

Ingenieurinformatik I

Programmierung

Name	Vorname	Semestergruppe	Hörsaal

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Summe

Prüfer: Küpper, Ressel, Straub, Tasin

Bearbeitungszeit: 60 Minuten

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

Die Bearbeitung darf ausschließlich auf den ausgeteilten Prüfungsbögen erfolgen; mitgebrachte oder vorgefertigte Blätter werden nicht bewertet.

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

Aufgabe 1: (ca. 25 Punkte)

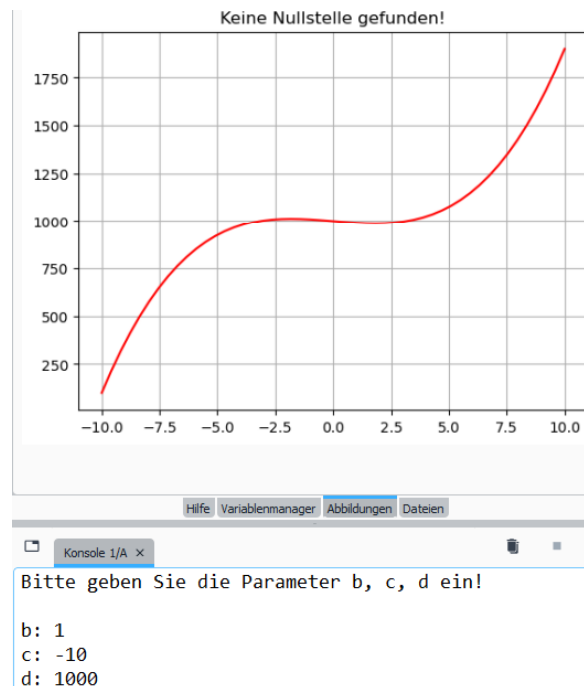
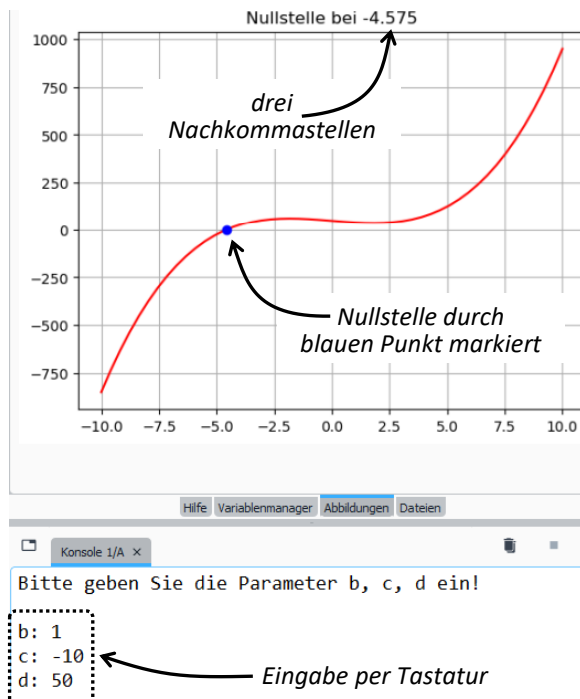
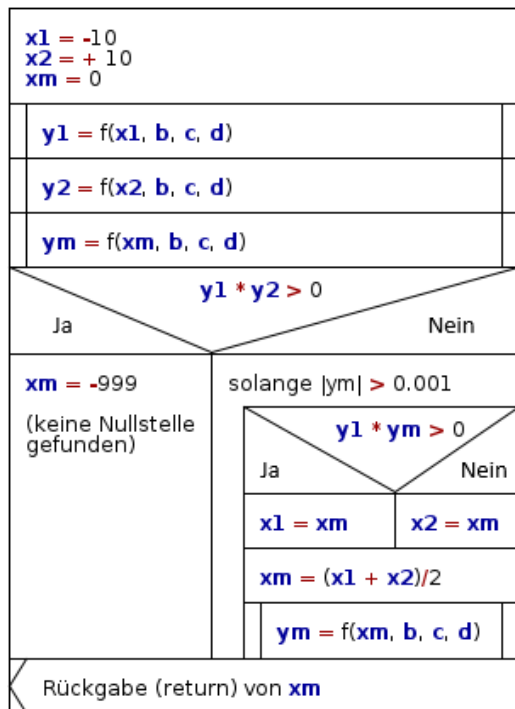
Schreiben Sie ein Python-Skript zur Berechnung einer Nullstelle der Funktion $f(x) = b \cdot x^3 + c \cdot x + d$.

