

Ingenieurinformatik

Name	Vorname	Matrikelnummer	Sem.-Gr.:	Hörsaal	Platz

Zulassung geprüft
vom Aufgabensteller:

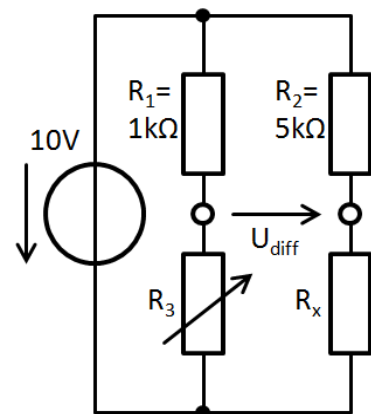
Teil 1	Aufg. 2	Aufg. 3	Aufg. 4	Summe	Note

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)

Mit der dargestellten Schaltung kann der Wert des unbekannten Widerstands R_x bestimmt werden. Dazu wird der regelbare Widerstand R_3 so eingestellt, dass die Differenzspannung U_{diff} verschwindet ($U_{diff} = 0V$). In diesem Fall gilt: $R_3 / (R_1 + R_3) = R_x / (R_2 + R_x)$, damit kann R_x ausgerechnet werden. In einem C-Programm soll die Reaktion der Schaltung auf eine Veränderung des regelbaren Widerstands R_3 simuliert werden.

