

## C/C++-Kurs — Übungsblatt 2

### Aufgabe 1: Strcmp()

Implementieren sie `Strcmp()` gemäß §7.21.4.2 (The `strcmp` function) der C99-Norm.

### Aufgabe 2: void\* Reloaded

Sortieren Sie die Reihung mit Elementtyp `char const*`, welche mit den Werten

```
"Neo"  
"Morpheus"  
"Trinity"  
"Smith"
```

initialisiert ist, absteigend mittels der Funktion `qsort()`, welche in `<stdlib.h>` zu finden ist. Dokumentation zu `qsort()` finden Sie u.a. in der C99-Norm im Abschnitt §7.20.5.2. Mit `printf("%s", x)`; können Sie jeweils einen `char const*` ausgeben.

### Aufgabe 3: Printf()

Erweitern Sie die in der Vorlesung (noch nicht) begonnene `printf()`-Reimplementierung. Die genauen Wirkungen der einzelnen Angaben sind in der C99-Norm in §7.19.6.1 beschrieben.

1. Machen Sie die feste Puffergröße zur Ausgabe von Ganzzahlen portabel. Notwendige Information hierzu finden Sie in §7.10 der C99-Norm.
2. Hexadezimalausgabe: `%X` und `%x` veranlassen Ausgabe des übergebenen `unsigned`-Wertes als Hexadezimalzahl. Entsprechend der Groß-/Kleinschreibung werden Groß-/Kleinbuchstaben verwendet.
3. Minimale Feldbreite: `%10u` gibt an, dass, falls die auszugebende Zahl weniger als 10 Ziffern benötigt, die Ausgabe *links* mit Leerzeichen aufgefüllt wird. In keinem Fall werden Teile der Ausgabe abgeschnitten. Selbiges gilt für andere Formattypen.
4. Führende Nullen: `%010u` gibt an, dass die Ziffer 0 statt Leerzeichen zum Auffüllen verwendet wird. Ist die Formatangabe kein Zahlenformat, so ist das Verhalten *undefiniert*.

5. Linksbündig: %-10u gibt an, dass der Wert linksbündig ausgegeben wird. Falls sowohl - und 0 verwendet wird, so zählt -.

Der generelle Aufbau ist %[Flags][Breite]Format. - und 0 sind Flags. Ihre Reihenfolge und die Häufigkeit ihres Auftretens ist unerheblich (%0-0-0-10u).

## Aufgabe 4: 0x80000000 != 2147483648?

Warum schlägt der erste Versuch *nur* Bit 31 zu löschen fehl? Wie kann man das Problem vermeiden und dennoch eine Hexadezimalkonstante, die zur Visualisierung von Bits besser geeignet ist als eine Dezimalkonstante, verwenden? Übersetzen Sie das Programm mittels `cc -std=c99`.

```
#include <stdio.h>

int main(void)
{
    unsigned long long x;

    x = 0xFFFFFFFFFFFFFFFF;
    x &= ~0x0000000080000000;
    printf("0x%016lX\n", x);

    x = 0xFFFFFFFFFFFFFFFF;
    x &= ~2147483648;
    printf("0x%016lX\n", x);

    printf(0x0000000080000000 == 2147483648 ? "gleich\n" : "ungleich\n");
}
```

## Aufgabe 5: Was tut die nächste Zeile??/

Übersetzen Sie das Programm mittels `cc -std=c99`.

```
#include <stdio.h>

int main(void)
{
    int x = 23;
    // Was tut die naechste Zeile??/
    x += 19;
    printf("%d\n", x);
}
```