

Ingenieurinformatik

Teil 1, Programmierung in Python

Name	Vorname	Semestergruppe	Hörsaal

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Summe

Aufgabensteller: Reichl, Küpper und KollegInnen

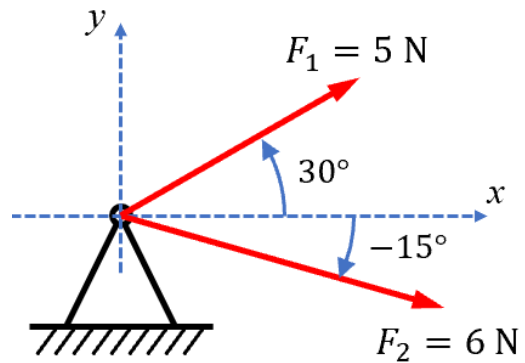
Bearbeitungszeit: 60 Minuten

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

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

Aufgabe 1: (ca. 25 Punkte)

Auf ein Festlager wirken beliebig viele 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.)



Schreiben Sie ein Python-Skript, welches die folgenden Aufgaben löst:

- Nach dem Start 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 eingegebenen Kräfte und Winkel werden in Python-Listen gespeichert.
- Nach dem Ende der Eingabe wird eine Tabelle mit allen Kräften und Winkeln ausgegeben. Die Kräfte und Winkel werden mit einer Feldbreite von 8 Zeichen und 3 Nachkommastellen ausgegeben. Zu Beginn jeder Zeile wird eine laufende Nummer ausgegeben, zum Beispiel „1:“.
- Schließlich wird die Gesamtkraft F_{ges} ausgerechnet: Der Betrag und der Winkel von F_{ges} werden mit 3 Nachkommastellen ausgegeben. Wenn keine Kräfte eingegeben wurden, dann lautet die Ausgabe: „Es wurden keine Kräfte eingegeben.“

Hinweise:

- Sie dürfen davon ausgehen, dass nur gültige Zahlenwerte eingegeben werden; Fehler bei der Eingabe müssen von Ihrem Skript also nicht erkannt werden.
- Die Ausgabe Ihres Skripts soll so aussehen, wie es in den Bildschirmfotos gezeigt ist!

```
Konsole 1/A x
Kraft / N: 5
Winkel / Grad: 30
Kraft / N: 6
Winkel / Grad: -15
Kraft / N: 0

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

Gesamtkraft: 10.170 N
Winkel: 5.344 Grad
```

Eingabe der Kräfte und Winkel durch den Anwender. Die Eingabe wird beendet, wenn der Anwender eine negative Kraft oder null eingibt.

Eine Tabelle mit allen Kräften und Winkeln wird vom Python-Skript ausgegeben (laufende Nummer zu Beginn der Zeile).

Betrag und Winkel der Gesamtkraft werden berechnet und ausgegeben.

```
Konsole 2/A x
Kraft / N: -1
Es wurden keine Kräfte eingegeben.
```

In diesem Beispiel wurde keine gültige Kraft eingegeben.

Aufgabe 2: (ca. 18 Punkte)

2.1. Das folgende Python-Skript läuft im Steuergerät eines Kühlschranks.

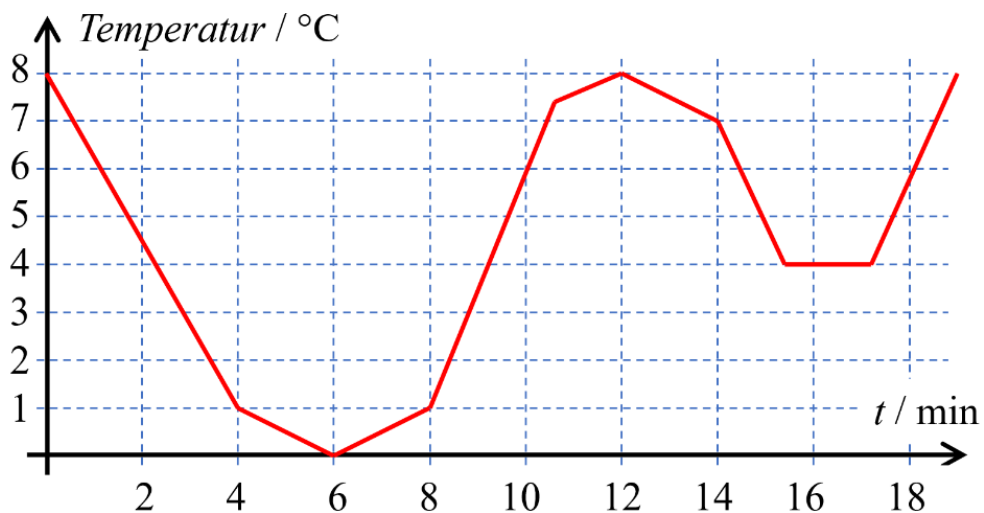
- Die Funktion `get_temp()` liefert die aktuelle Temperatur (in °C) im Kühlschrank.
- Die Funktion `set_compr()` dient zum Ein- bzw. Ausschalten des Kühlschrank-Kompressors.
- **Die genauen Definitionen dieser beiden Funktionen sind nicht abgedruckt, sie können im Modul `kuehlschrank.py` als gegeben vorausgesetzt werden..**

```
from kuehlschrank import get_temp, set_compr
```

```
temp1 = 6.0  
temp2 = 2.0  
state = 0
```

```
while True:  
    temp = get_temp()  
  
    if temp > temp1:  
        if state == 0:  
            set_compr(1) # einschalten  
            state = 1  
  
    if temp < temp2:  
        if state == 1:  
            set_compr(0) # ausschalten  
            state = 0
```

