

## Kontrollfragen A

1. Welcher Wert hat w?

```
int w = 034;
```

28 (Dezimal)      034 wird als Octal interpretiert → 34oct = 28dec

2. Wie lauten die Ergebnisse?

```
unsigned int a = 3, b = 5, c = 7;
```

```
unsigned int x, y, z;
```

```
x = a == 3;
```

```
y = (a << 1) & b;
```

```
z = (a > 3) ? b : c;
```

1      (a == 3) => true und dies in x speichern -> true = 1

4      (a << 1) a resp. 3 um 1 Bit nach links schieben = 0110bin = 6dec

6dec & (Bitweise AND) b resp. 5dec (0101bin) = 0100bin = 4dec

7      (expr) ? then : else

(a > 3) => false, else -> c = 7

## Kontrollfragen B

1. Folgender Code soll x auf 3 setzen, wenn a drei ist (sonst 4), und y auf 3 setzen, wenn a und b drei ist. Was macht dieser Code?

```
y = 4;
```

```
if (a == 3)
```

```
  x = 3;
```

```
  if (b == 3)
```

```
    y = 3;
```

```
else
```

```
  x = 4;
```

*Klammern fehlen. Der Compiler ignoriert das Einrückten.*

```
y = 4;
```

```
if (a == 3) {
```

```
  x = 3;
```

```
}
```

```
if (b == 3) {
```

```
  y = 3;
```

```
}
```

```
Else {
```

```
  x = 4;
```

```
}
```

## Kontrollfragen C

1. Was macht folgender Code?

```
int i = 0;
```

```
do
```

```
{
```

```
  if (i++ % 2)
```

```
    continue;
```

```
  printf("%d\n", i);
```

```
} while (i < 10);
```

*Beim ersten Durchgang ist i = 0:*

*(i++ % 2), zuerst wird (i % 2) überprüft und danach i++ ausgeführt → (0 % 2) = 0, 0 entspricht „false“ somit wird die IF Bedingung nicht erfüllt.*

*→ Ausgaben: 0\n*

*Beim zweiten Durchgang ist i = 1:*

*(1 % 2) != 0 somit If-Bedingung erfüllt-> Continue -> printf() wird übersprungen.*

*→ Keine Ausgabe*

*Dritten Durchgang ist i = 2, usw...*

*→ Ausgabe: 0, 2, 4, 6, 8*

## Kontrollfragen D

1. Erklären Sie folgender Code?

```
int i, limit = 100;
char c;
char s[100];
for (i = 0;
     i < limit-1 && (c = getchar()) != EOF && c != '\n'; i++)
{
    s[i] = c;
}
s[i] = '\0';
```

*Arraydeklaration für 100 char-Werte → Liest die Eingabe bis maximal 100 Zeichen, ein "ENTER" oder das Ende des Files kommt.*

2. Welche Ausgabe erhalten Sie mit folgender Anweisung?

```
float pi = 3.14156f;
printf("%s%f\n", "Pi=", pi);
```

*Pi=3.141560 (per Default werden 6 Stellen ausgegeben, das 0 am Ende wird hinzugefügt)*