

Prozedurale Programmierung (PPR)

Übungsblatt zu Kapitel 4 (Funktionen)

WS 2020/21

Aufgabe 4.1 (Typumwandlung)

Betrachten Sie das folgende Programm

```
1  int main(void)
2  {
3      unsigned char uc = 153;
4      char c          = uc;
5      unsigned int ui  = c;
6      int i           = ui;
7
8      printf("unsigned char (%d) : %d\n", uc);
9      printf("unsigned char (%u) : %u\n", uc);
10     printf("unsigned char (%x) : %x\n", uc);
11     printf("\n");
12
13     printf(" signed char (%d) : %d\n", c);
14     printf(" signed char (%u) : %u\n", c);
15     printf(" signed char (%x) : %x\n", c);
16     printf("\n");
17
18     printf("unsigned int  (%d) : %d\n", ui);
19     printf("unsigned int  (%u) : %u\n", ui);
20     printf("unsigned int  (%x) : %x\n", ui);
21     printf("\n");
22
23     printf(" signed int   (%d) : %d\n", i);
24     printf(" signed int   (%u) : %u\n", i);
25     printf(" signed int   (%x) : %x\n", i);
26     printf("\n");
27 }
```

das die folgenden Ausgaben erzeugt.

```
1  unsigned char (%d) : 153
2  unsigned char (%u) : 153
3  unsigned char (%x) : 99
4
5  signed char (%d) : -103
6  signed char (%u) : 4294967193
7  signed char (%x) : fffffff99
8
```

```

9  unsigned int    (%d) : -103
10 unsigned int    (%u) : 4294967193
11 unsigned int    (%x) : ffffffff99
12
13 signed int      (%d) : -103
14 signed int      (%u) : 4294967193
15 signed int      (%x) : ffffffff99

```

Erklären Sie, wie diese Ausgaben zustande kommen.

Aufgabe 4.2 (Wertebereiche der Datentypen)

Schreiben Sie zwei Programme zur Bestimmung der Wertebereiche von Variablen vom Typ **char**, **short**, **int** und **long**, sowohl für **signed** als auch **unsigned**,

- a) indem Sie die entsprechenden kleinsten und größten Werte aus der Standard-Definitionsdatei `limits.h` ausgeben.
- b) indem Sie dieselben Werte mit Hilfe von Bitoperatoren selbst bestimmen.

Aufgabe 4.3 (Speicherklassen)

Schreiben Sie das Programm `coffee_machine.c` so um, dass die Funktion `main` in einer eigenen Datei definiert wird, aber beide C-Dateien sich getrennt voneinander übersetzen lassen. Definieren Sie alle Elemente als **static**, die nur innerhalb einer der beiden Dateien benötigt werden.

Das Programm `coffee_machine.c` steht Ihnen bei den Code-Beispielen im Moodle-Kurs zum Download zur Verfügung.