**Fragen Woche 2**

### 2.1.1

Schreibe ein Programm, das die größere der Zahlen A und B ausgibt. Die Festlegung, welchen Wert A und B haben, wird vom Computer getroffen - Dein Programm soll die jeweils richtige Ausgabe erzeugen. Beginne dazu Dein Programm mit

from daten import A,B | wahlweise a,b=randint(0,100),randint(0,100)

Wenn beispielsweise A den Wert 3 und B den Wert 5 hat, soll Dein Programm die Ausgabe
5 ist die größere Zahl.
Falls die beiden Zahlen gleich sind, soll die Ausgabe
Beide Zahlen sind gleich.
erscheinen.

Hinweis: Zum Ausprobieren kannst Du zunächst mit festen Werten arbeiten, zum Beispiel

A=3

B=5

Zum Abgeben musst Du dann die import-Zeile oben benutzen.

### 2.1.2

Schreibe ein Programm, das für eine gegebene Zahl A den Absolutwert ausgibt. Den Absolutwert erhältst Du, wenn Du das Vorzeichen der Zahl weglässt: für -5 ist der Absolutwert also 5; für 5 (also +5) ist der Absolutwert ebenfalls 5. Die vom Computer vorgegebene Zahl erhältst Du, indem Du das Programm mit

from daten import A

beginnst.

Du musst keinen ganzen Satz ausgeben; es reicht, wenn Dein Programm bei -5 einfach

5

ausgibt.

Hinweis: Zum Ausprobieren kannst Du zunächst mit festen Werten arbeiten, zum Beispiel

A=-5

Zum Abgeben musst Du dann die import-Zeile oben benutzen.

Hinweis: Wenn Du die Fehlermeldung IndentationError bekommst, bedeutet das, dass die Zeilen nicht richtig eingerückt sind.

Hinweis: Um für eine Zahl A das Vorzeichen umzukehren, kann man -A schreiben, oder mit -1 multiplizieren.

### 2.1.3

Schreibe ein Programm, das für eine Variable Hunde ausgibt, wie viele es sind, und zwar in der Form

* Wir haben keinen Hund.
* Wir haben einen Hund.
* Wir haben zwei Hunde.
* Wir haben viele Hunde. Die Variable Hunde erhältst Du durch

from daten import Hunde | wahlweise Hunde=randint(0,100)

### 2.2.1

Schreibe ein Programm, das die Quadratzahlen von 0 \* 0 bis 20 \* 20 ausgibt

### 2.2.2

Schreibe ein Programm, das untereinander alle geraden Zahlen von 0 bis 25 ausgibt (also jede Zahl in eine neue Zeile).

### 2.2.3

Schreibe ein Programm, das gedrehte gleichseitige Dreiecke zeichnet (siehe Abbildung). Die Zahl der Dreiecke, ihre Größe, und der Winkel, um die sie gedreht sind, kannst Du Dir selbst aussuchen.



**Hinweis: Ein gleichseitiges Dreieck entsteht, wenn der Winkel im Dreieck 60° beträgt und alle Seiten gleich lang sind. Die Schildkröte muss sich an jeder Ecke um 120° nach links oder rechts drehen, damit der verbleibende Winkel 60° ergibt.**

### 2.3.1

Schreibe ein Programm, das die Summe der Zahlen aus einer Liste bildet. Die Liste erhältst Du durch from daten import Zahlen | (oder Liste mit randint füllen:

def randomlist(size):

 numlist=[randint(0,10) for i in range(size)]

 return numlist

### 2.3.2

Schreibe ein Programm, das den Durchschnitt der Zahlen aus einer Liste bildet. Den Durchschnitt erhältst Du, wenn Du die Summe der Zahlen durch ihre Anzahl (also die Länge der Liste) teilst. Die Liste der Zahlen bekommst Du durch(s.o.)

### 2.3.3

Schreibe ein Programm, das die kleinste und die größte Zahl in einer Liste namens Zahlen ausgibt, zum Beispiel in der Form

Die kleinste Zahl ist 10 und die größte Zahl ist 97.

Du musst aber nicht unbedingt einen ganzen Satz ausgeben.</p>

Die Liste erhältst Du mit

from daten import Zahlen| Liste mit randint füllen:

Hinweis: Neben einer Schleife musst Du hier auch if-Anweisungen einsetzen.

Hinweis: Zahlen können auch negativ sein.

Hinweis: Zum Ausprobieren kannst Du zunächst mit festen Werten arbeiten, zum Beispiel

Zahlen = [41, 25, 40, 97, 76, 40, 43, 10]

Zum Abgeben musst Du dann die import-Zeile oben benutzen.

### 2.4.1

Wie viele der ersten natürlichen Zahlen (1,2,3,4,…,N) muss man addieren, damit die Summe nicht mehr kleiner als (ein vorgegebenes) X ist. Schreibe ein Programm, das diese Frage beantwortet und N ausgibt.

### 2.4.2

Schreibe ein Programm, das das erste Wort in einer Liste Namen findet, das mit D beginnt.

