

Programmieren in C (Übungsblatt 1)

Aufgabe 1: Programmausführung

a) Bitte geben Sie den folgenden Programmcode ein:

```
/* Beispiel fuer Textausgaben */
#include <stdio.h>
int main()
{
    printf("C-Programmierung ist ganz einfach !\n");
    printf("Programm erfolgreich beendet. \n");
    return 0;
}
```

b) Speichern Sie den Programmcode unter dem Namen `bsp1.c` ab.

c) Übersetzen Sie das Programm mit: `gcc -o bsp1.exe bsp1.c`

d) Starten Sie das Programm mit: `./bsp1.exe`

Aufgabe 2: Ein- und Ausgabe von Variablen

Ändern Sie das Programm aus Aufgabe 1 so ab, dass 1 Integerzahl und eine Gleitkommazahl von der Tastatur eingelesen und anschließend wieder auf dem Bildschirm ausgegeben werden.

Vorgehensweise:

- 2 Integerwerte (xwert und ywert) deklarieren
- Variablen mit 0 bzw. 0.00 initialisieren
- die beiden Zahlenwerte mit der scanf-Funktion einlesen
- Ausgabe der Zahlenwerte mit der printf-Funktion

Übersetzen Sie das Programm und führen Sie es anschließend aus.

Aufgabe 3: Mathematische Operatoren I

Schreiben Sie ein Programm, das zwei Integerzahlen einliest und die folgenden Informationen auf dem Bildschirm ausgibt:

- Summe der beiden Zahlen
- Produkt der beiden Zahlen
- Division (mit und ohne Typenumwandlung)
- Modulo-Operator

Aufgabe 4: Mathematische Operatoren II

Schreiben Sie ein Programm, das die Oberfläche eines Quaders berechnet. Die Formelberechnung für die Oberflächenberechnung lautet: $A = 2(a*b + a*c + b*c)$

Aufgabe 5: Mathematische Operatoren III (Zusatzaufgabe)

Schreiben Sie ein Programm, das eine Sekundenanzahl von der Tastatur einliest und daraus die Anzahl der Stunden, Minuten und Sekunden berechnet.:

Beispielausgabe: Die eingegebenen 3665 Sekunden entsprechen 1 Stunde, 1 Minute und 5 Sekunden.

Aufgabe 6: Mathematische Funktionen I

Schreiben Sie ein Programm, das die Diagonale in einem Rechteck berechnet. Die Rechteckseiten sollen jeweils 10 und 15 Meter lang sein.

Aufgabe 7: Mathematische Funktionen II

Schreiben Sie ein Programm, das die n-te Wurzel aus einer Zahl berechnet.

Beispiel: $2^3=8$ $\sqrt[3]{8}=2$

Berechnung in C: $2=8^{(1/3)}$ ---> $2 = \text{pow}(8,1/3)$

Zur Berechnung benötigen Sie die Potenzfunktion pow aus der Standardbibliothek math.h.

Aufgabe 8: if-Bedingung

Schreiben Sie ein Programm, das eine Integer-Zahl von der Tastatur einliest und diese Zahl nur wieder auf dem Bildschirm ausgibt, wenn eine gerade Zahl eingegeben wurde.

Aufgabe 9: if-Bedingung

Ändern Sie das obige Programm so ab, dass zwei Integer-Zahlen eingelesen werden und anschließend folgende Mitteilungen ausgegeben werden können:

- Sie haben zwei gerade Zahlen eingegeben.
- Sie haben zwei ungerade Zahlen eingegeben .
- Sie haben eine gerade und eine ungerade Zahl eingegeben.

Aufgabe 10: Schleifen

Schreiben Sie jeweils ein Programm mit einer Schleife, das in 3er-Schritten von 1 bis 100 zählt und den jeweiligen Zahlenwert ausgibt.

