Repetition3

Dr. Günter Kolousek

21. Juli 2015

Lege wiederum ein Verzeichnis an. Nennes es `10_repetition3`! In diesem Verzeichnis sollen alle Dateien der jeweiligen Einheit abgelegt werden.

1 Wiederholung 3

1. Schreibe eine Funktion `add_times(h1, m1, s1, h2, m2, s2)`, die zwei Zeiten addiert. Jede Zeit ist gegeben durch einen Stundenanteil (`h1` bzw. `h2`), einen Minutenanteil (`m1` bzw. `m2`) und einen Sekundenanteil (`s1` bzw. `s2`).
2. Erweitere die Funktion `add_times()` so, dass jetzt auch Tage verarbeitet werden. Parameter `t1` bzw. `t2`!
3. Schreibe eine Funktion `add_fractions(n1, d1, n2, d2)`. "n" steht für "numerator" also zu Deutsch "Zähler" und "d" steht für "denominator" also zu Deutsch "Nenner".

   Beim Bruchrechnen bildet sich der Zähler und der Nenner des Ergebnisses folgendermaßen:

   $$n = n1 \cdot d2 + n2 \cdot d1$$
   $$d = d1 \cdot d2$$

4. Schreibe nun die weiteren Funktionen für das Bruchrechnen:
   - `sub_fractions(n1, d1, n2, d2)` zum Subtrahieren.
   - `mul_fractions(n1, d1, n2, d2)` zum Multiplizieren.
   - `div_fractions(n1, d1, n2, d2)` zum Dividieren.

5. Schreibe ein Hauptprogramm `fracmath.py`, das es dem Benutzer ermöglicht zwischen Addition, Subtraktion, Multiplikation und Division von Bruchzahlen auszuwählen. Danach gibt der Benutzer die beiden Bruchzahlen ein und das Ergebnis wird ausgegeben.

6. Schreibe eine Funktion `cnt_ones(b)`, die die Anzahl der Einser in einem String zählt, der nur die Zeichen "0" bzw. "1" enthält und diese Anzahl zurückliefert.

   Also der Funktionsaufruf `cnt_ones("1010100")` soll also 3 zurückliefern.

7. Schreibe eine Funktion `add_parity(b)`, die ein "Prüfbit" zu einem "Bitstring" hinzufügt.

   Folgendes Problem: Ein Bitmuster, das aus sieben Bit besteht, soll in einem Computernetz sicher übertragen werden. Der Sender der Nachricht fügt deshalb dem Bitmuster ein Prüfbit hinzu. Das dabei entstehende Bitmuster soll eine gerade Anzahl von Einsen enthalten. Der Empfänger der Nachricht überprüft, ob das Bitmuster auch wirklich eine ungerade Anzahl von Einsen enthält. Durch dieses Vorgehen sollen Übertragungsfehler erkannt werden.

   Schreibe dazu eine Funktion `add_parity(b)`, die zu einem String, der nur aus den Zeichen "0" und "1" besteht ein zusätzliche Zeichen hinzufügt, sodass eine gerade Anzahl an "1" Zeichen enthalten ist und diesen String danach zurückliefert.

   Verwende die schon entwickelte Funktion `cnt_ones()`!

8. Schreibe jetzt noch eine Funktion `check(b)`, die `True` zurückliefert, wenn keine "Übertragungsfehler" aufgetreten sind und ansonsten `False`.

9. Schreibe eine Funktion, die zu einer ISBN-10 Nummer die Prüfziffer berechnet.

   Eine ISBN (International Standard Book Number) besteht aus zehn Zeichen (wenn man die Minuszeichen nicht mitzählt). Das zehnte Zeichen stellt eine Prüfziffer dar. Für die ISBN 3897212161 lautet die Prüfziffer 1. Die Prüfziffer berechnet sich wie folgt. Zuerst wird die folgende Summe gebildet:

   $1 \cdot 3 + 2 \cdot 8 + 3 \cdot 9 + 4 \cdot 7 + 5 \cdot 2 + 6 \cdot 1 + 7 \cdot 2 + 8 \cdot 1 + 9 \cdot 6 = 166$

   Die Prüfziffer wird als Rest der Division von 166 durch 11 gewonnen:

   $$166 \bmod 11 = 1$$

   Ergibt sich bei der Berechnung der Prüfziffer die Zahl 10, dann wird statt dieser das Zeichen `X` verwendet!

   Schreibe eine Funktion `isbn1_checknumber(isbn)`, die `True` zurückliefert, wenn die ISBN syntaktisch korrekt ist, anderenfalls `False`.

10. Schreibe ein Programm, das den Flächeninhalt eines beliebigen Dreieckes berechnet, wenn die 3 Seiten a, b, c gegeben sind und sich die Fläche folgendermaßen berechnen lässt:

$$s = (a + b + c)/2$$
$$A = \sqrt{s(s-a)(s-b)(s-c)}$$