

3.1.A C-Datentypen/Datentyp char – Arbeitsblatt

Aufgabe 1: Ausgabe aller Zeichen

Das Programm `alleausgeben.c` soll alle Zeichen des ASCII-Zeichensatzes mit den Nummern 32 (Leerzeichen) bis 126 in aufsteigender Reihenfolge ausgeben:

```
schueler@debian964:~$ a.out
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e
f g h i j k l m n o p q r s t u v w x y z | ~
```

- Zeichnen Sie das Struktogramm!
- Schreiben Sie das Programm!

Aufgabe 2: Ausgabe eines Zeichen mit einer bestimmten Nummer

Das Programm `zahlzeichen.c` soll zu jeder eingegebenen Zahl (Datentyp: `int`) das entsprechende Zeichen ausgeben:

```
schueler@debian964:~$ a.out
Eingabe (Ende mit -1): 65
Zeichen: >>A<<
Eingabe (Ende mit -1): 98
Zeichen: >>b<<
Eingabe (Ende mit -1): -1
Tschuess!
```

- Zeichnen Sie das Struktogramm!
- Schreiben Sie das Programm!

Aufgabe 3: Ermitteln der Zeichenklasse

Das Programm `zeichenklasse.c` soll zu jeder eingegebenen Zahl angeben, zu welcher Zeichenklasse es gehört. Sie sollen also angeben, ob das entsprechende Zeichen eine Ziffer, ein Kleinbuchstabe, ein Großbuchstabe, ein Steuerzeichen oder ein anderes Zeichen ist:

```
schueler@debian964:~$ a.out
Eingabe (Ende mit -1): 65
Zeichenklasse: Grossbuchstabe
Eingabe (Ende mit -1): 11
Zeichenklasse: Steuerzeichen
Eingabe (Ende mit -1): -1
Tschuess!
```

- Schreiben Sie das Programm (`zeichenklasse_a.c`), wobei Sie zum Feststellen der Zeichenklasse die ASCII-Nummer verwenden (48...57: Ziffer, 0...31: Steuerzeichen, usw.)!
- Schreiben Sie das Programm (`zeichenklasse_b.c`), wobei Sie zum Feststellen der Zeichenklasse Funktionen aus `<ctype.h>` verwenden!

Aufgabe 4: Ausgabe aller Zeichenklasse, der ein Zeichen angehört

Das Programm `alleausmitklassen.c` soll nacheinander alle Zeichen mit den Nummern 32 bis 127 ausgeben, und zwar jedes Zeichen in einer eigenen Zeile. Hinter dem Zeichen soll angegeben sein, ob das Zeichen zu einer oder mehrerer der elf in C angegebenen Zeichenklassen (`alnum`, `alpha`, `cntrl`, `digit`, `graph`, `lower`, `print`, `punct`, `space`, `upper`, `xdigit`) gehört.

Terminal

```
schueler@debian964:~$ a.out
AACDGLPPSUX
-----
00000010100
! 00001011000
" 00001011000
# 00001011000
$ 00001011000
.....
z 11001110000
00001011000| 00001011000 00001011000
~ 00001011000
```

- a) Schreiben Sie dieses Programm (`alleausmitklassen_a.c`)!
- b) Erweitern Sie dieses Programm um die Zeichen 0...32 und 127 (`alleausmitklassen_b.c`)! Bei diesen Zeichen soll allerdings in der ersten Spalte nur ein Leerzeichen ausgegeben werden. Ausnahme: Bei den Zeichen mit den Ersatzdarstellungen `\f`, `\n`, `\r`, `\t` und `\v` soll am Anfang der Zeile die Ersatzdarstellung ausgegeben werden.

Aufgabe 5: Echo-Programm I

Das Programm `echo.c` soll zu jedem mit `getchar` eingegebenen Zeichen dasselbe Zeichen ausgeben. Bei Eingabe von `q` soll es abbrechen.

- a) Schreiben Sie dieses Programm!
- b) Woran erkennt man die Zeilenpufferung der Eingabe?

Aufgabe 6: Echo-Programm II

Das Programm `grossecho.c` soll zu jedem mit `getchar` eingegebenen Zeichen dasselbe Zeichen ausgeben, außer bei Kleinbuchstaben. Bei Eingabe eines Kleinbuchstabens soll der entsprechende Großbuchstabe ausgegeben werden. Bei Eingabe von `q` oder `Q` soll das Programm abbrechen.

- a) Schreiben Sie dieses Programm!

Aufgabe 7: Ausgabe der ASCII-Codenummer

Das Programm `asciiout.c` soll zu jedem mit `getchar` eingelesenen Zeichen mit `printf` die ASCII-Codenummer als Dezimalzahl ausgeben.

- a) Schreiben Sie dieses Programm!
- b) Testen Sie das Programm mit möglichst unterschiedlichen Eingaben!

Aufgabe 8: Ausgabe der Zeichenklasse

Man kann die alphanumerischen Zeichen in verschiedene Zeichenklassen einteilen: Kleinbuchstaben, Großbuchstaben, Ziffern, übrige Zeichen. Das Programm `zklasse.c` soll zu jedem mit `getchar` eingegebenen Zeichen sagen, zu welcher Zeichenklasse es gehört:

- Bei einem Kleinbuchstaben soll ein 'a' ausgegeben werden.
- Bei einem Großbuchstaben soll ein 'A' ausgegeben werden.
- Bei einem Ziffernzeichen soll eine '0' ausgegeben werden.
- Bei einem sonstigen Zeichen soll ein Punkt ausgegeben werden.

Aufgabe 9: Caesar-Verschlüsselung

Das bestehende Programm `caesar.c` soll verbessert werden (neuer Name: `caesar2.c`):

- Für jeden eingegebenen Buchstaben von A bis Y soll der im Alphabet folgende Buchstabe (also B bis Z) ausgegeben werden.
- Beim Eingabe des Buchstabens Z soll ein A ausgegeben werden.
- Kleinbuchstaben sollen genauso behandelt werden.
- Alle anderen ASCII-Zeichen werden genauso ausgegeben, wie sie eingegeben wurden.

Aufgabe 10: Zeilen zählen

Das Programm `zeilcnt.c` soll die Anzahl der Zeilen ermitteln, die die Eingabe enthält. Dazu liest es einfach Zeichen ein (bis zum EOF) und zählt eine Variable `anzahl` jedesmal dann hoch, wenn ein `\n`-Zeichen gelesen wurde.

- a) Zeichnen Sie ein Struktogramm zu dieser Aufgabe!
- b) Schreiben Sie das zum Struktogramm passende Programm!

Aufgabe 11: Zeilennummerierung

Das Programm `zeilnr.c` soll am Beginn jeder Zeile (also ganz zu Beginn der Eingabe und anschließend nach jedem eingegebenen `'\n'`) zuerst die Zeilennummer und dann den Rest der Zeile ausgeben:

```
Terminal
schueler@debian964:~$ ./zeilnr < caesar.c
/* 1 */ #include <stdio.h>
/* 2 */
/* 3 */ int main(void)
/* 4 */ {
    ...
/* 14 */     return 0;
/* 15 */ }
schueler@debian964:~$
```

Dabei soll – wie im Beispiel gezeigt – die Zeilennummer in Kommentarklammern stehen.