



Übungsblatt 2

Willkommen zum Praktikum zu Programmieren 3.

Aufgabe 1. Zeichnen Sie in einen Speicher (je Byte ein Kästchen) der bei 0x00 beginnt den Speicherverbrauch und die Belegung der folgenden Definitionen ein.

```
1 int i=1;
2 int ia[2];
3 short int sa[3] = {1, 2, 3};
4 char ca[6] = "hi";
```

Nehmen Sie an die Definitionen wären innerhalb `main`. Was würde sich ändern, wenn die Definitionen außerhalb von `main` wären?

Aufgabe 2. Zeichnen Sie ein Bild im PPM-Format, ein einfaches textbasiertes Format, was z.B. für ein RGB Bild mit 800 Pixel Breite und 600 Pixel Höhe wie folgt beginnt.

```
P3
800 600      Ab der vierten Zeile steht jede Zeile für ein Pixel von links oben nach
255          rechts unten mit drei durch Leerzeichen getrennte Zahlen, die für den
0 255 0      Farbanteil rot, grün, dann blau stehen und zwischen 0 und 255 liegen; im
              Beispiel ein grüner Punkt links oben.
```

Verwenden Sie ein globales zweidimensionales Feld für das Bild. Schreiben Sie folgende Zeichenprimitive als Funktionen: Hintergrund füllen; Rechteck anhand von zwei Eckpunkten zeichnen, beziehungsweise Kreis anhand von Mittelpunkt und Radius, jeweils ungefüllt mit Angabe der (nach innen gehenden) Rahmendicke und gefüllt. Übergeben Sie jeweils eine Farbe als Integer-Wert. Stellen Sie sicher, dass das zu zeichnende Objekt abgeschnitten wird, wenn es außerhalb des Bilds liegt. Schreiben Sie ein Programm `zeichne`, das ein (nettes) PPM-Bild mit den Funktionen erzeugt und auf der Standardausgabe ausgibt. Anschauen z.B. mit `./zeichne > bild.ppm; display bild.ppm`

Aufgabe 3. Das Soundex-Verfahren wurde in den USA verwendet, um Volkszählungsdaten nach ähnlich klingenden Nachnamen (gleicher Soundex-Wert) zu gruppieren und so schreibfehlertolerant nach englischen Wörtern zu suchen. Der Soundex-Wert eines Wortes berechnet sich wie in folgender Tabelle dargestellt, dabei ist der erste Buchstabe des Soundex-Werts der erste Buchstabe des Worts.

Buchstabe	BFPV	CGJKQSZXZ	DT	L	MN	R	AEIOUWYH
Ziffer	1	2	3	4	5	6	-

Jeder nachfolgende Buchstabe wird (unabhängig von Groß- oder Kleinschreibung) schrittweise anhand voriger Tabelle in eine Ziffer umgewandelt oder ignoriert. Falls die Soundex-Ziffer gleich der vorhergehenden Soundex-Ziffer ist, dann wird sie ignoriert. Der Soundex-Wert besteht aus maximal 6 Zeichen, weitere Zeichen werden ignoriert. Wenn der Soundex-Wert weniger als 6 Zeichen hat, dann wird der Soundex-Wert mit der Ziffer 0 aufgefüllt.



Implementieren Sie ein Programm `soundex`, das beliebig viele Worte als Kommandozeilenparameter entgegen nimmt (je Wort maximal 99 Zeichen) und den Soundex-Wert jedes Wortes berechnet. Beispiel:

```
$ soundex soundex soundeggs flurbel
1          soundex : s53200
2          soundeggs : s53200
3          flurbel  : f46140
```

Aufgabe 4. Erweitern Sie `rot13` vom letzten Übungsblatt. Erstellen Sie eine Bibliothek, die eine Funktion `int rotate(int ch)` zur Verfügung stellt, die ein Zeichen konvertiert. Mit der Funktion `void set_rotate(int n)` wird die Anzahl um wie viel das Zeichen rotiert wird gesetzt. Ohne Aufruf von `set_rotate` soll um 0 verschoben werden.

Schreiben Sie nochmal ein Programm `rotlib13`, verwenden Sie diesmal Ihre Bibliothek. Schreiben Sie zwei weitere Programme `rotlib8` und `rotlib1`, die um 8 beziehungsweise 1 Zeichen rotieren. Testen Sie Ihre Programme, in dem Sie jeweils eine Datei so lange rotieren bis wieder die Original-Datei herauskommt.

Verwenden Sie in der Aufgabe separate Übersetzung und Header-Dateien. Schreiben Sie vollständiges Makefile, dass so wenig wie möglich übersetzt und so viel wie möglich gemeinsame Aufgaben abstrahiert. Stellen Sie sicher, dass mit `test` entsprechende Tests ausgeführt werden.

Aufgabe 5. Schreiben Sie ein Programm `roemisch`, das auf der Kommandozeile entweder eine positive ganze römische Zahl akzeptiert und als Dezimalzahl ausgibt oder umgekehrt. Wir akzeptieren Zahlen bei denen als Subtraktion I nur vor V und X stehen kann, X nur vor L und C stehen kann und C nur vor D und M stehen kann. Sie können davon ausgehen, dass die Zahlen nicht größer als 10.000 werden und die Eingaben korrekt sind. 9888 als römische Zahl hat die meisten Zeichen (21).

Hinweis 1. Lesen Sie in dem Informationsblatt “Umgebung für Entwicklung in C” den Abschnitt “Fortgeschrittenes Automatisieren mit make”.

Hinweis 2. Beginnen Sie die Aufgaben aus den “Wiederholaufgaben – C” als Wiederholung des Vorlesungsstoffs durchzuarbeiten.

<http://www.mi.hs-rm.de/~barth/hsrm/prog3>