DS 12 – Weitere Übungen

Algorithmen auf Listen: Summe, Minimum und Suche

Hier kannst du das Gelernte weiter üben. Die Punkte im Titel zeigen, wie schwer eine Aufgabe ist: ●○○ leicht, ●●○ mittel, ●●● knifflig. Fang mit den leichten an.

Die Aufgaben stehen auch im Skript im Abschnitt „Weitere Übungen“; die Lösungen im Lösungsheft.

Aufgabe 1: Maximum von Hand ●○○

Das Programm sucht das Maximum der Liste `[4, 9, 2, 9, 6]`:

```
zahlen = [4, 9, 2, 9, 6]
maximum = zahlen[0]
for zahl in zahlen:
    if zahl > maximum:
        maximum = zahl
print("Maximum:", maximum)
```

```
Maximum: 9
```

Fülle die Tabelle aus: Ist das Element größer als das bisherige Maximum? Welchen Wert hat `maximum` nach diesem Durchlauf? Gib dann an, wie oft `maximum` insgesamt geändert wird.

Durchlauf	`zahl`	`zahl > maximum?`	`maximum` danach
vor der Schleife	–	–	`4`
1	`4`		
2	`9`		
3	`2`		
4	`9`		
5	`6`		

Deine Antwort:

…

Aufgabe 2: Summe und Zählen ●○○

In der Liste stehen die Punkte, die ein Team in sechs Spielen geholt hat. Gib **ohne Computer** an, was das Programm ausgibt, und beschreibe in einem Satz, was `anzahl` zählt.

```
punkte = [3, 0, 1, 3, 0, 3]
summe = 0
anzahl = 0
for p in punkte:
    summe = summe + p
    if p == 0:
        anzahl = anzahl + 1
print(summe, anzahl)
```

```
10 2
```

Deine Antwort:

…

Aufgabe 3: Playlist ●○○

In einer Liste steht die Länge jedes Songs einer Playlist in Minuten: `songs = [3, 4, 2, 5, 3]`.

1. Implementiere eine Funktion `gesamtdauer(liste)`, die die Summe aller Elemente zurückgibt.
2. Implementiere eine Funktion `mittlere_dauer(liste)`, die die durchschnittliche Länge zurückgibt. Sie soll `gesamtdauer` aufrufen.
3. Teste beide Funktionen mit der Playlist. Erwartet sind 17 Minuten und 3,4 Minuten.

Aufgabe 4: Kälteste Nacht ●○○

In einer Woche wurden nachts diese Temperaturen gemessen (in °C): `nachts = [-3, 2, -7, 0, 4, -1, 1]`.

Implementiere eine Funktion `kaelteste(temperaturen)`, die die niedrigste Temperatur zurückgibt. Verwende nicht `min()`. Teste mit der Liste. Erwartet ist −7.

Aufgabe 5: Freies Schließfach ●●○

In der Liste `faecher` steht, wem welches Schließfach gehört. Ein leeres Fach ist mit der leeren Zeichenkette `""` gekennzeichnet:

```
faecher = ["Ada", "Ben", "", "Mia", ""]
```

1. Implementiere eine Funktion `erstes_freies(faecher)`, die mit einer linearen Suche den **Index** des ersten freien Fachs zurückgibt. Gibt es kein freies Fach, soll sie `-1` zurückgeben.
2. Gib im Hauptprogramm aus, welches Fach frei ist. Die Fächer sind für Menschen ab 1 nummeriert, also ist der Index 2 das Fach Nummer 3. Ist kein Fach frei, soll „Alle Fächer sind belegt.“ erscheinen.
3. Teste auch mit einer Liste ganz ohne freies Fach.

Aufgabe 6: Zwei Fehler im Maximum ●●○

Die Funktion soll die beste Tordifferenz einer Saison zurückgeben. Sie liefert falsche Ergebnisse.

```
def beste_differenz(differenzen):
    beste = 0
    for d in differenzen:
        if d > beste:
            beste = d
        return beste
```

1. Teste die Funktion mit `[3, -1, 5]` (erwartet: 5) und mit `[-2, -1, -5]` (erwartet: −1). Notiere die Ergebnisse.
2. Erkläre die beiden Fehler und verbessere die Funktion.

Deine Antwort:

…

Aufgabe 7: Frosttage ●●○

Ein Tag mit einer Temperatur **unter** 0 °C heißt Frosttag.

1. Implementiere eine Funktion `frosttage(temperaturen)`, die zählt, wie viele Werte der Liste unter 0 liegen.
2. Teste mit der Liste aus dem Auftrag *Kälteste Nacht* (erwartet: 3) und mit `[0, 1, 2]` (erwartet: 0).

Aufgabe 8: Besser als der Durchschnitt ●●●

Implementiere eine Funktion `ueber_durchschnitt(liste)`, die zählt, wie viele Elemente **größer** als der Durchschnitt der Liste sind. Teste mit:

Aufruf	erwartet	tatsächlich
`ueber_durchschnitt([2, 4, 6, 8])`	`2`	
`ueber_durchschnitt([10, 10, 10])`	`0`	
`ueber_durchschnitt([1, 1, 1, 9])`	`1`	

> **Tipp:** Du brauchst **zwei** Schritte nacheinander. Erst berechnest du mit einer Schleife den Durchschnitt. Erst danach kannst du mit einer zweiten Schleife zählen, denn vorher kennst du den Vergleichswert noch nicht.