

Programmieren in C – Wintersemester 2019 / 20

Aufgabenblatt 3

Abgabe: Montag, 25.11.2019, 08:30 Uhr (schriftlich und elektronisch)

Aufgabe 1 (Datentypen, Variablen und Wertebereiche):

- a) Markieren Sie alle „fehlerhaften“ Zeilen in folgendem Rumpfprogramm (der Begriff „Fehler“ umfaßt hier Compilerfehler, ggf. Compilerwarnungen und vor allem inhaltliche Fehler, die nicht notwendig zu Compilerfehlern oder Compilerwarnungen führen). Begründen Sie, warum es sich jeweils um einen Fehler handelt, und geben Sie mögliche Verbesserungen an. Bringen Sie dann das ursprüngliche (unverbesserte) Programm zum Laufen, indem Sie alle Zeilen, die zu Compilerfehlern führen, auskommentieren.

```
#include <stdio.h>
#include <limits.h>

int main(void)
{
    int i = 3.5;
    char Buchstabe = 280;
    double flaeche = 1.23e+5;
    short Max_X_Achse = 40000;
    unsigned short Max_Y_Achse = 50000;
    int 5k = 7;
    float umfang = 12345.6470589;
    long lzaehler = 3e9;
    long long llzaehler = 3e9;

    printf("i = %d\n", i);
    printf("Buchstabe = %d\n", Buchstabe);
    printf("flaeche = %d\n", flaeche);
    printf("Max_X_Achse = %hd\n", Max_X_Achse);
    printf("Max_Y_Achse = %hu\n", Max_Y_Achse);
    printf("5k = %d\n", 5k);
    printf("umfang = %f\n", umfang);
    printf("lzaehler = %ld\n", lzaehler);
    printf("long min = %ld, long max = %ld\n", LONG_MIN, LONG_MAX);
    printf("llzaehler = %lld\n", llzaehler);
    printf("sizeof(long): %f Byte\n", sizeof(long) );

    return 0;
}
```

- b) Schreiben Sie ein Programm, das ein einzelnes Zeichen abfragt und dann den zugehörigen ASCII-Code ausgibt.

Beispiel-Programmlauf:

```
Bitte geben Sie ein Zeichen ein:  
z  
Der zugehoerige ASCII-Code lautet: 122
```

Die Lösung soll das Programm und die Programmausgabe umfassen.

Aufgabe 2 (Einlesen von Zahlen, ausgeben in umgekehrter Reihenfolge):

Schreiben Sie ein Programm, das eine ganze Zahl mit genau 3 Ziffern als **int** (!) einliest und Ziffer für Ziffer von rechts nach links wieder ausgibt.

Beispiel für einen Programmlauf:

```
> ./Aufgabe2.exe  
Geben Sie eine ganze Zahl mit genau 3 Ziffern ein: 123  
Ausgabe der Ziffern:  
3  
2  
1
```

Hinweis: Verwenden Sie die ganzzahlige Division (/ und %) zur Lösung.

Die Lösung soll das Programm und die Programmausgabe umfassen.

Aufgabe 3 (Umrechnung und Ausgabe von Zeitangaben):

Schreiben Sie ein Programm, das ausgibt, wie viele Jahre, Wochen, Tage, Stunden, Minuten, Sekunden seit dem 01.01.1970 vergangen sind. Die Details der Ausgabe sind dem Beispiel unten zu entnehmen.

