

Ingenieurinformatik

C-Programmierung

Name	Vorname	Semestergruppe	Hörsaal

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Summe

Beginn Ihres Bachelorstudiums:

- vor WS13/14 (Kombiprüfung C + MATLAB) **
- von WS13/14 bis WS15/16 **
- von SS16 bis WS17/18 (Kombiprüfung C + MATLAB)
- ab SS18

**** Die Prüfung ist nur dann gültig, wenn Sie die Zulassungsvoraussetzung erworben haben (erfolgreiche Teilnahme am Praktikum).**

Aufgabensteller: Dr. Reichl, Dr. Küpper und Kollegen

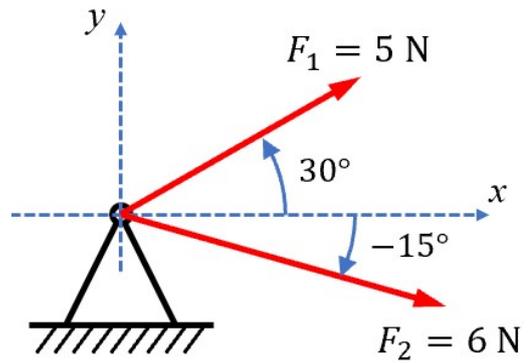
Bearbeitungszeit: 60 Minuten

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

***** Viel Erfolg! *****

Aufgabe 1: (ca. 23 Punkte)

Auf ein Festlager wirken bis zu 50 einzelne Kräfte F_1, F_2, F_3 usw. wie im Bild rechts dargestellt. (Im Bild sind nur zwei Kräfte F_1 und F_2 abgebildet.)



Nach dem Programmstart wird von allen Kräften der Betrag und der Winkel eingegeben. Sie dürfen davon ausgehen, dass alle Winkel im Bereich $-90^\circ < \alpha < +90^\circ$ liegen. Die Eingabe wird beendet, wenn als Kraft ein negativer Wert oder null eingegeben wird. Die Eingabe wird ebenfalls beendet, wenn die maximale Anzahl von 50 Kräften eingegeben wurde.

Nun schreibt das Programm eine Tabelle mit allen eingegebenen Kräften und Winkeln auf den Bildschirm. Die Kräfte und Winkel werden mit einer Feldbreite von 8 Zeichen und 3 Nachkommastellen ausgegeben.

Schließlich wird die Gesamtkraft F_{ges} ausgerechnet: Der Betrag und der Winkel von F_{ges} werden mit 3 Nachkommastellen ausgegeben. Wenn keine gültigen Kräfte eingegeben wurden, dann lautet die Ausgabe: „Es wurden keine Kräfte eingegeben.“

Die Ausgabe Ihres Programms soll aussehen, wie es in den Bildschirmfotos gezeigt ist!

C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe

```

Kraft / N: 5
Winkel / Grad: 30
Kraft / N: 6
Winkel / Grad: -15
Kraft / N: 0
    
```

Eingabe der Kräfte und Winkel durch den Anwender. Die Eingabe wird beendet, wenn der Anwender eine negative Kraft oder null eingibt (oder wenn die max. Anzahl von 50 Kräften eingegeben wurde).

```

1: 5.000 N, 30.000 Grad
2: 6.000 N, -15.000 Grad
    
```

Eine Tabelle mit allen Kräften und Winkeln wird vom Programm ausgegeben.

```

Gesamtkraft: 10.170 N
Winkel: 5.344 Grad
    
```

Betrag und Winkel der Gesamtkraft werden berechnet und ausgegeben.

C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe

```

Kraft / N: 0
Es wurden keine Kräfte eingegeben.
    
```

In diesem Beispiel wurde keine gültige Kraft eingegeben!

Aufgabe 2: (ca. 22 Punkte)

In einem C-Programm ist eine globale Matrix wie folgt definiert:

```
double m[10][10];
```

- 2.1. Schreiben Sie eine Funktion **void fill_mat(void)**, welche alle Elemente dieser Matrix mit unterschiedlichen Zufallszahlen im Bereich von -0,5 ... +0,5 füllt.

```
#include <stdlib.h>  
void fill_mat(void)  
{
```

```
}
```

- 2.2. Welchen Speicherplatz benötigt die Matrix **m** im Hauptspeicher des Rechners? (Gehen Sie davon aus, dass es sich um einen aktuellen Windows-10-Rechner mit Qt Creator handelt.)

Speicherbedarf von **double m[10][10];**

Bytes

- 2.3. Wie ist der Speicherbedarf, wenn der Datentyp von **double** auf **float** geändert wird?

