`, `Beschl3()`. Der Airbag wird nur dann ausgelöst, wenn das KFZ mit einer Geschwindigkeit von mindestens 30 km/h fährt und zugleich **mindestens zwei der drei Sensoren eine Bremsbeschleunigung von mindestens 40 m/s² melden**.

Wie lautet die neue, erweiterte Funktion `main()`?

```
int main(void)
{
```

```
    return 0;
}
```

Aufgabe 4: (ca. 22 Punkte)**Name:** _____

- 4.1. Definieren Sie eine Funktion `double laenge(double *v)`; zur Berechnung der Länge eines dreidimensionalen Vektors. Der Vektor, der an diese Funktion übergeben wird, besteht aus drei `double`-Elementen. (Die Länge des Vektors erhalten Sie, wenn Sie die drei Elemente des Vektors quadrieren, addieren und von dem Ergebnis die Quadratwurzel berechnen.)

- 4.2. Die Funktion aus Unterpunkt 4.1 soll im folgenden Hauptprogramm für den Vektor `myVec` aufgerufen und die berechnete Länge auf dem Bildschirm dargestellt werden. Ergänzen Sie die abgebildete Funktion `main()`.

```
int main(void)
{
    double myVec[] = { 3, 4, 0 };
    printf("Laenge von myVec: %f\n",  );
    return 0;
}
```

- 4.3. Das folgende Programm berechnet die „Spur“ einer Matrix und gibt das Ergebnis auf dem Bildschirm aus. (Die Spur einer Matrix ist die Summe aller Elemente auf der Hauptdiagonalen.)

Korrigieren Sie die fünf Fehler, die sich in den C-Quelltext eingeschlichen haben.

```
#include <studio.h>

int main(void)
{
    int i;

    double spur = 0, mat[3][3] =
        { { 1, 2, 3 },
          { 4, 5, 6 },
          { 7, 8, 9 } };

    for(i = 0, i < 3; ++i);
        spur += mat[i];

    printf("Spur: %.2d\n", spur);
    return 0;
}
```

- 4.4. Das folgende Programm sortiert alle Zeichen einer Zeichenkette in alphabetischer Reihenfolge. Anschließend wird die Zeichenkette auf dem Bildschirm ausgegeben.

Ergänzen Sie die fehlenden Anweisungen bzw. Befehle!

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

#define MAX_LENGTH 100
char str[MAX_LENGTH];

/* Deklaration (Funktionsprototyp) der Sortierfunktion */
[ ]

/* Hauptprogramm */
int main(void)
{
    printf("Zeichenkette eingeben: ");

    /* Eingabe der Zeichenkette - maximale Länge beachten! */
    [ ]

    /* Aufruf der Sortierfunktion */
    [ ]

    /* Ausgabe der sortierten Zeichenkette */
    [ ]

    return 0;
}

/* Zeichen in der Zeichenkette mittels Bubble-Sort sortieren */
void sortiere(void)
{
    int i, n = strlen(str);
    char hilf, sortiert;

    /* Schleife wiederholen, bis Zeichenkette sortiert ist */
    do
    {
        /* Annahme: Zeichenkette ist sortiert */
        sortiert = 'j';
        for(i = 1; i < n; i++)
        {
            if(str[i] < str[i-1])
            {
                /* str[i], str[i-1] tauschen */
                hilf = str[i];
                str[i] = str[i-1];
                str[i-1] = hilf;

                /* ...war doch noch nicht sortiert! */
                sortiert = 'n';
            }
        }
    }
    while( [ ] );
}
```