4 Ein-/Ausgabe - Übungen

Einführung scanf

- Lesen Sie eine Zahl mittels scanf() ein und geben Sie diese wieder in der Konsole aus:

```
Bitte geben Sie eine Zahl ein: 34

Sie haben 34 eingegeben!
```

L:

```c
int num;

printf("\nBitte geben Sie eine Zahl ein: ");
scanf("%d", &num);          // 1ter Parameter: Formatbezeichner, 2ter Parameter Variable MIT ADRESSOPERATOR &!

printf("\nSie haben %d eingegeben!\n", num);
```

Datum Einlesen (011)

- Lesen Sie ein Datum in der Form TT.MM.JJJJ ein und geben es in der Form JJJJ-MM-TT wieder aus:

```
Bitte geben Sie Ihr Geburtsdatum ein (TT.MM.JJJJ): 2.1.1999

Ihr Geburtsdatum: 1999-01-02
```

L:

```c
int tag, monat, jahr;
printf("Bitte geben Sie Ihr Geburtsdatum ein (TT.MM.JJJJ): ");

scanf("%d.%d.%d", &tag, &monat, &jahr);     // Nicht nur Ausgabe formatiert sondern auch Eingabe
printf("\nIhr Geburtsdatum: %04d-%02d-%02d\n", jahr, monat, tag);
```

Mehrfach Einlesen (012)

- Lesen Sie mittels scanf() einen Buchstaben von der Tastatur ein.
- Lesen Sie einen zweiten Buchstaben mittels scanf() von der Tastatur ein.
- Geben Sie beide Eingaben wieder aus:

```
Bitte geben sie den 1ten Buchstaben ein:a
Bitte geben sie den 2ten Buchstaben ein:b

Die beiden Buchstaben: a b
ASCII-Codes: 97 98
```

Wenn das Programm ausprobiert wird, sieht das Ergebnis wohl wie im Folgenden aus. Die 2te Eingabe wird übersprungen und die Ausgabe sieht so aus:

```
Bitte geben sie den 1ten Buchstaben ein:a
Bitte geben sie den 2ten Buchstaben ein:
Die beiden Buchstaben: a

ASCII-Codes: 97 10
```

Was ist der Grund für dieses Ergebnis?

L:

```
char a, b, temp;
// Einlesen eines einzelnen Zeichens
printf("Bitte geben sie den 1ten Buchstaben ein:");
scanf("%c",&a);
printf("Bitte geben sie den 2ten Buchstaben ein:");
scanf("%c",&b);
printf("\nDie beiden Buchstaben: %c %c\nASCII-Codes: %d %d\n", a, b, a, b);
```

scanf() mit %c ließt ein einzelnes Zeichen ein. Wird ein Buchstabe eingegeben und anschließend <ENTER> gedrückt werden zwei Zeichen eingegeben. Der Buchstabe und das <ENTER>-Zeichen. Wird anschließend ein zweites mal nach einem einzelnen Zeichen mit scanf() gefragt, dann entnimmt diese Abfrage das <ENTER>-Zeichen aus dem Tastaturstream. scanf() wartet also gar nicht auf eine zweite Eingabe.

Lösung: der Aufruf von `fflush(stdin)` leert den Tastaturstream (Eingabestream "stdin").

Eine andere Lösung ist für die erste Eingabe `scanf("%c\n" &a);` einzulesen, dann holt das scanf das eingegebene Zeichen UND die Eingabetaste ab. Für das nächste scanf ist der Stream dann leer.

Tipp: die gepufferten Streams können umgeleitet werden, siehe: https://de.wikibooks.org/wiki/Batch-Programmierung:_Batch-Operatoren

also z.B. `xy.exe < c:\temp\in.txt` oder `xy.exe < c:\temp\in.txt > c:\temp\out.txt`

Tastendruck (013)

- Lesen Sie den Tastendruck mittels getch() solange ein bis die **z**-Taste gedrückt wird:

```
Eingabe wird durch Taste "z" beendet.
abc die Katz

Process returned 0 (0x0)   execution time : 4.692 s
Press any key to continue.
```

L:

```
char c;
printf("Eingabe wird durch Taste \"z\" beendet.\n");
do {
    c = getch();
    printf("%c",c);
} while (c != 'z');
```

Pfeiltasten (060)

- Lesen Sie den Tastendruck mittels getch() ein und geben nur Pfeiltasten aus:

```
Benutze die Pfeiltasten oder druecke Esc zum Beenden!
Rauf!
Runter!
Links!
Rauf!
Rauf!
Links!
```

(77=Rechts, 75=Links, 80=Runter, 72=Rauf, 27=ESC)

L:

```c
char c;

printf("Benutze die Pfeiltasten oder druecke Esc zum Beenden!\n");
do {
    c = getch();
    if (c == 77) printf("Rechts!\n");
    else if (c == 75) printf("Links!\n");
    else if (c == 80) printf("Runter!\n");
    else if (c == 72) printf("Rauf!\n");
} while (c != 27);      // Esc
```

BufferedReadWrite (061)

