

Aufgabe 1

a)

HEX=0x70

DEZ=112

Das Bitweise oder von 1 (001) und 2 (010) ist 3 (011).

Der Operator << shifted nach links. Also wird a (0x0e/14) drei mal nach links geshiftet:

00001110	00011100	00111000	01110000
----------	----------	----------	----------

b)

```
while (1)
{
    printf("abc");
    break;
}
```

c)

```
int main ( void );
{
    int a = 1;
    ;
    while (a < 4);
        ++a;
    do ;
    {
        --a;
    };
    while (a > 2);
    return 0;
}
```

1)

- Semikolon verhindert, dass main deklariert wird -> Fehler da Deklaration keine Zuordnung zu einer Funktion hat
- gleicher Fehler, da Schleifenstatement nicht gefunden wird

2)

Während die bei 1) geschilderten Fehler das Programm nicht übersetzbar machen, da sie der Syntax widersprechen, hält sich der Fehler von 3) an die Syntax, weshalb der Compiler das Programm übersetzen kann. Das ändert aber nichts daran, dass es sich um Fehler handelt, welcher dem Programmablauf

- while Statement wird nicht erkannt, da Schleifenkörper durch Semikolon abgeschlossen	verändert.
3) - Das ++a ist notwendig, damit die Schleife terminiert -> Das Semikolon muss gestrichen werden, da die Schleife nicht vor dem Iterations-Statement geschlossen werden soll	4) Dieses Semikolon beendet kein Statement, deshalb kann es den Programmablauf nicht verändern.

d)

Die Warnstufe wird mit -W eingestellt. Um über alle Warnungen informiert zu werden, wird die Warnstufe "all" benötigt.

Eine Warnung kann mit -we, gefolgt von der Nummer des Fehlers, hochgestuft werden.

Bei einem Fehler ist das Programm nicht übersetzbar. Der Compiler kann oder darf nicht übersetzen. Bei einer Warnung übersetzt der Compiler das Programm doch, da das Programm nicht zwangsläufig defekt ist, oder bestimmte Warnungen beabsichtigt sind.

e)

- 1) $(3 < 5 - 2)$
 $(3 < 3)$
 $\Rightarrow 0$
- 2) $(!0 == 2)$
 $(1 == 2)$
 $\Rightarrow 0$

f)

- 1) 3
- 2) 1

Aufgabe 2

a) touch ggt.c ggt.h ggttest.c Makefile

b) git init

git add *

git commit -m "Initial commit"
=> per remote mit Gitlab verbunden

c)

- Kommentarblock
- ggttest: Objektliste
- run: ggttest
- %.o: %.c
- clean

d)

