



## Lehrveranstaltung "Informatik I für TI-Bachelor"

### Übungsblatt 10

#### Hinweise:

Dieses Übungsblatt ist zur Zulassung zu der Klausur erfolgreich zu bearbeiten ("*Erfolgreich*" bedeutet: Keine Programmabstürze bzw. Endlosschleifen, Aufgabenstellung einschl. der Nebenbedingungen müssen eingehalten sowie Kommentierung und Einrückung korrekt sein!).

Die Aufgaben werden überwiegend in den Übungszeiten bearbeitet. Allerdings genügt die Zeit hierfür unter Umständen nicht, so dass Sie auch außerhalb dieser Zeiten die Aufgaben bearbeiten müssen. Der Abgabetermin für diese Aufgabe ist **spätestens** der **24. Juli 2015**.

Nutzen Sie die Übungen auch, um ggf. Fragen, die sich in den Vorlesungen ergeben haben, anzusprechen.

---

**Aufgabe:** Ziel dieser Übung ist das Verwenden von Zeigern auf Arrays und als Parameter in Funktionen.

Schreiben Sie ein Programm, welches die Anzahl der verschiedenen Zeichen in einer vom Anwender eingegebenen Zeichenkette zählt und diese tabellarisch auf dem Bildschirm ausgibt.

Verwenden Sie im Hauptprogramm jeweils ein Array für den eingegebenen Text (Zeichenkette) und für die Anzahl der einzelnen Zeichen (Array von `int`). Das Array für die Anzahl der Zeichen sollte 256 Elemente enthalten; für jeden ASCII-Wert ein Arrayelement. Dann können die ASCII-Werte der einzelnen Zeichen als Index für das Array verwendet werden. Auf keinen Fall dürfen diese Arrays als globale Variablen angelegt werden! Um in den Funktionen auf diese Arrays zugreifen zu können, müssen diese als Parameter an die Funktionen übergeben werden. **In den Funktionen soll nur mit Zeigern auf die Arrays zugegriffen werden!**

Der Benutzer soll einen beliebigen Text eingeben dürfen (mindestens aber 1 Zeichen), von dem nur die ersten 150 Zeichen verwendet werden sollen. Anschließend werden die verschiedenen Zeichen des Textes (auch deutsche Umlaute, Ziffern, Leerzeichen, Sonderzeichen, usw.; also wirklich alle Zeichen mit den ASCII-Werten von 0 bis 255) gezählt und die Anzahl in dem Array gespeichert. Groß- und Kleinschreibung soll dabei nicht unterschieden werden. Am Ende wird die Anzahl der einzelnen Zeichen tabellarisch ausgegeben (nur die Zeichen, die auch im Text vorkommen!), und zwar so, dass die gesamte Tabelle auf einer Bildschirmseite zu sehen ist. D.h. es muss eine mehrspaltige Tabelle verwendet werden (siehe Beispiel). Dabei soll jeweils erst das Zeichen – *bei ASCII-Werten unter 32 ein Leerzeichen* –, dahinter in Klammern der he-

xadezimale ASCII-Wert des Zeichens und dann die Anzahl der gezählten Zeichen ausgegeben werden. Beachten Sie auch, dass Sie die Tabelle in der letzten Zeile bis zum rechten Rand ausgeben, auch wenn es keine weiteren Einträge mehr gibt!

**Beispieltext (Wie kommt man nur auf einen solch unsinnigen Satz?):**

Franz jagt im komplett verwahrlosten Taxi quer durch Bayern.

Um sich die Tipparbeit für diesen Beispielsatz zu sparen, können Sie diesen Satz markieren, dann mit `Strg+C` in die Zwischenablage kopieren, und schließlich je nach Terminal über den Menübefehl "Bearbeiten, Einfügen" bzw. der Tastenkombination `Umschalt+Strg+Einf` an der aktuellen Cursorposition wieder einfügen.

Am Ende soll der Benutzer gefragt werden, ob er denn noch einmal möchte oder nicht.

Der Quelltext soll wieder auf mehrere Module verteilt werden. So können z.B. die Funktionen zum Löschen des Tastaturpuffers und die Frage, ob noch einmal, in einem Modul geschrieben werden (die beiden Funktionen können von der letzten Aufgabe übernommen werden).

**Tipp:** Sollte es Probleme mit den deutschen Umlauten geben, müssen Sie die Zeichenkodierung des Terminalfensters auf ISO-8859-15 stellen (Wenn sich die Zeichenkodierung am Terminalfenster nicht einstellen lässt, kann mit dem Befehl `xterm -en iso-8859-15` ein neues Terminalfenster mit der gewünschten Zeichenkodierung öffnen)!

Das Programm soll benutzerfreundlich sein, d.h. dem Benutzer soll mitgeteilt werden, was er tun soll und was er falsch gemacht hat.

Das Compilieren, Linken und Starten des Programms soll wieder mittels einer Make-Datei durchgeführt werden.

### Beispiele:

```
=====
||           ||
|| Zeichenzaehler ||
||           ||
||           ||
=====
```

Geben Sie bitte den Text ein (mind. 1 Zeichen / max. 150 Zeichen), dessen Zeichen gezaehlt werden sollen:

Franz jagt im komplett verwahrlosten Taxi quer durch Bayern.

Anzahl der Zeichen:

```
-----
| (0x20): 8 | . (0x2e): 1 | a (0x61): 5 | b (0x62): 1 |
| c (0x63): 1 | d (0x64): 1 | e (0x65): 5 | f (0x66): 1 |
| g (0x67): 1 | h (0x68): 2 | i (0x69): 2 | j (0x6a): 1 |
| k (0x6b): 1 | l (0x6c): 2 | m (0x6d): 2 | n (0x6e): 3 |
| o (0x6f): 2 | p (0x70): 1 | q (0x71): 1 | r (0x72): 6 |
| s (0x73): 1 | t (0x74): 5 | u (0x75): 2 | v (0x76): 1 |
| w (0x77): 1 | x (0x78): 1 | y (0x79): 1 | z (0x7a): 1 |
-----
```

Moechten Sie noch einmal (j/n)? j

```
=====
||                               ||
||   Zeichenzaehler             ||
||                               ||
=====
```

Geben Sie bitte den Text ein (mind. 1 Zeichen / max. 150 Zeichen),  
dessen Zeichen gezaehlt werden sollen:

Dies            ist ein            Satz mit            Tabulatoren!

Anzahl der Zeichen:

```
-----
| (0x09): 4 | (0x20): 2 | ! (0x21): 1 | a (0x61): 3 |
| b (0x62): 1 | d (0x64): 1 | e (0x65): 3 | i (0x69): 4 |
| l (0x6c): 1 | m (0x6d): 1 | n (0x6e): 2 | o (0x6f): 1 |
| r (0x72): 1 | s (0x73): 3 | t (0x74): 5 | u (0x75): 1 |
| z (0x7a): 1 |           |           |           |
-----
```

Moechten Sie noch einmal (j/n)? \_