- Programmieren Sie gleich zu Beginn Ihres Skripts die Python-Funktion $f(x, b, c, d)$ zur Berechnung von $f(x)$ unter Berücksichtigung der Parameter b, c, d .
- Das Struktogramm zeigt den Ablauf der Python-Funktion $\text{nullst}(b, c, d)$. Schreiben Sie auch diese Funktion. Ihre Funktion soll exakt dem Struktogramm entsprechen. Vergessen Sie nicht, in der Schleifenbedingung den Betrag von y_m zu bilden.

Die Bildschirmfotos zeigen den weiteren Ablauf:

- Zunächst werden die Werte b, c, d eingegeben (Eingabeaufforderung inkl. Überschrift im Bildschirmfoto beachten!). Eine Überprüfung der Eingabe ist nicht nötig: Gehen Sie davon aus, dass stets „vernünftige“ Werte eingegeben werden.
- Der Verlauf von $f(x)$ im Intervall $-10 \leq x \leq +10$ wird grafisch dargestellt. Denken Sie auch an die Ausgabe des Koordinatensystems (Gitterlinien).
- Anschließend ermittelt Ihr Skript mithilfe der zuvor programmierten Funktion $\text{nullst}(b, c, d)$ eine Nullstelle der Funktion $f(x)$.
- Je nachdem, ob eine Nullstelle gefunden wurde oder nicht, wird der Titel der grafischen Darstellung entsprechend gewählt (beachten Sie die Bildschirmfotos!). Die Nullstelle wird außerdem in der grafischen Darstellung markiert.

Funktion $\text{nullst}(b, c, d)$



(Hinweis: Es werden keine Ergebnisse auf der Konsole ausgegeben, zum Beispiel mittels `print`.)

Aufgabe 2: (ca. 17 Punkte)

- 2.1. Vervollständigen Sie das folgende Struktogramm: Es werden nacheinander 1000 Zufallszahlen im Bereich -5000 ... +5000 generiert. Ihr Programm/Struktogramm soll die größte und die kleinste Zufallszahl ermitteln und auf dem Bildschirm ausgeben.

Außerdem soll der arithmetische Mittelwert (Durchschnitt) der Zufallszahlen berechnet und ausgegeben werden.

Die Funktionen `max()`, `min()` oder andere fertige Funktionen zur Berechnung des Maximums bzw. des Minimums dürfen nicht verwendet werden. Ebenso ist es nicht zulässig, Listen zu verwenden. Außer `randint()` dürfen keine weiteren Funktionen importiert werden.

(Hinweis: In dieser Aufgabe ist kein Python-Quellcode gefragt!)

Importiere die Funktion <code>randint()</code> aus dem Modul <code>random</code> .
--

- 2.2. Wie lautet die Ausgabe des folgenden Python-Skripts? Hinweis: Mit `end=""` wird beim `print`-Befehl kein Zeilenumbruch am Ende der Ausgabe eingefügt.

```
for j in range(5):
    k = 5 - j
    while k > 0:
        print("0", end="")
        k -= 1
    print("1")
```

Aufgabe 3: (ca. 8 Punkte)

- 3.1. Korrigieren Sie die fünf Fehler im folgenden Python-Skript zur Ermittlung von Primzahlen. Hinweis: Einer der fünf Fehler ist ein Logikfehler.
- 3.2. Erweitern Sie das Python-Skript, sodass alle in der Liste `pp` gespeicherten Primzahlen untereinander (!) auf dem Bildschirm ausgegeben werden.

```
pp = [] # Leere Liste anlegen

for n = range(100000, 200000):
    prim = True
    t = 1

    while t * t <= n:
        if n % t == 0:
            prim = False
            break
        t += 1

    if prim:
        pp.append(n)

print(f"{len(pp)} Primzahlen gefunden.")

# TODO: Primzahlen in Liste pp untereinander ausgeben.
```

