Variablen und Eingabe:

Aufgabe 1) Umwandlung von Fahrenheitgraden in Celsiusgrade (1 Pkt)
Schreiben Sie ein Progamm, welches Fahrenheitgrade einliest und Celsiusgrade ausgibt. Verwenden Sie den Befehl „`input`“ zur Eingabe.

Aufgabe 2) Politikergehälter (1 Pkt)
Herr und Frau P. sind Politiker. Sie haben je ein Grundeinkommen von 100.000 Euro. Dazu kommt eine Abgeordnetenpauschale von 40.000 für Herrn P. Und 50.000 für Frau P. Zusätzlich bekommen sie für jede Rede 500 Euro und für jede Stunde Sekretariatsarbeit 200 Euro. Schreiben Sie ein Programm, das aus den ausgeführten Reden und Arbeiten das Gesamteinkommen des Ehepaares berechnet. Verwenden Sie den Befehl „`input`“ zur Eingabe.

Aufgabe 3) Zeitrechnung (1 Pkt)
Schreiben Sie ein Programm, das eine Anzahl von Sekunden einliest und in die Anzahl von Stunden, Minuten und Sekunden umrechnet und ausgibt. Die Eingabe 1234 soll zur formatierten Ausgabe von 0:20:34 führen. Verwenden Sie einen String-Formatter.

Verzweigungen:

Aufgabe 4) Dreiecksbestimmung (1 Pkt)
Schreiben Sie ein Programm, das die Seitenlängen eines Dreiecks einliest und prüft, ob es ein gleichseitiges, ein gleichschenkeliges, ein rechtwinkeliges, ein sonstiges gültiges oder ein ungültiges Dreieck ist. Ein Dreieck ist ungültig, wenn die Summe zweier Seitenlängen kleiner oder gleich der dritten Seitenlänge ist. Beachten Sie, dass ein Dreieck sowohl rechtwinkelig, als auch gleichschenkelig sein kann.

Aufgabe 5) Zahl umwandeln in Wochentag (1 Pkt)
Schreiben Sie ein Programm, welches eine Zahl einliest und den zugehörigen Wochentag ausgibt. Zahlen kleiner 1 und über 7 sollen zu einer Fehlermeldung führen. Verwenden Sie `match/case`.

Aufgabe 6) Plausibilitätsprüfung (2 Pkte)
Lesen Sie ein Datum in Form dreier Zahlen für den Tag, den Monat und das Jahr ein. Prüfen Sie, ob es sich um ein gültiges Datum handelt. Berücksichtigen Sie auch Schaltjahre. Ein Jahr ist ein Schaltjahr, wenn es durch 4 teilbar ist. Jahre, die durch 100, aber nicht durch 400 teilbar sind, sind keine Schaltjahre.

Schleifen:

Aufgabe 7) Einmaleins (1 Pkt)
Schreiben Sie ein Programm, welches das kleine Einmaleins als Tabelle ausgibt.

Aufgabe 8) Ziffernsumme einer Zahl (1 Pkt)
Schreiben Sie ein Programm, das die Ziffernsumme einer Zahl x berechnet und ausgibt. Für 4711 beträgt die Ziffernsumme zum Beispiel 13.

Aufgabe 9) Binärzahlen als Menge (3 Pkte)
Die binäre Darstellung einer Zahl kann als Zahlenmenge interpretiert werden. Das Element i ist in der Menge enthalten, wenn das Bit an der Stelle i gesetzt ist. Die Zahl 44 lautet zum Beispiel 101100 und entspricht der Menge {2,3,5}.
Schreiben Sie ein Programm, das eine integer-Zahl einliest und in dieser Darstellung wieder ausgibt. Bitte kein Python-Set verwenden, das Wort „Menge“ ist im mathematischen Sinne gemeint.

Hinweis: Sie können die jeweils letzte Binärziffer einer Zahl n durch n/2 abspalten bzw. mit n%2 prüfen, ob sie 0 oder 1 ist. Die Elemente der Zahlen können in eigene Zeilen geschrieben werden:
2
3
5
Verwenden Sie KEINE Bibliotheks-Funktionen zur Konvertierung von Dezimal in Binär.

Zeichenketten und Funktionen:

Aufgabe 10) Leet (3 Pkte)
Schreibe ein Programm, das Buchstaben durch Ziffern ersetzt, ähnlich dem Internet-Slang Leet (https://en.wikipedia.org/wiki/Leet).
Die folgenden Buchstaben sollten ersetzt werden:
L / 1
E / 3
T / 7
O / 0
S / 5
Das Programm muss korrekt mit Groß- und Kleinbuchstaben funktionieren.