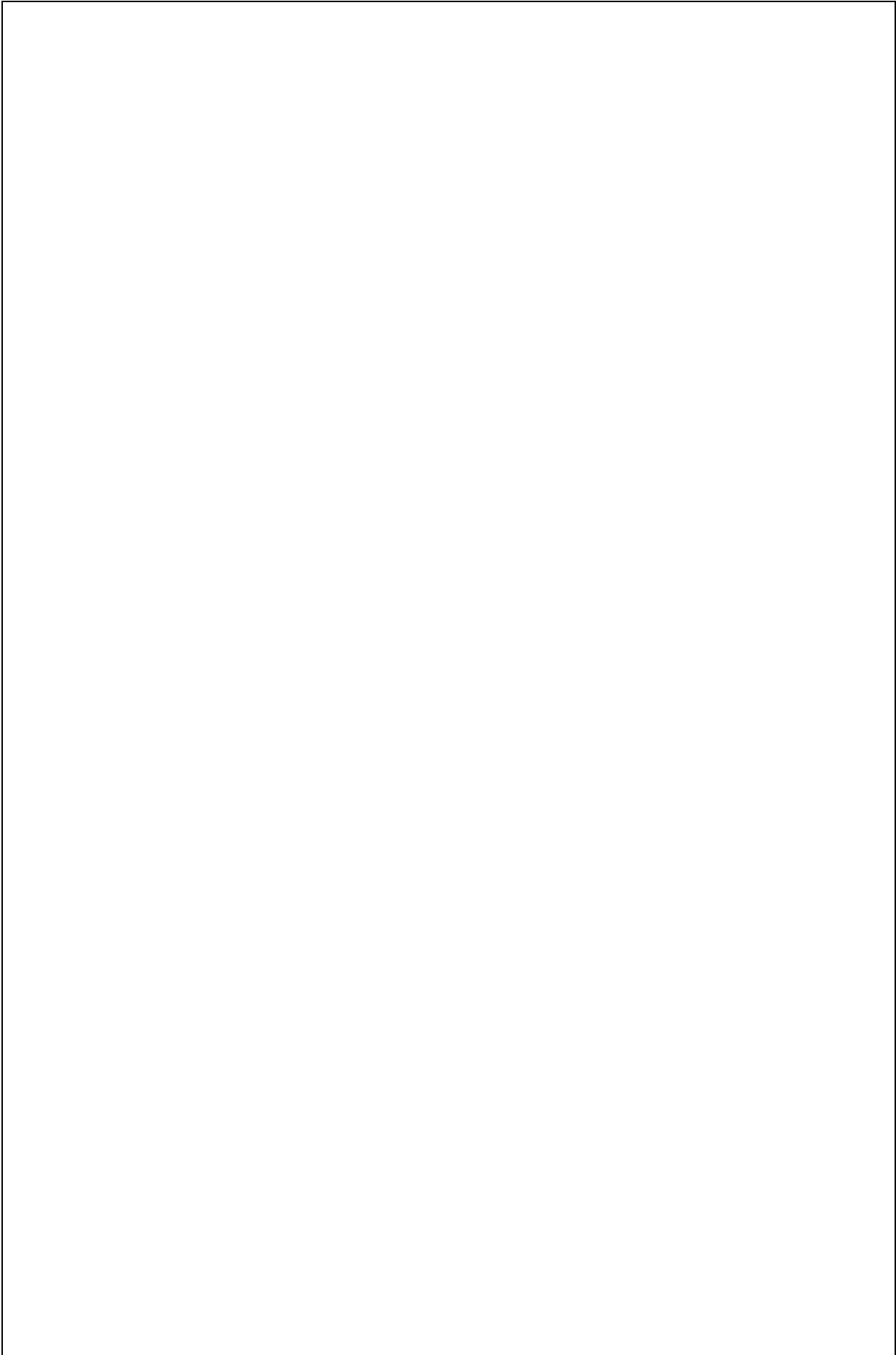


Schreiben Sie ein Programm, welches vom Anwender zunächst die Eingabe des „unbekannten“ Widerstandswerts R_x verlangt. Anschließend soll die Differenzspannung U_{diff} für alle Werte R_3 von 0...5kΩ in Schritten von 1Ω berechnet und auf dem Bildschirm mit zwei Nachkommastellen ausgegeben werden. U_{diff} kann wie folgt berechnet werden: $U_{diff} = 10V \cdot R_3 / (R_1 + R_3) - 10V \cdot R_x / (R_2 + R_x)$.

Nach dem Ende der Berechnungen soll auf dem Bildschirm derjenige Wert von R_3 ausgegeben werden, bei dem die betragsmäßig kleinste Differenzspannung U_{diff} bestimmt wurde (Achtung: U_{diff} nimmt sowohl positive als auch negative Werte an!).

```

C:\Windows\system32\cmd.exe
Widerstand Rx eingeben: 3200
R3 = 0 Ohm, Udiff = -3.90 Volt
R3 = 1 Ohm, Udiff = -3.89 Volt
R3 = 2 Ohm, Udiff = -3.88 Volt
R3 = 3 Ohm, Udiff = -3.87 Volt
R3 = 4 Ohm, Udiff = -3.86 Volt
R3 = 4996 Ohm, Udiff = 4.43 Volt
R3 = 4997 Ohm, Udiff = 4.43 Volt
R3 = 4998 Ohm, Udiff = 4.43 Volt
R3 = 4999 Ohm, Udiff = 4.43 Volt
R3 = 5000 Ohm, Udiff = 4.43 Volt
Udiff ist minimal bei R3 = 640 Ohm
Drücken Sie eine beliebige Taste . . .
    
```



Aufgabe 3: (ca. 18 Punkte)

Das abgebildete Programm berechnet den Preis, der für einen Mietwagen zu bezahlen ist. Der Anwender muss dazu die Mietdauer (in Tagen) sowie die gefahrenen Kilometer eingeben.

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

#define INKL_KM 60.0
#define KM_PREIS 0.40
#define TAG_PREIS 60.0

int main(void)
{
    char weiter;
    double km, tage, preis, zus_km;

    do
    {
        printf("Tage: "      ); scanf("%lf", &tage);
        printf("Kilometer: "); scanf("%lf", &km );

        zus_km = km - tage * INKL_KM;
        if(zus_km < 0) zus_km = 0;

        preis = tage * TAG_PREIS + zus_km * KM_PREIS;
        printf("Preis: %.2f EUR\n", preis);

