

**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)
<b>X</b>	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	<b>5</b>
------------	----------

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

<b>Benjamin Ritter</b>
<b>Marina Huber</b>
<b>Anton Lydike</b>

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

Aufgabe		
Aufgabe		
Aufgabe		
Aufgabe		
<b>Gesamt</b>		

(vom Tutor auszufüllen)

# Assignment 6

---

21)

---

a)

1. `static double a = 1.0;`
2. `double v[N];`
3. `s[1] = getchar();`
4. `int i; for (i = L; i >= 0; i--) { a[i] = 0; } (oder int a[L] = { 0 }; oder a = { 0 } )`
5. `strcmp(s, "abc") == 0`
6. `strlen(s) < 10`
7. `strcpy(s, "abc");`
8. `strncat(t, s, 2);`
9. `0`
10. `3`

b)

1. `9`
2. `10`
3. `9`
4. `10`
5. `8`
6. `10`
7. `8`

**22)****a)**

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

char *my_strcat(char* dest, char* src)
{
    int src_len = strlen(src),
        dest_len = strlen(dest),
        i;

    for(i = 0; i < src_len; i++) {
        dest[dest_len + i] = src[i];
    }

    dest[dest_len + src_len] = '\0';

    return dest;
}

int main (void)
{
    char *dest = malloc(12 * sizeof(char));

    my_strcat(dest, "hello ");
    my_strcat(dest, "world");

    printf("%s\n", dest);

    return 0;
}
```

**b)**

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

char *my_strncpy(char* dest, char* src, int size)
{
    int i,
        srclen = strlen(src);

    for(i = 0; i < size; i++) {
        if (i < srclen) {
            dest[i] = src[i];
        } else {
            dest[i] = '\0';
        }
    }

    return dest;
}

int main (void)
{
    char *dest = malloc(12 * sizeof(char));

    my_strncpy(dest, "hello world", 12);

    printf("%s\n", dest);

    my_strncpy(dest, "hello dude", 12);

    printf("%s\n", dest);

    return 0;
}
```

c)

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

/* zweite funktion, um einrückungstiefe zu beschränken */
unsigned int charcount (char needle, char* haystack, int hlen)
{
    int i, r = 0;

    for (i = 0; i < hlen; i++) {
        if (haystack[i] == needle) r++;
    }

    return r;
}

unsigned int strcount(char* needle, char* haystack)
{
    int needle_len = strlen(needle),
        haystack_size = strlen(haystack),
        r = 0,
        i;

    for(i = 0; i < needle_len; i++) {
        r += charcount(needle[i], haystack, haystack_size);
    }

    return r;
}

int main (void)
{
    printf("%i\n", strcount("HALLO", "LAGER"));

    return 0;
}
```

d)

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

int isprefix(char* source, char* prefix)
{
    int srclen = strlen(source),
        prefixlen = strlen(prefix),
        i;

    if (prefixlen > srclen) {
        return 0;
    }

    for(i = 0; i < prefixlen; i++) {
        if (source[i] != prefix[i]) return 0;
    }

    return 1;
}

int main (void)
{
    printf("%i\n", isprefix("hallo", "welt"));
    printf("%i\n", isprefix("hallo", "ha"));
    printf("%i\n", isprefix("hallo", "hallo welt"));
    printf("%i\n", isprefix("hallo", "nein"));
    return 0;
}
```

**23)****a)**

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

int lotto()
{
    return (rand() % 49) + 1;
}

int main (void)
{
    time_t t;

    int random = 0,
        had_one = 0,
        had_49 = 0;

    srand((unsigned) time(&t));

    while (!had_one || !had_49) {
        random = lotto();

        if (random == 1) had_one = 1;

        if (random == 49) had_49 = 1;
    }

    printf("had 1 and 49, am done.\n");

    return 0;
}
```

**b)**

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

char* colors[] = { "red", "blue", "yellow" };

char* rand_color()
{
    return colors[rand() % 3];
}

int main (void)
{
    time_t t;

    srand((unsigned) time(&t));

    printf("%s\n", rand_color());

    return 0;
}
```

c)

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

char rand_char()
{
    /* 122 - 33 + 1 = 90 */
    int num = (rand() % 90) + 33;

    if (num > '9' && num < 'A') {
        return rand_char();
    } else if (num > 'Z' && num < 'a') {
        return rand_char();
    } else {
        return num;
    }
}

int main (void)
{
    time_t t;

    srand((unsigned) time(&t));

    printf("%c\n", rand_char());

    return 0;
}
```



## 24)

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_INPUT_LENGTH 10

int read_num(char input[]) {
    int inlen = 0;
    char read = getchar();

    if (read == '\n') return 0;

    if (!isdigit(read)) {
        while (getchar() != '\n') {};
        return 0;
    }

    if ((read == '0' && getchar() == '\n')) {
        input[0] = read;
        return 1;
    } else if (read == '0'){
        return 0;
    }

    input[0] = read;
    inlen++;

    while(inlen <= MAX_INPUT_LENGTH) {
        read = getchar();
        if (read == '\n') {
            return 1;
        } else if (isdigit(read)) {
            input[inlen] = read;
            inlen++;
        } else {
            while (getchar() != '\n') {};
            return 0;
        }
    }

    while (getchar() != '\n') {};

    return 0;
}

int main (void)
{
    char result[MAX_INPUT_LENGTH + 1] = { 0 };

    if (read_num(result)) {
        printf("%s\n", result);
    } else {
        printf("readerror!\n");
    }

    return 0;
}
```