

Ingenieurinformatik

Name	Vorname	Matrikelnummer	Sem.-Gr.:	Hörsaal	Platz

Zulassung geprüft
vom Aufgabensteller:

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

Aufgabensteller: Dr. Jäger-Hezel, Dr. Küpper, Dr. Reichl, Dr. Selting

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. 22 Punkte)

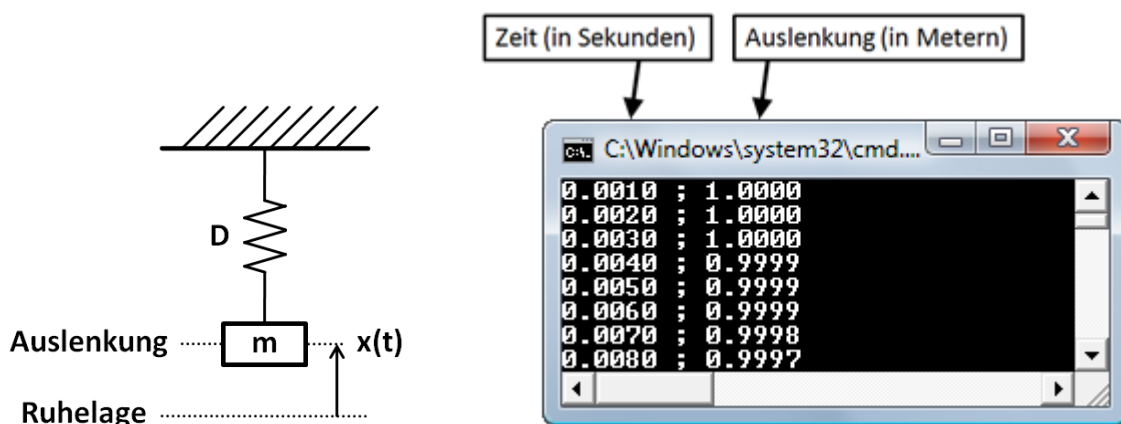
Ein beweglicher Körper der Masse m ist an einer Feder mit der Federkonstante D befestigt. Die Auslenkung des Körpers aus der Ruhelage ist $x(t)$, seine Geschwindigkeit beträgt $v(t)$. Falls Auslenkung $x(t)$ und Geschwindigkeit $v(t)$ zum Zeitpunkt t bekannt sind, können $x(t + \Delta t)$ und $v(t + \Delta t)$ zum Zeitpunkt $(t + \Delta t)$ berechnet werden:

$$x(t + \Delta t) \approx x(t) + \Delta t \cdot v(t)$$

$$v(t + \Delta t) \approx v(t) - \Delta t \cdot \frac{D}{m} \cdot x(t)$$

Erstellen Sie ein C-Programm zur Berechnung des Bewegungsvorgangs:

- Die Berechnung beginnt zum Zeitpunkt $t = 0$.
- Eine Schleife durchläuft ANZAHL Zeitschritte der Länge DELTA_T.
- Federkonstante D , Masse m , Start-Auslenkung $x(t = 0)$ und Start-Geschwindigkeit $v(t = 0)$ sind als Konstanten definiert.
- Bei jedem Schleifendurchlauf sind jeweils die aktuelle Zeit t und die Auslenkung $x(t)$ **mit vier Nachkommastellen** auf dem Bildschirm auszugeben.



```
#define DELTA_T 0.001      /* Zeitschritt fuer Naehungsverfahren */
#define ANZAHL 10000      /* Anzahl der Simulationsschritte */
#define D 5.0              /* Federkonstante in N/m */
#define M 0.5              /* Masse in kg */
#define X_START 1.0        /* Auslenkung in m bei t = 0 */
#define V_START 0.0        /* Geschwindigkeit in m/s bei t = 0 */
```

Aufgabe 3: (ca. 18 Punkte)

