

Datenstrukturen in C

Aufgabe 1

1.1

Bauen Sie ein Struct *Studenteninformation*, welches die Variablen Name, Matrikelnummer und Notenschnitt hält.

1.2

Weisen sie dem Studenten Norman Namenlos die Matrikelnummer 12345 und den Notenschnitt 2,9 zu und für Student Armin Anonym mit der Matrikelnummer 98765 und dem Notenschnitt 3,1

1.3

Lassen Sie sich auf der Konsole Matrikelnummer und Notenschnitt der Studenten ausgeben.

Aufgabe 2

2.1

```
typedef union {
    float gesammtschnitt;
    float note;
} MyUnion;

int main() {
    MyUnion noten;
    noten.gesammtschnitt = 2,7;
    noten.note = 1,3;    //Verändert auch den Wert von a.foo!
    assert(note.gesammtschnitt == 1,3);
}
```

Ist dieser Test erfolgreich?

2.2

Falls ja, erklären Sie warum der Test erfolgreich war.

Aufgabe 3

3.1

Welche Größe haben die Arrays:

```
Char string [20] = („Armin hat bestanden!");
Char string [] = („Norman hat nicht bestanden!");
```

3.2

Stellen Sie das Array grafisch dar (Zeichnung):

```
int notenspiegel[6]= {5,2,17,14,9,0};
```

3.3

Worin liegt der unterschied zum Array aus 3.2:

```
int notenspiegel[6]= {5,2,17,14,9};
```

Aufgabe 4

Machen Sie sich zunächst mit dem Code vertraut

```
int printTrafficLight( int light )
{
    if      ( light == 0 ) printf( "Ampel ist aus\n" );
    else if( light == 1 ) printf( "Ampel ist grün\n" );
    else if( light == 2 ) printf( "Ampel ist gelb\n" );
    else if( light == 3 ) printf( "Ampel ist rot\n" );
    else if( light == 4 ) printf( "Ampel ist rot und gelb\n" );
    else          printf( "Ampel ist kaputt (unbekannter Zustand)\n"
);
}
```

Was bewirken die Zahlen 0-4 und wofür stehen sie?

Bauen Sie ein Enum, mit passenden Begriffen und ersetzen sie 0-4 mit Wörtern aus Ihrem Enum.