

## Prozedurale Programmierung

### Übungsaufgabe p11c Argumente von main()

Eine Möglichkeit, sichere Passwörter zu generieren, besteht darin sich einen Satz zu merken und jeweils nur die Anfangsbuchstaben der Wörter zu verwenden.

Beispiel:

*./genpass Es sind 120 Meilen bis Chicago wir haben Benzin im Tank es ist Nacht und wir haben Sonnenbrillen*

Passwort: „**Es1MbCwhBiTeiNuwhS**“ (Anfangsbuchstaben)

Aufgabenteil a): Schreiben Sie einen Passwortgenerator, der einen Text über die Kommandozeile entgegen nimmt und daraus eine Passwort generiert.

- Im Hauptprogramm ist ein char-Zeiger **passphrase** zu definieren.
- Allokieren Sie mit der **malloc()**-Funktion genau so viel Speicherplatz, dass Sie das generierte Passwort darin als Zeichenkette ablegen können.
- Implementieren Sie eine Funktion **genPasswd(...)**, die für jeden Kommandozeilenparameter den Anfangszeichen bestimmt und daraus die Passphrase erzeugt.
- Geben Sie den Satz und das Passwort im Hauptprogramm aus.
- Sie können die Bibliotheksfunktionen aus string.h verwenden.

Aufgabenteil b): Um das Verfahren noch sicherer zu machen, kann man bestimmte Buchstaben gegen Sonderzeichen oder Ziffern austauschen.

- Schreiben Sie eine Funktion **hardenPasswd(...)**, die innerhalb des Passwortes folgende Ersetzungen durchführt:

**S → \$    h → #    i → !    B → 8    x → %    C → (    e → 3**

- Verwenden Sie die switch-case Anweisung für diesen Aufgabenteil.
- Das gehärtete Passwort wird in der zweiten Zeichenkette im Hauptprogramm abgelegt.
- Geben Sie das gehärtete Passwort im Hauptprogramm aus.
- Geben Sie den belegten Speicher wieder frei.
- ACHTUNG! Die **hardenPasswd(...)**-Funktion soll nur dann aufgerufen werden, wenn der erste Parameter des Programmaufrufs ein „-h“ ist.

Beispiel:

*./genpass -h Es sind 120 Meilen bis Chicago wir haben Benzin im Tank es ist Nacht und wir haben Sonnenbrillen*

Passwort: „**Es1Mb(w#8!T3!Nuwh#\$**“

Es dürfen keine globalen Variablen verwendet werden.

Das Programm muss fehlerfrei und ohne Warnungen mit der Compileroption -Wall compilieren. Kommentieren Sie den Quellcode hinreichend.

Viel Spaß und verwenden Sie gute Passworte!

---

**Bewertungsschema:**

**Übersetzungsfehler ==> 0 P !**

sonst:	
Codeformatierung	2 P
Sinnvolle Kommentare	2 P
Bearbeitung der main()-Aufrufparameter	4 P
Aufgabenteil a) genPasswd()	4 P
Aufgabenteil b) hardenPasswd()	4 P
Hauptprogramm, malloc(), free() und Ausgaben	4 P
	-----
Summe:	20 P

---