

1. Übungsblatt

Ziel: Einstieg in die Python-Programmiersprache

1. Aufgabe (4 Punkte)

In der Bildverarbeitung werden unterschiedliche Formate verwendet, um Farben darzustellen.

Es gibt z.B. das RGB-Format, in dem mit Hilfe von drei ganzen Zahlen zwischen 0 und 255 (Rot, Grün und Blau-Werte) die Farben kodiert werden.

In Zeitschriften und Büchern wird das CMYK-Format verwendet, in dem Bilder aus einer Kombination der Farben Cyan, Magenta, Yellow und Black (CMYK) gedruckt werden.

Schreiben Sie ein Python-Programm mit dem Namen `RGB_2_CMYK`, das unter Verwendung folgender Formel nach Eingabe der RGB-Werte entsprechende CMYK-Werte ausgibt.

Wenn $RGB = (0,0,0)$, dann $CMYK = (0,0,0,1)$;

sonst werden die Werte wie folgt berechnet:

$$\begin{aligned}w &= \max(R / 255, G / 255, B / 255) \\C &= (w - (R / 255)) / w \\M &= (w - (G / 255)) / w \\Y &= (w - (B / 255)) / w \\K &= 1 - w\end{aligned}$$

2. Aufgabe (4 Punkte)

Eine Mersenne-Zahl hat die Form $2^n - 1$. Schreiben Sie eine Python-Funktion, die unter Verwendung einer Schleife und bei Eingabe von **n** die entsprechende Mersenne-Zahl berechnet.

Wann ist es besser, **for**-Schleifen statt **while**-Schleifen zu verwenden?

3. Aufgabe (4 Punkte)

Eine Annäherung der Zahl π kann mit Hilfe folgender Reihe wie folgt berechnet werden:

$$R_n = \sum_{i=1}^n \frac{1}{i^2} = \frac{1}{1^2} + \frac{1}{2^2} + \dots + \frac{1}{n^2} \approx \frac{\pi^2}{6}$$

Programmieren Sie eine Python-Funktion, die die Zahl π bei Eingabe von n annähern soll. Verwenden Sie dafür eine **for**-Schleife (keine **while**-Schleife).

4. Aufgabe (7 Punkte)

Schreibe eine Python-Funktion, die bei Eingabe einer ganzen Zahl und entsprechender Speicher-Wortlänge die Binärkodierung der Zahl berechnet.

- a) Positive und negative Zahlen müssen korrekt unter Verwendung der Zweierkomplement-Darstellung für die negative Zahlen berechnet werden.
- b) Beim Eingabe von zu großen Zahlen, deren Kodierung die Wortlänge des Speichers überschreiten, muss eine entsprechende Fehlermeldung ausgegeben werden.

Wichtige Hinweise:

- 1) Verwenden Sie geeignete Namen für Ihre Variablen und Funktionsnamen, die den semantischen Inhalt der Variablen oder die Funktionalität der Funktionen darstellen.
- 2) Verwenden Sie vorgegebene Funktionsnamen, falls diese angegeben werden.
- 3) Kommentieren Sie Ihr Programme.
- 4) Verwenden Sie geeignete Hilfsvariablen und Hilfsfunktionen in Ihren Programmen.
- 5) Löschen Sie alle Programmzeilen und Variablen, die nicht verwendet werden.
- 6) Schreiben Sie getrennte Test-Funktionen für alle 4 Aufgaben für Ihren Tutor.