

# ESP/Ass2 WS18

Aus Progwiki  
< ESP

## Inhaltsverzeichnis

- 1 Lernziel
- 2 ESPcrypter - Aufgabenstellung
  - 2.1 Einlesen und überprüfen
  - 2.2 Die Verschlüsselung
  - 2.3 Die Ausgabe
- 3 Fehlermeldungen
- 4 Beispielaufruf
  - 4.1 Hinweise
- 5 Spezifikation
  - 5.1 Abgabenliste
- 6 Verantwortliche Tutoren

## Lernziel

Basic I/O, Funktionen, Interpretation und Wertebereich von Datentypen, Datenstruktur: Array, Umgang mit Nachschlagewerken, einfaches Testen von Software

## ESPCrypter - Aufgabenstellung

Im zweiten Assignment wird ein Text mit einem einfachen Verschlüsselungsverfahren ähnlich der Caesar-Verschlüsselung[1] (<https://de.wikipedia.org/wiki/Caesar-Verschlüsselung>) chiffriert. Dazu muss das Programm folgende Aufgaben erledigen:

### Einlesen und überprüfen

Der Benutzer wird aufgefordert den zu verschlüsselnden Text einzugeben. Dieser Text darf maximal 255 Zeichen lang sein und nur aus den Kleinbuchstaben des englischen Alphabets ('a' bis 'z') sowie aus Leerzeichen bestehen. Die Eingabe endet mit '\n' und/oder EOF. Fehlerhafte Eingaben führen zu Fehlermeldungen.

```
plain text: [Eingabe]
```

### Die Verschlüsselung

256 wird durch die Anzahl der Buchstaben im Eingabetext dividiert. Der Restbetrag dieser Division bildet einen Schlüssel. Jeder Buchstabe des Eingabetextes wird um diesen Schlüssel im Alphabet zyklisch rotiert (d.h. bei einem Verschieben über z wird mit a weitergezählt). Bsp:

```
bruteforce
```

wird zu

```
hxazkluxik
```

Sollte der Restbetrag 0 ergeben, wird der Text rückwärts ausgegeben.

## Die Ausgabe

Im Erfolgsfall wird der verschlüsselte Text ausgegeben, 0 retourniert und das Programm beendet.

```
encrypted text: [Ausgabe]\n
```

## Fehlermeldungen

Bei Eingabe über 255 Buchstaben (oder Leerzeichen) wird -1 retourniert und das Programm beendet sich mit folgender Fehlermeldung:

```
[ERR] too many characters\n
```

Bei invaliden Zeichen innerhalb der ersten 255 Zeichen wird -2 retourniert und das Programm beendet sich mit folgender Fehlermeldung:

```
[ERR] invalid characters\n
```

Sollten beide Fehler eintreten, wird erstere(-1) ausgegeben.

## Beispielaufruf

```
$ ./ass2
plain text: whyn0tall ASCII?
[ERR] invalid characters
$ ./ass2
plain text: notverysecure thiscaeserwas
encrypted text: jkpranu0ayqna pdeoywaoanswo
$ ./ass2
plain text: four
encrypted text: ruof
$ ./ass2
plain text:
encrypted text:
```

```
$ ./ass2  
plain text:  
encrypted text:  
$
```

## Hinweise

Lesen Sie sich in Nachschlagewerke[2] (<http://www.cplusplus.com/reference/cstring/>) ein, es gibt nützliche String und Ein/Ausgabe Funktionen (beachten Sie deren Umgang mit terminating null und newline character)! Testen Sie ihr Programm!

## Spezifikation

- Keinerlei weitere Ausgaben.
- alle Ausgaben haben auf stdout zu erfolgen
- Erlaubte Bibliotheken
  - alle C Standard Bibliotheken
- Abgabe
  - Dateinamen laut Abgabenliste
  - Archiv beinhaltet keine Verzeichnisse oder andere Dateien
  - Abgabe bis spätestens 06.12.2018 14:00:00 Uhr

## Abgabenliste

- Quellcode (ass2.c) in einem .tar.gz oder .zip Archiv (ass2.tar.gz oder ass2.zip)

## Verantwortliche Tutoren

- Michael Hancianu

Von „[https://palme.iicm.tugraz.at/wiki/ESP/Ass2\\_WS18](https://palme.iicm.tugraz.at/wiki/ESP/Ass2_WS18)“

---

- Diese Seite wurde zuletzt am 2. Dezember 2018 um 18:11 Uhr geändert.
- Inhalt ist verfügbar unter der Attribution-NonCommercial-NoDerivs 3.0 Austria.