DS 13 – Weitere Übungen

Projekt Verschlüsselung (Teil 1): die Caesar-Chiffre

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: Caesar mit Schlüssel 5 ●○○

Benutze das Alphabet mit Positionen. Ein Tipp: Schreibe dir zuerst die Positionen auf, addiere 5 und rechne bei Ergebnissen ab 26 noch 26 ab.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

1. Verschlüssele das Wort `PIZZA` mit dem Schlüssel 5.
2. Entschlüssele den Geheimtext `KJWNJS`. Er wurde mit dem Schlüssel 5 verschlüsselt.

Deine Antwort:

…

Aufgabe 2: Rechnen mit % 26 ●○○

Berechne zu jeder Zeile die neue Position `(position + schluessel) % 26` und den Buchstaben, der dort steht. Benutze die Positionstabelle aus dem vorigen Auftrag.

position	schluessel	(position + schluessel) % 26	Buchstabe
24	4		
10	13		
3	30		
25	26		

Was fällt dir in der letzten Zeile auf? Erkläre.

Deine Antwort:

…

Aufgabe 3: Zweimal verschlüsseln ●●○

Die Funktionen `verschluessle_buchstabe` und `verschluessle_text` (verbesserte Fassung aus Teilaufgabe B) stehen vor diesen beiden Zeilen:

```
print(verschluessle_text("HALLO", 13))
print(verschluessle_text(verschluessle_text("HALLO", 13), 13))
```

```
Traceback (most recent call last):
  File "<Zelle 1>", line 1, in <module>
NameError: name 'verschluessle_text' is not defined
```

1. Gib an, was die beiden Zeilen ausgeben.
2. Erkläre, warum in der zweiten Zeile wieder der Klartext herauskommt.
3. Gibt es einen weiteren Schlüssel zwischen 1 und 25 mit dieser Eigenschaft? Begründe.

Deine Antwort:

…

Aufgabe 4: Geheimtext in Fünfergruppen ●●●

Früher hat man Geheimtexte ohne Leerzeichen und in Gruppen zu je fünf Buchstaben aufgeschrieben. So verrät die Länge der Wörter nichts mehr über den Klartext.

1. Implementiere eine Funktion `fuenfergruppen(text)`. Sie entfernt zuerst alle Leerzeichen und fügt dann nach je fünf Zeichen ein Leerzeichen ein. Aus `"YWJKKJS ZR FHMY"` wird so `"YWJKK JSZRF HMY"`.
2. Teste die Funktion zusammen mit `verschluessle_text`: Verschlüssele „Treffen um acht“ mit dem Schlüssel 5 und gib das Ergebnis in Fünfergruppen aus.

> **Tipp:** Zerlege die Aufgabe in zwei Schritte. Schritt 1: Baue einen neuen Text ohne Leerzeichen auf (Zeichen nur anhängen, wenn es kein `" "` ist). Schritt 2: Laufe über die **Indizes** dieses Textes. Vor jedem Zeichen, dessen Index durch 5 teilbar ist (außer beim Index 0), hängst du zuerst ein Leerzeichen an. „Durch 5 teilbar“ prüfst du mit `i % 5 == 0`.