

### Aufgabe 1

Das folgende Programm ist inkorrekt. Der Versuch es auf unserem Linux-System auszuführen, resultiert in einem Speicherzugriffsfehler. Auf anderen Systemen, wird häufig folgendes ausgegeben:

```
abcSonne  onne  abcSonne
```

Diese Ausgabe ist durchaus verständlich und sagt etwas über den verwendeten Compiler aus. Erklären Sie was passiert und korrigieren Sie das Programm.

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

int main( void )
{
    char *p1 = "abc", *p2 = "Sonne";

    printf("%s %s %s\n", p1, p2, strcat(p1,p2));
    exit(EXIT_SUCCESS);
}
```

### Aufgabe 2

Schreiben Sie eine Funktion **replace**, die in einer Zeichenkette *string* das Zeichen *oldch* durch das Zeichen *newch* ersetzt. *oldch* soll an **allen** Positionen, wo es in *string* vorkommt, geändert werden. Die Funktion soll einen Zeiger auf *string* zurückgeben.

### Aufgabe 3

Schreiben Sie eine Funktion **swap**, die zwei Zeiger auf Zeichenketten vertauscht. Zum Testen wird Ihnen das Hauptprogramm `swapzeiger_main.c` zur Verfügung gestellt.

### Aufgabe 4

Schreiben Sie eine Funktion `strip()`, die Leerzeichen in dem als Argument übergebenen String entfernt. Ob Leerzeichen am Anfang oder/und am Ende der Zeichenkette abgeschnitten werden sollen, soll durch Übergabe eines Wertes vom Typ

```
enum strip_type {LEADING = 1,
                  TRAILING = 2,
                  BOTH = LEADING+TRAILING
                };
```

angegeben werden können. (Von anderen Zeichen umgebende Leerzeichen sollen nicht entfernt werden.)

Die Funktion soll als Zeigerversion implementiert werden(, wobei kein Indexoperator und auch keine Variable mit einem ganzzahligen Indexwert benutzt werden sollte).

**Aufgabe 5**

Die Funktion `genswap()` soll zwei beliebige Datenobjekte mit der Größe  $n$  Bytes miteinander vertauschen. Sollen beispielsweise zwei **double**-Variablen vertauscht werden, so ist beim Aufruf für  $n$  **sizeof(double)** zu übergeben.

```
double    a=2.0,
          b=1.0;
char      c[MAX_LEN]="Mond",
          d[MAX_LEN]="xyz";
short int e=-1,
          f=2;

genswap(&a, &b, sizeof(double));

=> a == 1.0
   b == 2.0

...

genswap(&e, &f, sizeof(short int));

=> e == 2
   f == -1
```

**Aufgabe 6** (Zusatzaufgabe, Bearbeitung freiwillig)

Implementieren Sie eine eigene Version der C-Bibliotheksfunktion **memmove**. Ihre Funktion sollte mit Hilfe von Zeigern (anstelle von Vektorindizes) formuliert werden und sollte keinen zusätzlichen Zwischenspeicher für das Kopieren verwenden.