Das folgende C-Programm berechnet zu einer vom Anwender eingegebenen Höhe den Luftdruck (in Pascal) und die Temperatur (in Grad Celsius). Ein- und Ausgabe finden im Hauptprogramm (Funktion „main“), Berechnung von Luftdruck und Temperatur finden in der Funktion „DruckTemp“ statt.

```
#include <stdio.h>

double druck, temp;
int DruckTemp(double hoehe);

int main(void)
{
    int nochmal;
    double h;

    do
    {
        printf("Hoehe in Metern? (0...20000) ");
        scanf("%lf", &h);

        if(DruckTemp(h) == 1)
        {
            printf("Druck: %f Pascal, ", druck);
            printf("Temperatur: %f Grad Celsius\n", temp);
        }
        else
        {
            printf("Hoehe ist ungueltig!\n");
        }

        printf("Nochmal berechnen? (0 = Nein, 1 = Ja) ");
        scanf("%d", &nochmal);
    }
    while(nochmal == 1);
    return 0;
}

int DruckTemp(double hoehe);
{
    int rueckgabe, int auswahl;
    auswahl = 1;

    if(hoehe > 11000) auswahl = 2;
    if(hoehe > 20000 || hoehe < 0) auswahl = 0;

    switch(auswahl);
    {
    case 1:    druck = 101325.0 - 7.153909 * hoehe;
              temp  = 15.0 - 0.0065 * hoehe;
              rueckgabe = 1;

    case 2:    druck = 43601.788889 - 1.906344 * hoehe;
              temp  = -56.5;
              rueckgabe = 1;

    default:  rueckgabe = 0;
              break;
    }

    return rueckgabe;
}
```

- 3.1. Korrigieren Sie die fünf Fehler in der Definition der Funktion „DruckTemp“.
- 3.2. Zeichnen Sie ein Struktogramm der (korrigierten) Funktion „DruckTemp“.



- 3.3. Wodurch unterscheiden sich „abweisende“ und „nicht abweisende“ Schleifen? Welcher dieser beiden Schleifentypen wird in der Funktion „main“ eingesetzt?



Aufgabe 4: (ca. 27 Punkte)

- 4.1. In einem C-Programm ist die Matrix `a` definiert und bereits mit Werten belegt. Schreiben Sie eine Funktion „`mat_ident`“. Diese Funktion soll überprüfen, ob die Matrix `a` die Einheitsmatrix ist (Rückgabewert `== 1`) oder nicht (Rückgabewert `== 0`). Hinweis: In der Einheitsmatrix sind die Elemente auf der Hauptdiagonale gleich 1 und alle anderen Elemente gleich 0.

```
#define DIM 10
int a[DIM][DIM];

int mat_ident(void)
{

}

}
```

- 4.2. Das folgende C-Programm entschlüsselt einen geheimen Text und gibt ihn auf dem Bildschirm aus. Wie lautet die Ausgabe des Programms?

```
#include <stdio.h>
int Decrypt(char *text);
int main(void)
{
    char geheim[] = "Qdhhgk";
    int x = Decrypt(geheim);
    printf("\n%d", x);
    return 0;
}

int Decrypt(char *text)
{
    int count = 0;
    char ch;
    while(*text != 0)
    {
        ch = *text;
        printf("%c", ch + 1);
        ++text; ++count;
    }
    return count;
}
```

Ausgabe:

- 4.3. Das folgende C-Programm füllt einen globalen int-Vektor mit einer Zahlenreihe und gibt anschließend den Inhalt des Vektors auf dem Bildschirm aus. Das Programm soll so verändert werden, dass der Vektor nicht mehr als globale Variable, sondern als lokale Variable in der Funktion „main“ definiert ist. Die einzelnen Werte der Zahlenreihe sollen durch diese Änderung nicht beeinflusst werden. Wie lauten Deklaration (Prototyp), Definition und Funktionsaufruf der neuen Funktion „fill“?

Alte Version:

```
#include <stdio.h>

#define SIZE 10
int reihe[SIZE];

void fill(void);

int main(void)
{
    int i;
    fill();
    for(i = 0; i < SIZE; ++i)
        printf("%d ", reihe[i]);
    return 0;
}

void fill(void)
{
    int i;
    reihe[0] = 0;
    reihe[1] = 1;
    for(i = 2; i < SIZE; ++i)
        reihe[i] = reihe[i-1]
                + reihe[i-2];
}
```

Neue Version:

```
#include <stdio.h>

#define SIZE 10

/* Deklaration von "fill": */

int main(void)
{
    int i, reihe[SIZE];

    /* Aufruf von "fill": */

    for(i = 0; i < SIZE; ++i)
        printf("%d ", reihe[i]);
    return 0;
}

/* Definition von "fill": */
```