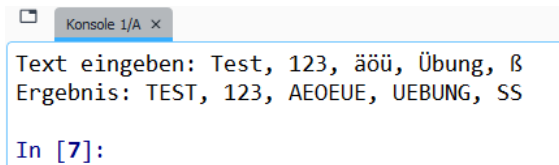
Aufgabe 4: (ca. 7 Punkte)

In dem folgenden Python-Skript wird die eingegebene Zeichenkette folgendermaßen verarbeitet:

- Kleinbuchstaben werden in Großbuchstaben umgewandelt, alle anderen Zeichen bleiben gleich.
- Alle Umlaute und das scharfe „ß“ werden ersetzt: Ä → AE, Ö → OE, Ü → UE, ß → SS
- Im Bildschirmfoto sehen Sie ein Beispiel. Beachten Sie auch die Eingabeaufforderung.

Schreiben Sie die fehlenden Befehle in die vorbereiteten Kästchen.

```
txt =   
ergebnis =   
for z   
    u =   
    if u == "Ä":  
        ergebnis += "AE"  
    elif u == "Ö":  
        ergebnis += "OE"  
    elif u == "Ü":  
        ergebnis += "UE"  
    elif z ==   
        ergebnis +=   
    else:  
        ergebnis +=   
print(f"Ergebnis: {ergebnis}")
```



Konsole 1/A x

Text eingeben: Test, 123, äöü, Übung, ß
Ergebnis: TEST, 123, AEOEUE, UEBUNG, SS

In [7]:

Aufgabe 5: (ca. 10 Punkte)

- 5.1. Eine Ganzzahl wird mit 8 Byte gespeichert. Wie viele unterschiedliche Werte lassen sich damit darstellen? (Angabe als Zweierpotenz genügt!) Hinweis: Es geht hier um die allgemeine Binär-darstellung von Ganzzahlen, nicht um die spezielle Python-Implementierung von int-Variablen.
- 5.2. Wie viele unterschiedliche float-Zahlen gibt es in Python maximal? Geben Sie eine Obergrenze an. (Angabe als Zweierpotenz genügt!) Hinweis: float-Zahlen belegen im Speicher des Rechners 8 Bytes. Es reicht, die theoretische Obergrenze aller möglichen Bitmuster anzugeben, auch wenn einige Bitmuster keine „echten“ Zahlen darstellen (NaN, $+\infty$, $-\infty$...).
- 5.3. Wandeln Sie die folgende Dezimal-Kommazahl in eine Dual-Kommazahl (Binärkommazahl) um. Geben Sie alle Zwischenschritte der Umwandlung an. Hinweis: Es ist nur die Dual-Kommazahl gefragt, nicht deren 32-Bit-Fließkommadarstellung.

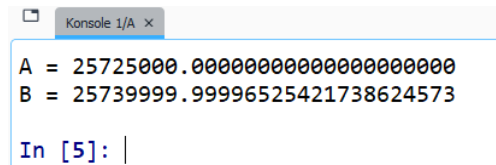
$$257,25_{10} = ???_2$$

- 5.4. Schauen Sie sich die Bildschirm-Ausgabe des folgenden Python-Skripts an. Erklären Sie, warum bei (A) ein exaktes („glattes“) Ergebnis erscheint, während bei (B) ein Ergebnis mit vielen Nachkommastellen angezeigt wird.

```
a = 257.25
b = 257.40
A = 0
B = 0

for i in range(100000):
    A += a
    B += b

print("")
print(f"A = {A:.20f}")
print(f"B = {B:.20f}")
```



```
Konsole 1/A x
A = 25725000.00000000000000000000
B = 25739999.99996525421738624573

In [5]: |
```

(Platz für Notizen und Zwischenrechnungen)