Hinweis: Es gibt zwei Möglichkeiten:

1. Du schreibst eine while-Schleife, die der Reihe nach die Namen anschaut, solange sie nicht mit D anfangen. Dazu verwendest Du einen Zähler, der bei 0 anfängt, und der Reihe nach alle Namen abfragt.
2. Du schreibst eine for-Schleife, die alle Namen durchgeht und abbricht (mit break), wenn ein Name mit D gefunden wurde </ol></p>

Hinweis: Den ersten Buchstaben eines Wortes W bekommst Du mit W[0]. Probier's aus!

lst =['Klaus', 'Anna', 'Dora', 'Julia', 'Hans', 'Daniel', 'Jana']

### 2.4.3

Schreibe ein Programm, das alle Teiler einer Zahl N ausgibt, jeden in eine eigene Zeile. Zum Beispiel soll für N = 24 die Ausgabe

2

3

4

6

8

12

entstehen (die Zahlen 1 und 24 werden also jetzt nicht als Teiler verstanden - es sind nur die "echten" Teiler gefragt). Den Wert N erhältst mit from daten import N | s.o.

Hinweis: Um zu bestimmen, ob eine Zahl Teiler ist, kann man fragen, ob der Rest bei der Division 0 ist. Die Berechnung des Rests hast Du ja schon in Aufgabe 1.2.3 durchgeführt.

Hinweis: Um die Teiler alle zu bekommen, kann man einfach alle Zahlen von 2 bis N-1 ausprobieren. Ein Mensch kann leicht viele von diesen Zahlen ausschließen; für ein Programm ist es aber einfacher, sie alle durchzuprobieren.

**Fragen Woche 3**

3.1.1 Schreibe eine Funktion Weihnachtsbaum, die 2 Paramater x und y erwartet und den Weihnachtsbaum an diesen Koordinaten zeichnet. Die genaue Form des Weihnachtsbaums ist Dir überlassen.

Rufe dann die Funktion mit den Koordinaten x=50 und y=100 auf.

3.1.2 Schreibe eine Funktion Weihnachtsbaum, die 2 Paramater x und y erwartet und den Weihnachtsbaum an diesen Koordinaten zeichnet.

Schreibe eine weitere Funktion Weihnachtswald, die 9 Weihnachtsbäume an verschiedene Positionen malt. Die Positionen kannst Du Dir selber aussuchen (und zum Beispiel die Bäume in einem 3x3-Feld anordnen)

Rufe dann die Funktion Weihnachtswald auf

3.1.3 Schreibe eine Funktion Summe, die die Summe der ersten N Zahlen berechnet und mit print ausgibt. Dabei soll N der Parameter der Funktion sein. Zum Beispiel soll Summe(4) die Summe 1+2+3+4 = 10 ausrechnen und die Zahl 10 ausgeben.

Zum Testen kannst Du überprüfen, ob Summe(100) den Wert 5050 ergibt.

Hinweis: Vermeide die Verwendung einer Summenformel. Du kannst die Summe auch dadurch bestimmen, indem einfach die Zahlen alle zusammenaddiert werden

3.2.1 Schreibe eine Funktion Wörter\_mit\_a, die aus einer Liste mit Wörtern die ermittelt, die mit a anfangen.

3.2.2 Schreibe eine Funktion Anzahl\_Vokale, die für ein Wort bestimmt, wie viele Vokale es hat und diese Anzahl zurückgibt. Umlaute zählen hier auch als Vokale; Großbuchstaben müssen nicht (aber dürfen) berücksichtigt werden. Z.B. soll sie für das Wort "bestimmt" den Wert 2 als Ergebnis liefern.

Schreibe dann eine Funktion Viele\_Vokale, die zwei Parameter hat. Der erste Parameter ist eine Liste von Wörtern, und der zweite Parameter eine Anzahl. Das Ergebnis der Funktion ist die Liste der Wörte, die mindestens die angegebene Anzahl von Vokalen hat.

Z.B. soll ein Aufruf von

Viele\_Vokale(['schreibe', 'dann', 'eine', 'funktion', 'viele\_vokale,',

 'die', 'zwei', 'parameter', 'hat'], 4)

die Liste ['viele\_vokale,', 'parameter'] als Ergebnis liefern.

3.2.3 Schreibe eine Funktion ist\_Palindrom, die ein Wort als Parameter erwartet und wahr zurückgibt, wenn das Wort ein Palindrom ist, ansonsten falsch. Ein Palindrom ist ein Wort, das vorwärts und rückwärts gelesen gleich lautet, z.B. Rentner, Reittier, oder Lagerregal.

Zur Erinnerung: Die Werte wahr und falsch werden in Python durch True und False ausgedrückt.

Du kannst für die Aufgabe auf Großschreibung verzichten (also annehmen, dass alle Wörter kleingeschrieben werden).

Zum Testen des Programms kannst Du alle Palindrome in einem Satz ausgeben lassen, z.B.

Worte = "anna und otto haben ein pferd als reittier".split()

for Wort in Worte:

 if ist\_Palindrom(Wort):

 print(Wort)