- Lesen Sie den Tastendruck mittels **fgetc(stdin)** ein und gebe das Zeichen direkt mittels **fputc(stdout)** aus. Beenden Sie die Eingabe mit **fgetc(stdin) = EOF**. EOF steht für EndOfFile.

```
Beendet werden kann die Eingabe nur durch Halten der ALT-Taste und gleichzeitiger Eingabe des ASCII-Codes 26 am
Nummernblock der Tastatur. 26dez steht fuer EOF. Abschliessend muss Enter gedrueckt werden:
Tschüss→

Process returned 0 (0x0)   execution time : 29.503 s
Press any key to continue.
```

L:

```c
int ch = 0;

while (ch != EOF) {       // EOF für Windows: <ALT>+26
    fputc(ch, stdout);
    ch = fgetc(stdin);
}
```

Sterne 1 (111)

- Lesen Sie eine Zahl num von der Tastatur ein
- Geben Sie num viele Sterne * aus

```
Geben Sie die Anzahl der * ein: 15
***************
```

Sterne 2 (112)

- Lesen Sie eine Länge und eine Breite von der Tastatur ein
- Geben Sie ein Rechteck aus:

```
Geben Sie die Laenge und die Breite ein: 3.5
*****
*****
*****
```

Sterne 3 (113)

- Lesen Sie eine Zahl num von der Tastatur ein
- Geben Sie ein Dreieck aus:

```
Geben Sie die Anzahl der * ein: 6
******
*****
****
***
**
*
```

Sterne 4 (114)

- Lesen Sie eine Zahl num von der Tastatur ein
- Geben Sie folgende Form aus:

```
Geben Sie die Anzahl der * ein: 6
****** *
***** **
**** ***
*** ****
** *****
* ******
```

Tabelle (115)

- Lesen Sie eine Länge und Breite einer Tabelle ein
- Geben Sie folgende Form aus:

```
Geben Sie die Laenge und die Breite ein: 6.3
   |A|B|C|D|E|F|
 ---------------
  1| | | | | | |
  2| | | | | | |
  3| | | | | | |
 ---------------
```

LowerUpperCase (004)

- Ausgabe: "Geben Sie etwas ein (mit <ENTER> koennen Sie die Eingabe beenden): "
- Lesen Sie ein Zeichen von der Tastatur ein unterdrücken Sie die Ausgabe.
- Geben Sie eingegebene Kleinbuchstaben als Großbuchstaben aus, eingegebene Großbuchstaben als Kleinbuchstaben und Ziffern gestürzt:
  0->9, 1->8 ... 8->1, 9->0. Sämtliche andere Zeichen sollen als Stern ausgegeben werden.
- Wiederholen Sie den Vorgang bis die Eingabetaste gedrückt wird.

Hilfe:

- mit `int getch()` kann ein Zeichen ohne Ausgabe von der Tastatur eingelesen werden.
- mit `'A'-'a'` kann der Abstand zwischen dem Zeichen Groß-A und Klein-A in der ASCII Tabelle ermittelt werden.
- Ausführungs-Beispiel:

```
Geben Sie etwas ein (mit <ENTER> koennen Sie die Eingabe beenden): hALLO*jAMES*bOND*992
```

Kommentare (062)

- In einer c-Quellcode Datei sollen die Kommentare /* .... */ gelöscht werden.
- Dazu zeichenweise von der Tastatur einlesen und ausgeben.
- Wenn ein /*-Tag kommt, dann wird begonnen nicht auszugeben.
- Wenn ein */- Tag kommt, dann wird aufgehört die Ausgabe zu unterdrücken.
- Bei der Ausführung des Programms wird die Eingabe auf eine c-Quelldatei umgeleitet und die Ausgabe in eine Zweite

Atoi (010, 063)

- Es gibt die Standard-Funktion atoi welche eine numerische Zeichenkette in eine Nummer umwandelt. Schreiben Sie diese Funktion.
- Ausgabe des doppelten eingegebenen Werts

```
int z = 0;
int num = 0;

printf("Bitte geben Sie eine Zahl ein: ");
while (z != '\n') {
  z = fgetc(stdin);  // Einlesen eines Zeichens nach dem Anderen

  // Einbau der Umrechnungslogik

}

  printf("\nDie doppelte Eingabe entspricht %d\n\n", num*2);
```

- Ausführungs-Beispiel:

```
Bitte geben Sie eine Zahl ein: 1234
Die doppelte Eingabe entspricht 2468
```

Hex2dec (064)

- Es gibt die Standard-Funktion hex2dec welche eine hexadezimale Zeichenkette in eine dezimale Nummer umwandelt. Schreiben Sie diese Funktion.

```
int z = 0;
int num = 0;

printf("Bitte geben Sie eine hexadezimale Zahl ein: ");
while (z != '\n') {
  z = fgetc(stdin);


  // Einbau der Umrechnungslogik


}
printf("\nDie eingegebene hexadezimale Zahl entspricht dezimal \"%d\"\n\n", num);
```

- Ausführungs-Beispiel:

```
Bitte geben Sie eine hexadezimale Zahl ein: AB34
Die eingegebene hexadezimale Zahl entspricht dezimal "43828"
```