

Einführung in die Programmierung, WS 2015/2016 Blatt 5

Simon Hangl, Sebastian Stabinger, Benedikt Hupfaut, Johannes Kessler

2015–11–11

- Abgabe bis spätestens Dienstag 21:59:59 über OLAT (<https://lms.uibk.ac.at/olat/dmz/>).
- Bereiten Sie jede Aufgabe so vor, dass Sie Ihre Lösung im Proseminar präsentieren können!
- Benennen Sie Ihre Abgabe nach folgendem Schema:
Gruppennummer-Nachname-blattÜbungsblattnummer.tar.gz Wenn Sie also Max Mustermann heißen und Gruppe 1 besuchen, heißt die Datei von Übung 5: *1-mustermann-blatt5.tar.gz*
- Compilieren Sie alle Programme mit den Optionen `-Wall -Werror -std=c99`

Feedback

Nutzen Sie die angebotenen Möglichkeiten, uns Feedback zu geben (eMail, Tutorium, Proseminar). Hier können Sie uns auf Probleme, notwendige Stoffwiederholungen, Unklarheiten, aber auch positive Dinge, die beibehalten werden sollten, hinweisen.

Testen und Dokumentation

Stellen Sie sicher, dass alle Lösungen fehlerfrei kompilieren. Testen Sie Ihre Lösungen ausführlich (z.B. auf falsche Eingaben, falsche Berechnungen, Sonderfälle) und dokumentieren Sie sie. Dies hilft Ihnen bei der Präsentation und uns beim Nachvollziehen Ihrer Entscheidungen. Lesen Sie die Aufgaben *vollständig* durch.

Aufgabe 1 (3 Punkte)

Schreiben Sie ein Programm welches die Zahlen von 1 bis j ausgibt wobei j von 1 bis n und wieder zurück zu 1 variieren soll. Für $n = 5$ soll die Ausgabe des Programms folgendermaßen aussehen.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Hinweis: Verwenden Sie verschachtelte Schleifen

Hinweis: Abgabe: 1-mustermann-a1.c

Aufgabe 2 (3 Punkte)

Schreiben Sie ein Programm, das alle Tripel (x , y , z) ausgibt, für die die folgende Bedingungen gelten:

- Die Variablen x , y und z sind Ganzzahlen zwischen 0 und n
- Keine Zahl darf mehr als ein mal im Tripel vorkommen.

Aufgabe 3 (4 Punkte)

Folgendes Programm ist gegeben:

```
#include <stdio.h>

int main(void) {
    float a = 1.2;
    double x = sizeof(a) * 2 && 23 / printf("abc\n") ^ 4 + 1;
    double y = ((int)a / 2 + (7 | 8) << 4);
    double z = a * 6;
    double u = 0xBADCAB1E ? 1 << 1 : 0xDEADCODE;
    double v = (u + z * 10) / 8;
```

```

double w = (int)(u + z * 10) / 8;

printf("x=%lf\n", x);
printf("y=%lf\n", y);
printf("z=%lf\n", z);
printf("u=%lf\n", u);
printf("v=%lf\n", v);
printf("w=%lf\n", w);
}

```

Erklären Sie wie es zu den jeweiligen Endwerten der Variablen kommt und beantworten Sie zusätzlich die folgende Fragen:

- wie werden boolsche-Werte in C repräsentiert?
- welche Werte entsprechen dem boolschen Wert *true*?
- welche Werte entsprechen dem boolschen Wert *false*?
- welche expliziten Typecasts wurden verwendet und welchen Einfluss haben sie auf das Ergebnis?
- Finden implizite Typecasts statt und wenn ja wo?

Hinweis: Abgabe: 1-mustermann-a3.txt