Speicherbedarf von **float m[10][10];**

Bytes

- 2.4. Wie ist der Speicherbedarf, wenn eine dritte Dimension hinzugefügt und der Datentyp auf **short** geändert wird?

Speicherbedarf von **short m[10][10][10];**

Bytes

- 2.5. Nun soll eine weitere Funktion **void auswertung(void)** erstellt werden. Diese Funktion ermittelt den Wert des Elements mit dem größten Betrag (Tipp: ...auch negative Werte sind möglich!) und zusätzlich auch den arithmetischen Mittelwert aller Elemente. Beide Ergebnisse (der Wert des Elements mit dem größten Betrag und der arithmetische Mittelwert) werden ausgegeben.

Zeichnen Sie ein Struktogramm der Funktion **void auswertung(void)**. Der Quelltext der Funktion **soll nicht (!)** aufgeschrieben werden.

Lösen Sie diese Aufgabe mit while-Schleifen, also nicht mit do-while- oder for-Schleifen!

Aufgabe 3: (ca. 5 Punkte)

Das folgende C-Programm liest zwei double-Werte a und b von der Tastatur ein. Dabei soll a positiv sein, b soll größer als a sein. Es wird die Abweichung der beiden Werte in Prozent berechnet und auf dem Bildschirm ausgegeben. Korrigieren Sie die fünf Fehler, die sich im Quelltext befinden!

```
#include <stdio.h>
void eingabe2(double *a, double *b)

int main(void)
{
    double a, b;
    eingabe2(&a, &b);
    if(a <= 0)
        printf("Fehler: a ist nicht positiv!\n");
    else if(b <= a)
        printf("Fehler: b ist nicht groesser als a!\n");
    else
    {
        double proz = b - a;
        printf("%.1f ist %.1f % groesser als %.1f.\n", b, proz, a);
    }
    return 0;
}

void eingabe2(double *a, double *b)
{
    double tmp1, tmp2;
    printf("Bitte eine positive Zahl eingeben!\n");
    printf("a: "); scanf("%lf", &tmp1);
    printf("Bitte eine groessere Zahl eingeben!\n");
    printf("b: "); scanf("%lf", &tmp2);
    *a = tmp1; b = tmp2;
    return 0;
}
```

Aufgabe 4: (ca. 5 Punkte)

Zu Testzwecken soll eine Zeichenkette erzeugt werden, die aus insgesamt 25 Buchstaben "a", "b" und "c" in zufälliger Reihenfolge (!) besteht. Vervollständigen Sie den folgenden C-Quelltext.

```
#include <stdio.h>
#include <stdlib.h>
#define LENGTH 25

int main(void)
{
    char str[LENGTH + 1]; str[LENGTH] = 0;
    for(int i = 0; i < LENGTH; ++i)
    {
        int zuf = 

        switch(zuf)
        {
            case 1: str[i] = ; break;
            case 2: str[i] = ; break;
            case 3: str[i] = ; break;
        }
    }
    printf("Zufaellige Buchstabenfolge: %s\n", str);
    return 0;
}
```

Aufgabe 5: (ca. 12 Punkte)

- 5.1. Wandeln Sie die folgende Hexadezimalzahl in eine Binärzahl um: **FF81** _{HEX} = ??? _{BIN}
- 5.2. Nehmen Sie an, dass es sich bei dem Ergebnis aus 5.1 um eine 16-Bit-Zweierkomplementzahl handelt. Woran erkennen Sie, welches Vorzeichen die 16-Bit-Zweierkomplementzahl hat?
(Kurze Erläuterung in Stichworten erforderlich!)
- 5.3. Wandeln Sie die 16-Bit-Zweierkomplementzahl aus 5.2 in eine positive 16-Bit-Zweierkomplementzahl mit demselben Betrag um. (Die einzelnen Umwandschritte aufschreiben!)
- 5.4. Geben Sie den Wert der (positiven) 16-Bit-Zweierkomplementzahl aus 5.3 als Dezimalzahl an.
- 5.5. Gehen Sie davon aus, dass Sie an einem aktuellen Windows-10-Rechner mit Qt Creator arbeiten: Wie viele unterschiedliche int-Zahlen können Sie auf diesem Rechner darstellen?
(Die Angabe einer 2er-Potenz als Antwort ist erlaubt.)
- 5.6. Ist die folgende Aussage wahr oder falsch: „Jede int-Zahl kann ohne Datenverlust in eine float-Zahl konvertiert werden.“
(Kurze Begründung in Stichworten erforderlich!)

(Platz für Notizen und Nebenrechnungen)