

Prozedurale Programmierung – Übung 10

WS 2018/19

Prof. Dr. F.J. Schmitt

Hochschule **Rosenheim**
University of Applied Sciences



In der Community wird ein (fast leeres) Projekt in der Datei „strings.zip“ bereitgestellt.

Aufgabe 1

Entwickeln Sie ein Programm, das eine beliebige Anzahl von Wörtern aus der Kommandozeile (! – **nicht** über `scanf()`!) einliest und eine Statistik über die Häufigkeit des Vorkommens der einzelnen Buchstaben erstellt.

Aufruf über Kommandozeile:

```
H:\... >strings argaaa argbbb AXY$$Z
```

Ausgabe:

```
-----  
Statistik Kommandozeile  
-----  
strings argaaa argbbb AXY$$Z  
Anzahl der Buchstaben der Parameter:  
$: 2 A: 1 X: 1 Y: 1 Z: 1 a: 5 b: 3 g: 2 r: 2
```

Anmerkungen:

- das Programm soll prinzipiell für beliebige der 256 ASCII-Zeichen funktionieren, sofern diese druckbar sind (also z.B. nicht für Leerzeichen oder newline), und mit Ausnahme von doppelten Anführungszeichen.
- die Statistik soll mit Hilfe eines Feldes erstellt werden; es bietet sich hier eines der Größe 256 an.
- Probleme, die durch Konvertierung von manchen Sonderzeichen (wie z.B. Umlaute) entstehen, können Sie vernachlässigen.
- um das Programm auf einfache Weise testen und wie o.g. starten zu können, sollten Sie eine Windows-Eingabeaufforderung öffnen und anschließend in das Verzeichnis wechseln, in dem Ihr selbst erstelltes Programm strings.exe liegt (im Verzeichnis „Debug“ des Projekts). Die Eingabeaufforderung finden Sie im Ordner „Zubehör“ des Startmenüs.

Aufgabe 2

Entwickeln Sie ein Programm, das eine beliebige Anzahl von Wörtern aus der Kommandozeile einliest und diese in umgekehrter Reihenfolge wieder ausgibt. Es soll dynamische Speicherverwaltung verwendet werden.

Aufruf über Kommandozeile:

```
H:\... >strings argaaa argbbb AXY$$Z
```

Ausgabe:

```
argaaa argbbb AXY$$Z  
AXY$$Z argbbb argaaa
```

Kopieren Sie die Wörter aus der Kommandozeile in ein Feld von Zeichenketten. Der Speicher des Felds soll dynamisch in der aktuell erforderlichen Größe alloziert werden. Geben Sie den Text aus. Invertieren Sie dann die Reihenfolge der Wörter und geben Sie den Text erneut aus. Geben Sie sämtlichen dynamisch allozierten Speicher wieder frei.

Implementieren Sie dazu folgende Funktionen:

```
void printText(char *text[], int nrWords);  
// gibt einen Text ((Feld von Zeichenketten) mit nrWords Worten aus  
  
void textReverse(char *text[], int nrWords);  
// Invertiert in einem Text (Feld von Zeichenketten) die Reihenfolge  
// der Wörter; die Invertierung erfolgt ausschließlich durch Vertauschung  
// von Zeigern, nicht durch umkopieren der Wörter!  
  
void deleteText(char *text[], int nrWords);  
// gibt den Speicherplatz für die Wörter des Textes wieder frei
```