

```
unsigned int one = 1;

if (one < -1)
    cout << "Üble Sache!";
else if (-1 < one)
    cout << "So geht das!";
else
    cout << "???";
```

Dieses Programm ... ?

- 1 Kompiliert nicht
- 2 Bricht mit einem Laufzeitfehler ab
- 3 Gibt "Üble Sache!" aus
- 4 Gibt "So geht das!" aus
- 5 Gibt "???" aus



```
unsigned int one = 1;

if (one < -1)
    cout << "Üble Sache!";
else if (-1 < one)
    cout << "So geht das!";
else
    cout << "???";
```

Dieses Programm ... ?

- 1 Kompiliert nicht
- 2 Bricht mit einem Laufzeitfehler ab
- 3 Gibt "Üble Sache!" aus ●
- 4 Gibt "So geht das!" aus
- 5 Gibt "???" aus



```
unsigned int one = 1;

if (one < -1)
    cout << "Üble Sache!";
else if (-1 < one)
    cout << "So geht das!";
else
    cout << "???";
```

- Variable `one` ist vom Typ `unsigned int`, Literal `-1` ist ein `int`



```
unsigned int one = 1;

if (one < -1)
    cout << "Üble Sache!";
else if (-1 < one)
    cout << "So geht das!";
else
    cout << "???";
```

- Variable `one` ist vom Typ `unsigned int`, Literal `-1` ist ein `int`
- `one < -1` ist daher ein *gemischter* ganzzahliger Ausdruck, in dem `-1` zu `unsigned int` konvertiert wird



```
unsigned int one = 1;

if (one < -1)
    cout << "Üble Sache!";
else if (-1 < one)
    cout << "So geht das!";
else
    cout << "???";
```

- Variable `one` ist vom Typ `unsigned int`, Literal `-1` ist ein `int`
- `one < -1` ist daher ein *gemischter* ganzzahliger Ausdruck, in dem `-1` zu `unsigned int` konvertiert wird
- Somit wird aus `-1` der grosse Wert `-1 + 2B ...`

Ganzzahlige Ausdrücke



```
unsigned int one = 1;

if (one < -1)
    cout << "Üble Sache!";
else if (-1 < one)
    cout << "So geht das!";
else
    cout << "???";
```

- Variable `one` ist vom Typ `unsigned int`, Literal `-1` ist ein `int`
- `one < -1` ist daher ein *gemischter* ganzzahliger Ausdruck, in dem `-1` zu `unsigned int` konvertiert wird
- Somit wird aus `-1` der grosse Wert $-1 + 2^B \dots$
- ...und `one < -1` evaluiert zu `true`