

Deckblatt: Übung zur Vorlesung Informatik 1

Fakultät für Angewandte Informatik

Lehrprofessur für Informatik

PROF. DR. LORENZ, MARIUS BRENDLE, JOHANNES METZGER, LEV SOROKIN

WS 2017/18

Hinweis: Es sind alle Felder auszufüllen! Abgabe der Übungsblätter immer **mittwochs** (Ausnahme wenn Feiertag: donnerstags) bis **spätestens 12:00 Uhr** in die entsprechend gekennzeichneten Briefkästen der Veranstaltung im Erdgeschoss des Instituts für Informatik (Gebäude N). Zuwiderhandlung wird mit Strafe geahndet! (Punktabzug)

Übungsblatt	
-------------	--

(hier die Nummer des bearbeiteten **Übungsblatts** eintragen)

	Übung 01 (1055 N) Montag 08:15 - 09:45 Uhr (Lennart Eing)
	Übung 02 (1056 N) Montag 08:15 - 09:45 Uhr (Alexander Fuchs)
	Übung 03 (1057 N) Montag 08:15 - 09:45 Uhr (Michelle Lienhart)
	Übung 04 (1055 N) Montag 12:15 - 13:45 Uhr (Henning Cui)
	Übung 05 (1056 N) Montag 12:15 - 13:45 Uhr (Christian Schavitz)
	Übung 06 (1055 N) Montag 14:00 - 15:30 Uhr (Maximilian Demmler)
	Übung 08 (1056 N) Montag 17:30 - 19:00 Uhr (Moritz Feldmann)
	Übung 09 (1057 N) Montag 17:30 - 19:00 Uhr (Dat Le Thanh)
	Übung 10 (1057 N) Dienstag 12:15 - 13:45 Uhr (Alexander Szöke)
	Übung 11 (1057 N) Dienstag 14:00 - 15:30 Uhr (Denise Böhm)
	Übung 12 (1056 N) Dienstag 17:30 - 19:00 Uhr (Marvin Drexelius)
	Übung 13 (1057 N) Dienstag 17:30 - 19:00 Uhr (Tom Wolfskämpf)
	Übung 14 (1055 N) Mittwoch 08:15 - 09:45 Uhr (Jonas Junge)
	Übung 15 (1055 N) Mittwoch 10:00 - 11:30 Uhr (Elisabeth Korndörfer)
X	Übung 16 (1054 N) Donnerstag 14:00 - 15:30 Uhr (Florian Magg)
	Übung 17 (1057 N) Donnerstag 14:00 - 15:30 Uhr (Lukas Lodes)
	Übung 18 (1054 N)* Donnerstag 17:30 - 19:00 Uhr (Patrick Eckert)
	Übung 19 (1058 N) Freitag 08:15 - 09:45 Uhr (Lena Tikovsky)
	Übung 20 (1054 N) Freitag 10:00 - 11:30 Uhr (Felix Fischer)
	Übung 21 (1055 N)* Freitag 14:00 - 15:30 Uhr (Isabell Rücker)
	Übung 23 (1057 N) Freitag 15:45 - 17:15 Uhr (André Schweiger)

(hier die eingeteilte **Übungsgruppe** ankreuzen)

*(1056 N bis 03.11.17)

Teamnummer	5
------------	----------

(hier die Nummer des eingeteilten **Teams** eintragen)

Benjamin Ritter
Marina Huber
Anton Lydike

(hier die **Vor- und Nachnamen** aller Teammitglieder eintragen)

Aufgabe		
Aufgabe		
Aufgabe		
Aufgabe		
Gesamt		

(vom Tutor auszufüllen)

Assignment 7

25)

a)

1. `#define FALSE 0`
2. `#define mittelwert(a, b) (((a) + (b))/2)`
3. `#define print_terminated() printf("Program terminated")`
4. `#define toupper(c) (((c) >= 'a' && (c) <= 'z') ? (c) + 32 : (c))`

b)

1. `gcc -c a.c`
2. `gcc -E a.c` führt nur den Präprozessor aus und wandelt den Quellcode in erweiterten Quellcode um.
3. `gcc a.o b.o c.o -o c.out` (evtl wird noch eine `main.c` datei benötigt, also `gcc a.o b.o c.o main.c -o c.out`)

c)

1. `int endswith(char *string, char end);`
2. `double average(int field[], int length);`

d)

```
/* 25/d.h */
#ifndef GETSIGN_INCLUDED
#define GETSIGN_INCLUDED

#define getsign(num) ((num) >= 0 ? 1 : -1)

#endif
```

26)**a)**

```
#include <stdio.h>

int flush_buff(void);

int read_latin_lowercase(void);

/*
 * tries to read a sinngle lowercase latin character
 * returns:
 *   'a' - 'z': if a valid character was entered
 *   -1: if a invalid  character entered
 *   -2: buffer error (EOF encountered)
 */
int read_latin_lowercase(void)
{
    int c = getchar(),
        d,
        flush_successful;

    if (c == '\n') {
        /* if no character was supplied */
        return -1;
    } else if (c == EOF) {
        /* buffer error in first char */
        return -2;
    } else if ((d = getchar()) == '\n' && 'a' <= c && c <= 'z') {
        /* valid input */
        return c;
    } if (d == EOF) {
        /* buffer error in second char */
        return -2;
    } else {
        flush_successful = flush_buff();
        return flush_successful ? -1 : -2;
    }
}

int flush_buff(void)
{
    int c;
    while ((c = getchar()) != '\n' && c != EOF) {}
    return c != EOF;
}
```

b)

```
#include <stdio.h>

int flush_buff(void);

double read_probability(void);

/*
 * tries to read a probability between 0 and 1 (inclusive)
 * returns:
 *   number in [0 - 1]: if a valid character was entered
 *   -1: if a invalid input was entered
 *   -2: buffer error (EOF encountered)
 */
double read_probability(void)
{
    double read;

    int c = scanf("%lf", &read),
        d = getchar(),
        flush_successful;

    if (d != '\n' || c == 0) {
        /* more input - flush */
        flush_successful = flush_buff();
        return flush_successful ? -1 : -2;
    } else if (read < 0 || read > 1) {
        return -1;
    } else if (d == EOF) {
        /* buffer error */
        return -2;
    }

    return read;
}

int flush_buff(void)
{
    int c;
    while ((c = getchar()) != '\n' && c != EOF) {}
    return c != EOF;
}
```