Zur Lösung dieser Aufgabe ist die Verwendung der Funktion `time()` sinnvoll, die im folgenden Rahmenprogramm verwendet wird, um die Anzahl Sekunden seit dem 01.01.1970 auszugeben.

```
#include <stdio.h>
/* #include <ctime> */ /* fuer C++ Compiler (z.B. Visual Studio) */
#include <time.h> /* fuer C Compiler (z.B. Dev-C++, CodeBlocks) */

int main(void)
{
    int zeitInSekunden = time(0); /* Zahl der Sekunden
                                   seit dem 01.01.1970 */

    printf("Seit dem 01.01.1970 sind %d Sekunden vergangen.\n",
           zeitInSekunden);

    return 0;
}
```

Die geforderte Ausgabe Ihres Programms soll nach Erweiterung des Rahmenprogramms wie folgt aussehen:

```
Seit dem 01.01.1970 sind vergangen:
1323805749 Sekunden
Das sind          22063429 Minuten
oder              367723 Stunden
oder              15321 Tage
oder              2188 Wochen
oder              41 Jahre
```

```
Seit dem 01.01.1970 sind vergangen:
 41 Jahre,  356 Tage,  19 Stunden,  49 Minuten und   9 Sekunden
```

Wie immer, soll die Lösung das Programm selbst und die Programmausgabe umfassen.

Aufgabe 4 (Ein- und Ausgabe, Datentypen, arithmetische Operatoren):

Schreiben Sie ein Programm, das Ihnen eine „Reisebilanz“ erstellt, d.h. die Durchschnittsgeschwindigkeit, Gesamtkosten, gefahrenen Kilometer und benötigte Zeit für eine Auto- und eine Bahnreise erstellt. Dabei sollen zunächst die gefahrenen km (Datentyp **double**), der Benzinverbrauch (Datentyp **double**) und die benötigte Zeit für den mit dem Auto zurückgelegten Teil der Reise vom Benutzer abgefragt werden. Die benötigte Zeit soll dabei im Format (std.min) eingegeben werden, wobei die Anzahl der Stunden (std) und der Minuten (min) jeweils **int**-Variablen sind. Das Programm errechnet aus diesen Angaben zunächst den Benzinverbrauch pro 100 km, die Durchschnittsgeschwindigkeit (in km/h) und die Benzinkosten (in Euro) aus (nehmen Sie hierzu einen Benzinpreis von 1.55€ pro Liter an). Anschließend werden die gefahrenen Kilometer (Datentyp **double**), der Preis für die Fahrkarte in Euro (Datentyp **double**) und die benötigte Zeit (in demselben Format wie oben) vom Benutzer abgefragt. Daraus berechnet das Programm dann die Durchschnittsgeschwindigkeit der Bahn (in km/h). Anschließend berechnet das Programm die Durchschnittsgeschwindigkeit, Gesamtkosten, gefahrenen Kilometer und benötigte Zeit für die Gesamtstrecke aus den beiden Reiseabschnitten und gibt alle berechneten Werte auf dem Bildschirm aus.

Deklarieren Sie alle Variablen **const**, für die es möglich ist.

Bemerkung:

Da wir noch keine if-Abfrage kennen, können Sie zur Vereinfachung davon ausgehen, daß sämtliche Benutzereingaben größer als 0 sind, so daß nicht durch 0 dividiert wird (dies sollte bei der Eingabe dann auch beachtet werden...)

Beispiel für einen Programmlauf:

```
Wieviele km sind Sie mit dem Auto gefahren?
100
Benzinverbrauch in Litern?
8
Benötigte Zeit (Format: std.min)?
0.30
```

```
Wieviele km sind Sie mit der Bahn gefahren?
200
Preis fuer die Fahrkarte in Euro?
25
Benötigte Zeit (Format: std.min)?
2.0
```

```
Reisebilanz:
=====
```

```
Auto:
-----
```

```
Verbrauch: 8.00 l/100km
Durchschnittsgeschwindigkeit: 200.00 km/h
Preis: 12.40 Euro
```

Bahn:

Durchschnittsgeschwindigkeit: 100.00 km/h

Gesamt:

Gefahrene Kilometer: 300.000000

Benötigte Zeit: 2 Stunden, 30 Minuten

Durchschnittsgeschwindigkeit: 120.00 km/h

Gesamtkosten: 37.40 Euro