        do
        {
            printf("Nochmal (j/n) bzw. (J/N): ");
            scanf(" %c", &weiter);
            weiter = tolower(weiter); /*1*/
        }
        while(weiter != 'j' && weiter != 'n'); /*2*/
    }
    while(weiter == 'j'); /*3*/
    return 0;
}
```

- 3.1. Zeichnen Sie auf der folgenden Seite 4 ein Struktogramm des abgebildeten C-Quelltextes. Alle Befehle und insbesondere die verwendeten Kontrollstrukturen (Verzweigungen, Schleifen) sollen im Struktogramm erkennbar sein. Verwenden Sie im Struktogramm keine C-Befehle!
- 3.2. Gegen Ende des C-Programms sind drei Zeilen mit /*1*/, /*2*/ und /*3*/ markiert. Die Zeile /*1*/ wird gelöscht: Wie müssen die **while**-Anweisungen in den Zeilen /*2*/ und /*3*/ geändert werden, damit das Verhalten des Programms gleich bleibt?

Zeile /*2*/:

Zeile /*3*/:

Struktogramm:

Aufgabe 4: (ca. 25 Punkte)**Name:** _____

- 4.1. Das abgebildete Programm berechnet die Geschwindigkeit und die zurückgelegte Strecke, die ein Körper im freien Fall (Fallbeschleunigung = $9,81 \text{ m/s}^2$) zum Zeitpunkt t erreicht hat. Die Berechnungen werden von der Funktion **freier_fall** durchgeführt, die ihre Ergebnisse mittels globaler Variablen an das Hauptprogramm übermittelt.

Ändern Sie das Programm, so dass keine globalen Variablen mehr erforderlich sind (die Berechnungen sollen weiterhin von einer einzelnen Funktion **freier_fall** durchgeführt werden; die Ausgabe der beiden Ergebnisse soll weiterhin in der Funktion **main** erfolgen)!

```
#include <stdio.h>

double s, v;
int freier_fall(double t);

int main(void)
{
    double t;

    printf("Sekunden: ");
    scanf("%lf", &t);

    if(freier_fall(t) == 0)
    {
        printf("Eingabefehler!\n");
        return 0;
    }

    printf("v = %.2fm/s\n", v);
    printf("s = %.2fm\n", s);
    return 0;
}

int freier_fall(double t)
{
    if(t < 0) return 0;
    v = 9.81 * t;
    s = 0.5 * 9.81 * t * t;
    return 1;
}
```

```
#include <stdio.h>

int main(void)
{
    double t;

    printf("Sekunden: ");
    scanf("%lf", &t);
```

4.2. Das folgende C-Programm multipliziert eine Matrix mit einem Vektor.

```
#define DIM = 3
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i; row, col;
    double z1[DIM], z2[DIM];
    double z3[DIM][DIM] = {{2,0,0},{0,2,0},{0,0,2}};
    for(i = 0; i < DIM; ++i)
        z1[i] = 20.0 * rand() / RAND_MAX - 10.0;
    for(row = 0; row < DIM; ++row)
    {
        for(z2[row] = 0, col = 0; col < DIM; ++col)
            z2[row] += z3[row][col] * z1[col]; /**/
    }
    for(i = 0; i <= DIM; ++i)
        printf("%8.2f\t%8.2d\n", z1[i], z2[i]);
    return 0;
}
```

4.2.1. Korrigieren Sie die fünf Fehler, die sich im C-Quelltext befinden!

4.2.2. In welcher Variable wird das Ergebnis der Multiplikation abgespeichert?

4.2.3. Wie oft wird die mit `/**/` markierte Zeile ausgeführt?

4.2.4. **z1** wird mit Zufallszahlen gefüllt. In welchem Zahlenbereich liegen diese Zufallszahlen?

4.3. Beantworten Sie die folgenden Fragen zur Programmierung mit Zeichen bzw. Zeichenketten.

4.3.1. Wie groß ist eine Variable des Typs **char**?

7 Bit

1 Byte

16 Bit

4.3.2. Was kann in einer Variablen des Typs **char** gespeichert werden? Ein Zeichen aus dem...

ASCII-Zeichensatz

erweiterten ASCII-Zeichensatz

Unicode-Zeichensatz

4.3.3. Definieren Sie einen Vektor zum Speichern einer Zeichenkette mit max. 100 Zeichen.

4.3.4. Wie lautet der **printf**-Befehl zur Ausgabe dieses Vektors auf dem Bildschirm?

4.3.5. Wie ermittelt dieser **printf**-Befehl die Anzahl der auszugebenden Zeichen?