Das Diagramm zeigt den zeitlichen Verlauf der Temperatur im Kühlschrank. **Markieren Sie im Diagramm die Zeitintervalle**, in denen der Kühlschrank-Kompressor eingeschaltet ist.



- 2.2. Das Python-Skript aus Unterpunkt 2.1 soll schon einmal getestet werden, obwohl der Kühlschrank noch gar nicht aufgebaut worden ist. Zu diesem Zweck wird die Funktion `get_temp()` durch die folgende Variante ersetzt: Dadurch werden simulierte (zufällige) Temperaturwerte erzeugt.

```
import random as rnd

def get_temp():
    t = rnd.randrange(15, 65) / 10
    return t
```

Welches ist der minimale bzw. maximale (simulierte) Temperaturwert, der von der abgebildeten Funktion `get_temp()` als Rückgabewert zurückgegeben werden kann?

Minimaler Temperaturwert:

 °C

Maximaler Temperaturwert:

 °C

- 2.3. Zeichnen Sie ein Struktogramm, welches den genauen Ablauf des Python-Skripts aus dem Unterpunkt 2.1 zeigt. Alle Schleifen und Verzweigungen sollen im Struktogramm sichtbar sein.

Aufgabe 3: (ca. 7 Punkte)

Die Funktion `find_first()` ermittelt die Position (0, 1, 2 usw.), an der das Zeichen „ch“ zum ersten Mal in der Zeichenkette „s“ vorkommt. Falls das Zeichen „ch“ gar nicht in der Zeichenkette „s“ vorkommt, gibt `find_first()` den Rückgabewert -1 zurück. Vervollständigen Sie den Quelltext von `find_first()`.

Hinweis: Die Python-String-Methode `find()` darf in dieser Aufgabe nicht verwendet werden!

```
def find_first( ):
    anz = 
    pos = -1
    for i in (0, anz):
        if :
            break
    return
```

```
s = input("Zeichenkette eingeben: ")
ch = 'x'
pos = find_first(s, ch)
```

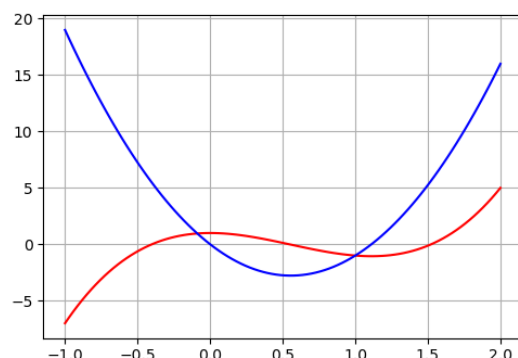
```
if pos == -1:
    print(f"Das Zeichen '{ch}' wurde nicht gefunden.")
else:
    print(f"Das Zeichen '{ch}' wurde an Position {pos} gefunden.")
```

Aufgabe 4: (ca. 10 Punkte)

Schreiben Sie ein Python-Skript, welches den Verlauf der Funktion $f(x) = 3x^3 - 5x^2 + 1$ im Bereich von $-1 \leq x \leq +2$ zusammen mit deren Ableitung $f'(x)$ grafisch auf dem Bildschirm ausgibt. In der Abbildung soll auch das Koordinatensystem sichtbar sein (siehe Bildschirmfoto).

Die Funktion $f(x)$ soll in roter Farbe, die Ableitung $f'(x)$ soll in blauer Farbe gezeichnet werden.

Tipp: Am einfachsten ist es, wenn Sie die Formel für die Ableitung $f'(x)$ selbst berechnen.



Aufgabe 5: (ca. 7 Punkte)

- 5.1. Wandeln Sie die folgende Dezimalzahl in eine Dualzahl (Binärsystem) um. Geben Sie nicht nur das Endergebnis, sondern auch alle Zwischenschritte Ihrer Berechnung an.

$$149_{\text{DEZ}} = ??_{\text{BIN}}$$

- 5.2. Wandeln Sie dieselbe Zahl 149_{DEZ} ins Hexadezimalsystem um.

$$149_{\text{DEZ}} = ??_{\text{HEX}}$$

(Platz für Notizen und Nebenrechnungen)