

Hardwarenahe Programmierung

Übungsaufgaben – 5. Dezember 2019

Diese Übung enthält Punkteangaben wie in einer Klausur. Um zu „bestehen“, müssen Sie innerhalb von 60 Minuten unter Verwendung ausschließlich zugelassener Hilfsmittel 10 Punkte (von insgesamt 21) erreichen.

Aufgabe 1: Löschen aus Strings

Wir betrachten das folgende Programm ([aufgabe-1.c](#)):

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void delete_char_from_string (char *s, int pos)
5  {
6      for (int i = strlen (s) - 1; i > pos; i--)
7          s[i - 1] = s[i];
8  }
9
10 int main (void)
11 {
12     char test[] = "Hochschule_Boochum";
13     delete_char_from_string (test, 12);
14     printf ("%s\n", test);
15     return 0;
16 }
```

Die Ausgabe des Programms lautet: Hochschule Bmmmmmm

- (a) Erklären Sie, wie die Ausgabe zustandekommt. (3 Punkte)
- (b) Schreiben Sie die Funktion `delete_char_from_string()` so um, daß sie aus dem String `s` das Zeichen an der Stelle `pos` löscht.
Die Ausgabe des Programms müßte dann Hochschule Bochum lauten. (4 Punkte)
- (c) Was kann passieren, wenn Sie `char test[] = "Hochschule_Boochum";` durch `char *test = "Hochschule_Boochum";` ersetzen? Begründen Sie Ihre Antwort. (2 Punkte)
- (d) Schreiben Sie eine Funktion `void delete_from_string (char *s, int pos, int length)`, die `length` Zeichen ab der Stelle `pos` aus dem String `s` löscht, indem sie die in Aufgabenteil (b) korrigierte Funktion `void delete_char_from_string (char *s, int pos)` wiederholt aufruft.
(2 Punkte)
- (e) Schreiben Sie eine *optimierte* Funktion `void quick_delete_from_string (char *s, int pos, int length)`, die ebenfalls `length` Zeichen ab der Stelle `pos` aus dem String `s` löscht, aber mit einer einzigen Schleife auskommt. (4 Punkte)

Aufgabe 2: Hexdumps

Das folgende Programm ([aufgabe-2.c](#)) liest einen String ein und gibt die ASCII-Werte der Buchstaben hexadezimal aus. (Anders als z. B. `scanf()` akzeptiert die Funktion `fgets()` zum Lesen von Strings auch Leerzeichen, und sie vermeidet Pufferüberläufe.)

```
1  #include <stdio.h>
2
3  int main (void)
4  {
5      char buffer[100];
6      fgets (buffer, 100, stdin);
7      for (char *p = buffer; *p; p++)
8          printf ("%02x", *p);
9      printf ("\n");
10 }
```

Beispiel: Bei der Eingabe von `Dies ist ein Test.` erscheint die Ausgabe
`44696573206973742065696e20546573742e0a.`

Schreiben Sie ein Programm, das diese Umwandlung in umgekehrter Richtung vornimmt, also z. B. bei Eingabe von `44696573206973742065696e20546573742e0a` wieder `Dies ist ein Test.` ausgibt.

(6 Punkte)

Hinweis für die Klausur: Abgabe in digitaler Form ist erwünscht, aber nicht zwingend.

Viel Erfolg!