

# Hausaufgabe 1 (1) (5 Punkte)

Abgabe bis 16.11.2014

## Aufgabe Römische Zahlen

Das römische Zahlensystem ist ein additives System, bei dem es verschiedene Ziffern für bestimmte Zahlenwerte gibt:

1000	500	100	50	10	5	1
M	D	C	L	X	V	I

Eine Zahl wird gebildet, indem sie (evtl. mehrfach) in die jeweils größtmöglichen passenden Ziffern zerlegt wird, z.B.  $1924 = 1000 + 500 + 100 + 100 + 100 + 100 + 10 + 10 + 1 + 1 + 1 + 1 \rightarrow \text{MDCCCCXXIII}$

- a) Schreiben Sie eine Funktion, die eine als Funktionsargument übergebene natürliche Zahl im römischen Zahlensystem auf dem Bildschirm ausgibt.
- b) Bei einer moderneren Variante des römischen Zahlensystems darf es nie mehr als drei aufeinander folgende gleiche Ziffern geben. Stattdessen bewirkt das Voranstellen der nächstkleineren Ziffer an eine Ziffer deren Subtraktion (Subtraktionsregel), z.B. IIII für 4 ist nicht zulässig, stattdessen muss IV (5–1) verwendet werden oder IX (10–1) für 9 statt VIII usw. Schreiben Sie eine weitere Funktion, die eine natürliche Zahl unter Berücksichtigung der Subtraktionsregel auf dem Bildschirm ausgibt.

# Hausaufgabe 1 (2) (5 Punkte)

Abgabe bis 16.11.2014

c) Schreiben Sie eine `main()`-Funktion, die eine Integerzahl ausschließlich mit Hilfe der aus der Vorlesung bekannten Funktion `readint()` von der Tastatur einliest und mit Hilfe der Funktionen aus a) und b) als römische Zahlen auf dem Bildschirm ausgibt.

Die eingelesene Zahl muss eine natürliche Zahl  $< 4000$  sein. Ist diese Vorbedingung nicht erfüllt, geben Sie statt der römischen Zahlen eine Fehlermeldung aus.

## Beispiel

**Geben Sie eine natürliche Zahl  $< 4000$  ein : 1999**

**Die Zahl 1999 im altrömischen System lautet: MDCCCLXXXVIII**

**Die Zahl 1999 im neuromischen System lautet: MCMXCIX**

# Hinweise zu Hausaufgabe 1

## Vorausgesetzter Vorlesungsstoff

Für Hausaufgabe 1 ist **ausschließlich** das Verständnis des **Vorlesungsstoffs der Einheiten 1 bis 3** (C-Tutorial) und des dazugehörigen Übungskomplexes (Einfache Programme) notwendig.

## Verwendbare Konstrukte

- Steuerstrukturen Sequenz `{ ; ... ; }`, Selektion `if-else` und Iteration `while`
- Datentyp Integer `int`
- Bildschirmausgabe **ausschließlich** mit der Funktion `printf()` der Bibliothek `<stdio.h>`
- Integer-Werteeingabe **ausschließlich** mit der in der Vorlesung vorgestellten Funktion `readInt()`
- einfache Ausdrücke mit Grundrechenarten `+`, `-`, `*`, `/` und Restwertberechnung `%`
- einfache Vergleichsoperationen (Bedingungen) mit **einem** Vergleichsoperator
- Definition und Aufruf einfacher Funktionen