

Aufgabe max

- Schreiben Sie eine Funktion die für zwei gegebene Zahlen die größere zurückgibt
- Schreiben Sie eine Funktion, die für eine Liste von Zahlen die größte zurückgibt

Aufgabe Palindrom

Schreiben Sie eine Funktion *palindrom*, die überprüft, ob ein String ein Palindrom ist (also vorwärts und rückwärts gelesen das gleiche ergibt).

Implementieren Sie diese Funktion sowohl iterativ, als auch rekursiv!

read_file

Schreiben Sie eine Funktion *read_file*, die einen Dateinamen als Argument erwartet und eine Liste mit den Zeilen der ausgelesenen Datei zurückgibt

write_file

Schreiben Sie eine Funktion *write_file*, die als Argumente einen Dateinamen, sowie eine Liste mit Strings erwartet und diese in eine Datei schreibt.

Zahlen von 1 bis n

Schreiben Sie eine Funktion, die bei Eingabe einer natürlichen Zahl *n* die Zahlen von 1 bis *n* ausgibt. Jedoch ohne eine Schleife zu benutzen!

Matrix Multiplikation

Schreiben Sie eine Funktion *mat_mul*(*matrix_a*, *matrix_b*), die zwei Matrizen als Eingabe nimmt und ihr Produkt ausgibt.

Hinweis:

Eine Matrix können Sie als Liste von Listen speichern:

```
mat = [[1, 0, 0],  
       [0, 1, 0],  
       [0, 0, 1]]
```

Sichtbarkeit und Lebensdauer von Variablen in Funktionen

- Was gibt folgender Code aus?

```
x = 1
def f():
    print x
f()
```

- Was passiert jetzt?

```
x = 1
def f():
    x = 5
    print x
f()
print x
```

- Und hier?

```
def f():
    x = 5
f()
print x
```

- Verschachtelt:

```
x = 1
def f_aussen():
    x = 5
    def f_innen():
        print x
    return f_innen

f_innen = f_aussen()
f_innen()
```

- ???

```
def f(liste=[]):
    liste.append(1)
    print liste
f()
f()
f()
f()
```

Versuchen Sie, die Resultate der einzelnen Code-Fragmente zu erklären!

Aufgabe Cäsar Verschlüsselung

Die *Cäsar Verschlüsselung* ist eine Verschlüsselung, bei der jeder Buchstabe des zu verschlüsselnden Textes mit einem anderen Buchstaben ersetzt wird, dessen Position im Alphabet um n verschoben ist.

Beispielsweise könnte man jeden Buchstaben um 4 Positionen verschieben:

```
a -> e
b -> f
...
z -> d
```

- Schreiben Sie eine Funktion, die für Wörter, die aus Kleinbuchstaben bestehen, die sogenannte *ROT13* Verschlüsselung implementiert (also die Cäsar Verschlüsselung für $n=13$).

```
rot13("hallo welt")
```

Ausgabe:

```
'unyyb jryg'
```

- Hierfür ist es hilfreich, ein Dictionary zu erstellen, das jedem Buchstaben seinen “verschlüsselten” Buchstaben zuweist

- Kleine Hilfe:

```
alphabet = 'abcdefghijklmnopqrstuvwxyz'
```

Erstellen Sie nun daraus einen String `alphabet_verschoben` (Slicing hilft!), der mit dem 13. Buchstaben beginnt.

- Wie sieht die Funktion aus, die verschlüsselten Text wieder entschlüsselt?

Pascalsches Dreieck

(Quelle: https://de.wikipedia.org/wiki/Pascalsches_Dreieck)

Das Pascalsche Dreieck ist ein Dreieck aus Zahlen. In jeder Zeile sind die erste und die letzte Zahl jeweils Einsen. Für die weiteren Zahlen gilt folgende rekursive Formel: $z(\text{zeile}, i) = z(\text{zeile}-1, i-1) + z(\text{zeile}-1, i)$.

So gilt beispielsweise für die Zahl 6 (vierte Zeile) im obigen Dreieck: sie steht in der 4. Zeile (bei Null anfangen zu zählen!) an zweiter Stelle, und ist die Summe aus der Zahl in der 3. Zeile an erster Stelle und der Zahl in der 3. Zeile an zweiter Stelle.

Schreiben Sie eine Funktion `pascal_zeile`, die für eine gegebene Zeilennummer die Einträge des Pascalschen Dreiecks ausgibt.

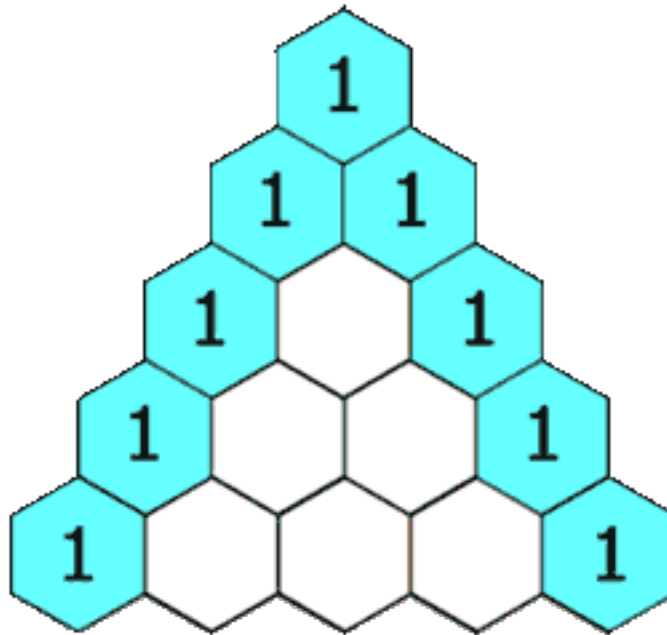


Figure 1: PascalschesDreieck

Aufgabe Student

Schreiben Sie eine Klasse Student mit folgenden Eigenschaften:

- Ein neues Studenten-Objekt wird erstellt, indem man dem Konstruktor (d.h. der **init** Funktion Namen und Vornamen des Studenten übergibt:
python student = Student(name="Müller", vorname="Heinz")
- Die Klasse soll eine Objektvariable *noten_liste* besitzen, die zu Beginn eine leere Liste ist, sowie eine Methode *neue_note*, die der *noten_liste* eine neue Note hinzufügt
- Schreiben Sie eine Methode *durchschnitts_note*, die den Durchschnitt aller Noten in *noten_liste* bildet