

1 Strings als Container 2

1.1 Aufgabe

Schreibe eine Funktion `contains`. Dieser wird ein String und ein Zeichen übergeben. Sie gibt zurück, ob das Zeichen in dem String enthalten ist.

```
1 contains("Hello", 'l')
```

true

```
1 contains("Bye", 'c')
```

false

1.2 Aufgabe

Schreibe eine Funktion `count`. Dieser wird ein String und ein Zeichen übergeben. Sie gibt zurück, wie oft das Zeichen in dem String enthalten ist.

```
1 count("Hello", 'l')
```

2

```
1 count("Bye", 'c')
```

0

1.3 Aufgabe

Schreibe eine Funktion `charsAndBars`. Dieser werden zwei Strings übergeben. Sie gibt einen String zurück, in dem alle Zeichen aus dem ersten String, die nicht im zweiten String vorkommen durch Bindestriche ersetzt wurden. Alle anderen Zeichen werden unverändert übernommen.

```
1 charsAndBars("Hello", "Holla")
```

H-llo

```
1 charsAndBars("Hello", "Bye")
```

-e---

1.4 Aufgabe

Schreibe eine Funktion `replace`. Dieser wird ein String, eine ganze Zahl und ein Zeichen übergeben. Sie gibt einen String zurück. In diesem wurde das Zeichen, dessen Index der übergebenen Zahl entspricht, durch das übergebenen Zeichen ersetzt.

```
1 replace("Hello", 0, 'C')
```

Cello

```
1 replace("Bye", 1, 'ö')
```

Böe

1.5 Aufgabe

Schreibe eine Funktion `slice`. Dieser werden ein String und zwei ganze Zahlen übergeben. Sie gibt die Zeichen aus dem String zurück, deren Indizes zwischen diesen Zahlen liegen.

```
1 slice("Hello", 1, 3)
```

ell

```
1 slice("Bye", 0, 2)
```

Bye

Du kannst davon ausgehen, dass die übergeben Zahlen nicht größer als der größte Index oder kleiner als der der kleinste Index des Strings sind.

1.6 Aufgabe

Schreibe eine Funktion `withIndex`. Dieser wird ein String übergeben. Sie gibt einen String zurück, in dem vor jedem Zeichen aus diesem String, der Index des Zeichens steht.

```
1 withIndex("Hello")
```

0: H 1: e 2: l 3: l 4: o

```
1 withIndex("Bye")
```

0: B 1: y 2: e

1.7 Aufgabe

Schreibe eine Funktion `smallest`. Dieser wird ein Integer übergeben. Sie gibt die kleinste Ziffer in dieser Zahl zurück.

```
1 smallest(4819)
```

1

```
1 smallest(2359)
```

2

Konvertiere die Character zu Strings und anschließend zu Integern um sie zu vergleichen.

1.8 Aufgabe

Schreibe eine Funktion `removeLast`. Dieser wird ein String übergeben. Sie gibt diesen String ohne das letzte Zeichen zurück.

```
1 removeLast("Hello")
```

Hell

```
1 removeLast("Bye")
```

By

1.9 Aufgabe

Schreibe eine Funktion `last`. Dieser wird ein String übergeben, der nicht leer ist. Sie gibt das letzte Zeichen des Strings zurück.

```
1 last("Hello")
```

o

```
1 last("Bye")
```

e

1.10 Aufgabe

Schreibe eine Funktion `dropLast`. Dieser wird ein String und eine ganze Zahl n übergeben. Sie gibt diesen String ohne die letzten n Zeichen zurück.

```
1 dropLast("Hello", 2)
```

Hel

```
1 dropLast("Bye", 3)
```

1.11 Aufgabe

Schreibe eine Funktion `everyThird`. Dieser wird ein String übergeben. Sie gibt einen String zurück, in dem nur jedes 3. Zeichen aus dem übergeben String enthalten ist.

```
1 everyThird("Hello")
```

Hl

```
1 everyThird("Emmendingen")
```

Eeie