

## Algorithmen

### Aufgabe 1

Eine Multiplikation ist eine wiederholte Addition von Zahlen.

Bsp:  $4 \cdot 3 = 12$  kann als wiederholte Addition von  $4 + 4 + 4$  ausgedrückt werden.

- Erstelle ein Programm `mul.py`
- Schreibe eine Funktion `mul(a, b)` welche eine Multiplikation als wiederholte Addition durchführt
- Teste die Funktion ausführlich!

### Aufgabe 2

#### Sieb des Eratosthenes

Suche aller Primzahlen von 2 bis 100.

Wie funktioniert der Algorithmus?

- Zunächst werden alle Zahlen von 2 bis 100 in einer Liste gespeichert
- Beginnend bei 2 werden alle Vielfachen von 2 in der Liste als Vielfache markiert bis ein Vielfaches  $i = 100$  erreicht wird.
- Nach diesem Schritt wird wiederholt jede unmarkierte Zahl  $k$  aus der Liste genommen und die Vielfachen dieser Zahl markiert bis  $k = 100$  ist

Aufgabe:

- Erstelle ein Programm `sive.py` und lass es mit dem vorgestellten Algorithmus Primzahlen suchen
- die Erstellung der List soll in einer eigenen Funktion erfolgen, welche einen Parameter `max` hat, der maximale Anzahl an Zahlen definiert
- erstelle eine eigene Funktion `sive(list)`, welche die Liste auf Primzahlen testet und eine neue Liste mit den gefunden Primzahlen ausgibt
- Gib alle gefunden Primzahlen auf der Konsole mit `print` aus

## Aufabe 3

Min Sort - *etwas schwereres*

- Eine Liste besteht aus zwei Teilen: unsortierter Teil **U** und sortierter Teil **S**
- Am Anfang ist **S** noch leer und **U** ist die gesamte Liste
- Suche das Minimum in **U** und vertausche es mit dem ersten Element **i** von **U**
- Die Liste ist nun bis zu diesem Teil sortiert.
- **U** besteht nun aus der Liste **i+1** bis **len(list)**
- wiederhole den Vorgang so lange bis die Größe von **S = len(list)** ist und die Länge von **U = 0**

Beispiel:

4	2	1	6	3	5
4	2	1	6	3	5
1	2	4	6	3	5
1	2	4	6	3	5
1	2	4	6	3	5
1	2	3	6	4	5
1	2	3	6	4	5
1	2	3	4	6	5
1	2	3	4	5	6

Aufgabe:

- Schreib ein Programm `minsort.py` und implementiere `Minsort` in einer eigenen Funktion
- teste deinen Algorithmus mit verschiedenen unsortierten Integer-Listen
- Tipp: um das Debugging zu erleichtern gib die Zwischenschritte mit dem `print` Befehl aus