

---

Übung zur Vorlesung  
**Informatik I (B.Sc. Physik)**  
SoSe 2020

## Aufgabenblatt 2: Alternativen, Schleifen, Zahlendarstellungen

---

Ausgabe: 02.06.2020

Abgabe: 09.06.2020

Die Besprechung der Übungsaufgaben findet eine Woche nach Abgabetermin als Video-Konferenz im Übungszeitraum statt. Beachten Sie die Hinweise zur Übung aus der Einführungsveranstaltung!

---

### Aufgabe 1 Darstellung ganzer Zahlen

(7 Punkte)

a) Vervollständigen Sie die folgende Tabelle:

dezimal	dual	8 Bit-Zweierkomplement
13	$(1101)_2$	00001101
-13	$(-1101)_2$	11110011
53		
	$(-1011001)_2$	
		10101010
		01010101

b) Berechnen Sie durch **Addition** in der 8 Bit-Zweierkomplement-Darstellung:

$$51 - 26 = 51 + (-26)$$

$$67 - 92 = 67 + (-92)$$

### Aufgabe 2 Darstellung rationaler Zahlen

(6 Punkte)

a) Wandeln Sie die Dezimalzahlen in die Dualzahl-Darstellung um:

$$12.0, 1.2, 5 \cdot 10^{-2}$$

b) Wandeln Sie die **Dualzahlen** in die dezimale Darstellung um:

$$0.1101, 1.101101, 1.01 \cdot 10^{-10}$$

### Aufgabe 3 Quadratische Gleichung

(6 Punkte)

Schreiben Sie ein C++-Programm zur Bestimmung der reellen Lösungen einer quadratischen Gleichung der Form  $a \cdot x^2 + b \cdot x + c = 0$  ! Lassen Sie sich dazu die Koeffizienten der Gleichung vom Anwender über die Konsole eingeben und geben Sie die Lösungen in Textform aus.

*Hinweise:*

- Die Wurzel eines double-Wertes lässt sich mit der Funktion `sqrt()` berechnen. Zur Anwendung dieser Funktion muss ein `#include <cmath>` am Anfang Ihres Programms eingefügt werden.
- Gehen Sie schrittweise vor: Berechnen Sie die Koeffizienten der Normalform  $x^2 + p \cdot x + q = 0$ , die Diskriminante und dann die Lösungen.
- Beachten Sie die Fälle, in denen keine Lösung existiert.

### Aufgabe 4 Zahlenraten

(6 Punkte)

Der Nutzer des Programms soll eine ganze Zahl aus dem Intervall von Null bis zu einem Maximum raten. Er gibt dazu jeweils eine Zahl an und erhält als Antwort, ob er die Zahl erraten hat oder ob die zu suchende Zahl größer oder kleiner ist als seine Eingabe. Am Ende soll ausgegeben werden, wieviele Versuche der Nutzer zum Raten gebraucht hat. Lassen Sie das Maximum vom Nutzer eingeben.

Hinweis: 'Würfeln' Sie die zu suchende Zahl mittels `rand()`. Um die Werte auf den Bereich 0..max zu beschränken, verwenden Sie `rand() % (max + 1)`. Zur Verwendung von `rand()` müssen Sie im Kopf ihres Programms ein `#include <cstdlib>` aufnehmen.

**Viel Erfolg!**