Aufgabe 11) Anagramme (1 Pkt)
Zwei Wörter sind Anagramme, wenn sie aus denselben Buchstaben in beliebiger anderer Reihenfolge bestehen (z.B. "Maus" und "Saum"). Schreiben Sie eine Methode isAnagram(s1,s2), die prüft, ob die beiden Strings s1 und s2 Anagramme sind und das Ergebnis als boolean-Wert zurückgibt. Groß und Kleinschreibung der Strings soll ignoriert werden.

Aufgabe 12) Lauflängencodierung (3 Pkte)
Die Lauflängencodierung (run length encoding) ist eine Komprimierungstechnik, bei der jede Zeichenfolge, die aus mehr als 2 gleichen Zeichen besteht, durch das Zeichen und die Länge der Folge codiert wird. Die Eingabe "ABBCCCKKKKKKK" wird zum Beispiel zu "ABBC3K7".
a) Schreiben Sie eine Methode, die einen Buchstaben-String nach diesem Verfahren codiert.
b) Schreiben Sie eine Methode, die einen nach diesem Verfahren codierten String decodiert.
c) Ist es sinnvoll, eine komprimierte Zeichenfolge nochmals zu komprimieren?

Datenstrukturen

Aufgabe 13) Datenmodellierung (3 Pkte)
Ein Buch ist beschrieben durch „Titel“, „Autor“ und „ISBN-Nummer“
Modellieren Sie:

- Verwenden Sie eine Beispiel-Datenstruktur, um auf diese Elemente mit zuzugreifen: "Numerisches Python", "Bernd Klein", "978-3-446-45076-9"
- Modellieren Sie weiter: das Buch wird auf ein Regal gestellt. Fügen Sie 3 weitere Bücher ihrer Wahl ein. Nun greifen Sie auf das dritte Buch im Regal zu.
- Ein Haus hat zumindest die Räume „Wohnzimmer“ und „Büro“. Ihr Regal steht im Wohnzimmer. Das Regal des Büros ist leer. Greifen Sie auf die ISBN-Nummer des 2.Buches im Regal des Wohnzimmers zu.
- Entnehmen Sie das erste Buch aus dem Regal im Wohnzimmer und fügen Sie es dem Regal im Büro hinzu.

Sonderaufgaben

Aufgabe 14) Manuelles Multiplizieren (5 Pkte)

Schreibe eine Funktion, die zwei ganze Zahlen multipliziert. Die Funktion sollte die folgende Signatur haben:

```
final_res = mult_manu(num_a, num_b)
```

mult_manu(123,456) sollte die folgende Ausgabe liefern:

```
123 * 456
---------
492
 615
  738
---------
56088
```

2 Pkt für richtige Zwischen- und Endergebnisse, 3 Punkte für komplett gleiches Layout (Anzahl der ‚-‘ sowie korrekte Einrückungen)!

Das Programm ist als ausführbare Sourcedatei gemeinsam mit einer Beschreibung der Lösung als PDF in einer ZIP-Datei abzugeben.

Aufgabe 15) Vigenere Bruteforce Cracker (7 Pkte)
Informiereen Sie sich was eine Vigenère-Verschlüsselung ist (https://de.wikipedia.org/wiki/Vigen%C3%A8re-Chiffre) und lassen Sie sich durch ein KI Tool (ChatGPT, etc) die Funktionen:

`vigenere_encrypt(plain_text, key)`
und `vigenere_decrypt(chiffre_text, key)`
erstellen. Validieren Sie die Funktionen !!!

Entwicklen Sie einen Brute-Force Key und Password Cracker der für einen Schlüssel / ein Passwort (key) alle möglichen Buchstabenkombinationen ausprobiert.

Folgendes ist bekannt:

- Der Eingabetext (chiffre) ist mit Vigenere verschlüsselt
- Key und Klartext beinhalten nur Kleinbuchstaben, keine Zahlen/Sonderbuchstaben
- Key-Länge ist <= 5
- Der Klartext beginnt mit "`hallo`"
- Der verschlüsselte Text lautet "`eqvpmtabpb`"

Wie lautet der key und wie lautet der volle Klartext? (beides ist nötig) (5 Pkte)

Hinweis: Achten Sie auf den Speicherverbrauch des Programmes und bauen Sie bei Bedarf die Schlüssel direkt in der Schleifen zusammen, ohne eine Liste von allen möglichen Schlüssel im Vorhinein zu erzeugen.

Weiters:

"`import time`" macht die den Befehl "`t = time.time()`" verfügbar, der in `t` die aktuellen Millisekunden seit January 1, 1970, 00:00:00 speichert

- Wie lange dauert es auf Ihrem Computer alle Schlüssel mit Länge <= 5 auszuprobieren? (1 Pkte)
- Wie lange dauert es alle Schlüssel mit der Länge <= 6 auszuprobieren? (1 Pkte)
  Vergleiche mit https://www.hivesystems.io/blog/are-your-passwords-in-the-green

Das Programm ist als ausführbare Sourcedatei gemeinsam mit einer Beschreibung der Lösung als PDF in einer ZIP-Datei abzugeben.