

1 Strings als Container 3

1.1 Aufgabe

Schreibe eine Funktion `neighbours`. Dieser wird ein String und ein Index i übergeben. Sie gibt die Nachbarn des Elements mit diesem Index inklusive dem Element selbst als String zurück. Wenn das Element keine 2 Nachbarn hat, wird ein kürzerer String zurückgegeben.

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

llo

```
1 neighbours(1, "GHSE")
```

GHS

```
1 neighbours(0, "GHSE")
```

GH

1.2 Aufgabe

Schreibe eine Funktion `countAsInNeighbourhood`. Dieser wird ein String und ein Index i übergeben. Sie zählt wie viele der Nachbarn des Elements mit diesem Index (inklusive dem Element selbst) `a` sind.

```
1 countAsInNeighbourhood(3, "abadef")
```

1

```
1 countAsInNeighbourhood(1, "abadef")
```

2

```
1 countAsInNeighbourhood(0, "abadef")
```

1

1.3 Aufgabe

Schreibe eine Funktion `reverse`. Diese gibt einen übergebenen String gespiegelt zurück.

```
1 reverse("Hello")
```

olleH

```
1 reverse("GHSE")
```

ESHG

1.4 Aufgabe

Schreibe eine Funktion `intAt`. Dieser wird ein Index und ein String übergeben, in dem an dieser Stelle eine Ziffer steht. Sie gibt die Ziffer als Integer zurück.

```
1 intAt(2, "ab5t")
```

5

```
1 intAt(0, "15xy")
```

1

1.5 Aufgabe

Schreibe eine Funktion `add`. Dieser werden zwei Strings die Vektoren darstellen übergeben. Sie gibt die Summe dieser Vektoren als String zurück.

```
1 add("(1,2)", "(3,1)")
```

(4,3)

```
1 add("(2,3)", "(2,0)")
```

(4,3)

Du kannst davon ausgehen, dass die Koordinaten der Vektoren zwischen 0 und 9 sind.