OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG
INFORMATIK UND MATHEMATIK

Übungsaufgaben Blatt 8 - Strings und Arrays

PG 1 - Einführung in die Programmierung mit C - Prof. Dr. Ruben Jubeh

---

Aufgabe 1 : Scrabble Score

In Scrabble ist jedem Buchstaben eine Punktzahl zugeordnet, die von seiner Häufigkeit in der entsprechenden Sprache abhängt.

Für die englische Version gelten die folgenden Punktzahlen:

Punkte	Buchstaben
1	A, E, I, L, N, O, R, S, T, U
2	D, G
3	B, C, M, P
4	F, H, V, W, Y
5	K
8	J, X
10	Q, Z

Das Wort **FARM** ist folglich 9 Punkte wert: 4 für das F, jeweils einen für A und R und 3 für das M.
Schreiben Sie eine Funktion `int getWordScore(char word[])` welche die Punkte für ein übergebenes Wort zurückgibt.

Um zu überprüfen, wie viele Punkte ein einzelner `char` bringt können Sie die einzelnen Buchstaben in jeweils einem einzelnen `char` Arrays pro Punktegruppe abspeichern (z.B.: `char onePointChars[] = "AEILNORSTU";`). Jetzt können Sie eine neue Funktion erstellen, die prüft, ob ein bestimmter Buchstabe darin vorkommt und die entsprechende Punktzahl zurückgibt.

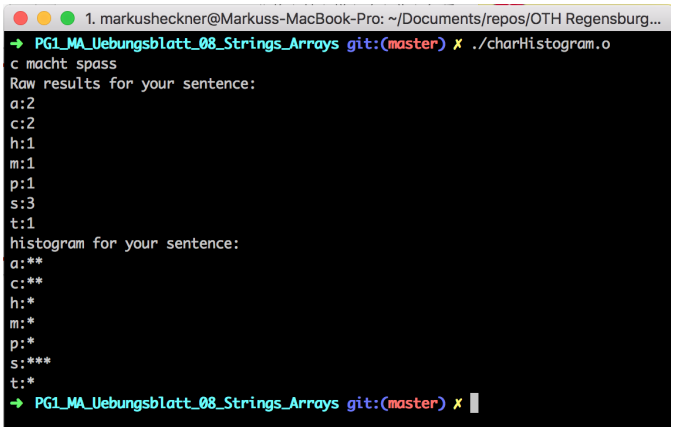
In einer ersten Version des Programms reicht es, wenn Sie die Punktzahl für Wörter aus Großbuchstaben errechnen.

Ihr Programm soll am Ende folgende Ausgabe produzieren können:

Aufgabe 2 : **Ermitteln der Buchstabenhäufigkeit in einem Text**

Erstellen Sie ein Programm welches eine Folge von **Kleinbuchstaben** bis max 1000 Zeichen in ein char-Array einliest. Ermitteln Sie für alle Kleinbuchstaben des Alphabets, wie oft der jeweilige Buchstabe in der Eingabe vorkommt. Geben Sie das Ergebnis tabellarisch aus. Verwenden Sie dafür die Information, dass die Kleinbuchstaben in der ASCII Tabelle in einer Reihe stehen. Erweitern Sie das Programm nun so, dass die Anzahl der gefundenen Buchstaben nun grafisch über einen Balken visualisiert wird. Die Größe des Balkens entspricht der Anzahl des Auftretens.

Eine Ausgabe des Programms für den String *hallo wie geht es ihnen* könnte wie folgt aussehen (Der String in diesem Beispiel ist im Programm hart codiert, Sie müssen den String in Ihrer Lösung vom Nutzer einlesen):



Optionale Erweiterung: Erweitern Sie Ihr Programm so, dass die Worthäufigkeit anhand eines größeren Textes bestimmt werden kann. Speichern Sie sich dazu beispielsweise den Text von Alice im Wunderland in einer Textdatei ab und lesen diesen für die Auswertung

ein (Quelle: `http://www.textfiles.com/etext/FICTION/alice13a.txt`). Achtung, in diesem Fall müssen Sie für das Histogramm relative Häufigkeiten verwenden, um nicht zu viele Sterne zu produzieren. Beispielsweise könnten Sie die höchste absolute Häufigkeit auf 30 Sterne setzen und die Häufigkeiten auf diese Weise skalieren.

Aufgabe 3 : Password Checker

Gute Passwörter sollten so aufgebaut sein, dass sie nicht leicht durch massives Ausprobieren aufgrund von Wörterbüchern und Zahlenfolgen erraten werden können.
Schreiben Sie eine Funktion `int isStrongPassword(char word[])` welche die Zeichen eines übergebenen Strings überprüft und nur dann `1` zurückgibt, wenn alle folgenden Bedingungen erfüllt sind.

Das Passwort besteht aus...

- ...mindestens 8 Zeichen,
- mindestens zwei Buchstaben,
- mindestens zwei Ziffern,
- mindestens einem Groß- und einem Kleinbuchstaben,
- mindestens einem Sonderzeichen (als Sonderzeichen gelten alle Zeichen außer Zahlen und Buchstaben).

Der folgende Screenshot zeigt die Ausgabe des Passwortcheckers, wenn auf minimale Zeichenanzahl des Passworts (hier: 8) und auf minimale Anzahl an Buchstaben (hier: 2) geprüft wird.