- a) Programmcode mit einer for-Schleife
- b) Programmcode mit einer while-Anweisung
- c) Programmcode mit einer do-while-Anweisung

Aufgabe 11: verschachtelte Schleifen

Schreiben Sie ein Programm, das das folgende "Baummuster" auf dem Bildschirm ausgibt:

```
*
***
*****
*****
*****
*****
*****
*****
*****
```

Aufgabe 12: switch-Anweisung I

Schreiben Sie ein Programm, das Sie auffordert eine Zahl zwischen 1 und 5 einzugeben. Je nach eingegebener Zahl soll der folgende Satz ausgegeben werden:

- Mit 1 haben Sie eine gute Wahl getroffen.
- Der Zahlenwert 2 ist auch nicht schlecht.
- Sie haben die Zahl 3 eingegeben.
- 4 ist eine gerade Zahl.
- Sie haben mit 5 wieder die höchst mögliche Zahl eingegeben.

Aufgabe 13a: switch-Anweisung II

Geben Sie den folgenden Programmcode ein und überprüfen Sie ob das Programm korrekt arbeitet. Falls nötig ändern Sie den Code für ein fehlerfreies Arbeiten ab.

```
#include <stdio.h>
int main()
{ int eingabe;
  printf("Bitte geben Sie die Zahlen 1 oder 2 ein \n");
  scanf ("%d",&eingabe);
  switch(eingabe)
  {
    default:
      printf("Sie haben weder 1 noch 2 eingegeben \n");
    case 1:
      printf("Sie haben die Zahl 1 eingegeben \n");
      break;
    case 2:
      printf("Sie haben die Zahl 2 eingegeben \n");
      break;}
  return 0;
}
```

Aufgabe 13b: switch- und if-else Anweisungen

Bilden Sie die switch-Anweisung aus Übung 13a mit if-Anweisungen nach.

Aufgabe 14: continue-Anweisung

Schreiben Sie zwei Programme, mit denen Sie die Summe aller geraden Zahlen von 2 bis 20 berechnen:

- a) ohne Verwendung einer continue-Anweisung
- b) mit Verwendung einer continue-Anweisung

Aufgabe 15: (Zusatzaufgabe)

Schreiben Sie ein Programm, das folgende Berechnungen durchführen kann:

- Fläche eines Kreises berechnen ($A = \pi \cdot r^2$)
- Umfang eines Kreises berechnen ($U = 2 \cdot \pi \cdot r$)
- Fläche eines Rechtecks berechnen ($A = \text{hoehe} \cdot \text{breite}$)
- Umfang eines Rechtecks berechnen ($U = 2 \cdot (\text{hoehe} + \text{breite})$)

($\pi = 3.141592$)

Der Benutzer soll bei jedem Durchlauf auswählen können, welche Berechnung er gerne durchführen möchte.

Aufgabe 16: (Zusatzaufgabe)

Schreiben Sie ein Programm, bei dem der Benutzer eine zufällig vom Computer ausgewählte Zahl erraten soll. Der Benutzer soll 5 Versuche haben um die Zahl zu erraten. Falls der Benutzer die richtige Zahl erraten hat, soll sowohl die Zahl als auch die Anzahl der benötigten Versuche ausgegeben werden. Falls die eingegebene Zahl nicht richtig war, wird vom Programm eine Meldung ausgegeben, ob die zu erratende Zahl kleiner oder größer als die eingegebene Zahl ist.

Nach 5 falschen Versuchen soll das Programm mit einer entsprechenden Meldung abbrechen.

Anmerkung:

Zur Erzeugung von Zufallszahlen benötigen Sie die Bibliotheksfunktion `rand() % [maxzahl]` aus der Standardbibliothek `<stdlib.h>`. Ihr Programmcode muss also die Zeile `#include <stdlib.h>` enthalten. Mit der folgenden Zeile erzeugen Sie eine Zufallszahl zwischen 0 und 100:

```
zufall=rand()%100;
```

Damit der Zufallszahlengenerator nicht immer die gleichen Zahlen ermittelt, muss zunächst der Zufallszahlengenerator über die Bibliotheksfunktion `srand((time(0)))` mit der Systemzeit initialisiert werden. Beachten Sie, dass Sie für die Funktion `time(0)` die Headerdatei `<time.h>` einbinden müssen.

Benötigte Include-Dateien:

```
#include <stdlib.h>
```

```
#include <time.h>
```

Ermittlung einer Zufallszahl zwischen 0 und 100:

```
srand(time(0));
```

```
zufallszahl = rand()%100;
```