

1. Ziel

Kennenlernen und Anwendung der Funktionen der Bibliothek string.h

2. Punkte

Bei dieser Aufgabe gibt es maximal 50 Punkte zu erreichen. Die Punkte teilen sich wie folgt auf die Teilaufgaben auf:

Nr.	Name	Punkte
1	Sortierte Namensliste	15
2	Mittlere Wortlänge	15
3	Hangman	20

3. Abgabetermin

Siehe Microsoft Teams

4. Abgabeform

Die Abgabe soll ein komprimiertes Verzeichnis (*.zip, *.7z, *.tar) mit den darin befindlichen C Dateien (*.c) enthalten.

Jede C Datei muss am Beginn einen Kommentarheader beinhalten (/**/) in dem der Name des Schülers und die Schulklasse, der Name der Aufgabe und das Erstellungsdatum sowie die Version des Programms enthalten ist.

```
1.  /*
2.      Name: DI Manuel Weigl - 2AHELS
3.      Titel: Erstes Programm
4.      Datum: 14.09.2019 - Version 1
5.  */
```

Die Namen der einzelnen C Dateien (Programme) sollen zumindest den Namen der Teilaufgabe und den eigenen Nachnamen enthalten.

Z.B: Teilaufgabe_Weigl.c

5. Aufgabenstellung

Löse die folgenden Teilaufgaben mit C in Code::Blocks.

Teilaufgaben:

1. Sortierte Namensliste: Schreibe ein Programm, das die Namen (Vor- und Nachnamen) von Schülern einer Klasse erfasst. Die Namen sollen in der Form `<Vorname>_<Nachname>` eingegeben werden.

Anschließend soll die Liste mit Schülern aufsteigend nach Nachnamen sortiert werden und mit den zugehörigen Katalognummern ausgegeben werden.

Beispiel:

Eingabe:

Paul Führig
Stefanie Artner
Daniel Zörgbauer

Ausgabe:

1. Stefanie Artner
2. Paul Führig
3. Daniel Zörgbauer

2. Mittlere Wortlänge: Schreibe ein Programm, das die mittlere Wortlänge eines Textes berechnet.

Als Trennzeichen zwischen den einzelnen Wörtern sind Leerzeichen, Beistriche und Punkte zugelassen.

Beispiel:

Eingabe Text:	Das ist mein eingelesener Satz.
Wortlängen:	3 3 4 12 4
Mittlere Wortlänge:	5,2

3. Hangman: Schreibe ein Programm, das das Spiel Hangman implementiert.

Es soll dabei ein zufälliges Wort aus einer fix vorgegebenen Liste ausgewählt werden. Der Benutzer soll nur der Reihe nach Buchstaben eingeben, die in einem leeren Feld an die richtigen Stellen geschrieben werden.

Der Benutzer bekommt 15 Versuche, um das Wort richtig zu erraten. Falls das Wort nicht richtig erraten wurde, dann soll es aufgedeckt werden. Andernfalls soll die Anzahl der benötigten Versuche ausgegeben werden.

5. Übung: Stringverarbeitung

Nachdem das Wort erraten wurde oder 15 Versuche verbraucht wurden, soll das Spiel von vorne beginnen und ein neues Wort ausgewählt werden.

Hinweis: Für ein besseres Erscheinungsbild verwende `console.h/c` um die Ausgaben zu platzieren, den Bildschirm zu leeren, etc.