

A0.

1. Implementieren Sie eine Klasse Schokolade. Objekte der Klasse Schokolade haben: Name und Preis. Der Name soll ein String sein. Der Preis soll ein int sein.
2. Implementieren Sie eine Methode, die eine Liste von Schokoladen als Parameter bekommt und den gesamten Preis aller Milka Schokoladen in der Liste berechnet. Alle Schokoladen, deren Preis unter 10 liegt, werden ignoriert!

A. For/While-Schleife und Rekursion sind nicht erlaubt!

B. Beispiel: Für die folgende

Liste l = [Schokolade("milka", 10), Schokolade("milka", 20), Schokolade("laura", 5), Schokolade("Milka", 7), Schokolade("kandia",10)]

gibt die Methode aus: 30

A1.

Ein Kollege hat eine Sorter-Klasse umgesetzt. Die Klasse besitzt eine sort(self, data) Methode und sieht so aus:

```
class AwesomeSorter:
    def __init__(self, data):
        self.data = data #eine Liste

    def sort(self, data):
        #hier is where the magic happens
        #Nehmen Sie an, dass die Methode richtig implementiert wurde und die gibt die korrekt sortierte Liste zurück.

        return data
```

Ihre Aufgabe ist es, eine zusätzliche Sorter-Klasse (auch mit einer Sort-Methode) zu implementieren. Diese Sort-Methode soll aber auch die sortierten Daten in einer Datei speichern, ohne die Logik fürs Sorten wieder umzusetzen.

A. Welche Alternativen haben Sie, um den Code (die Sort-Methode) des Kollegens zu verwenden? Beschreiben Sie zwei Alternativen. Copy/Paste und die Standard Sort-Methoden (sort, sorted) von Python sind natürlich nicht erlaubt.

B. Implementieren Sie die beiden Alternativen.

A2.

Beim Spazieren im Wald fand der Tom ein altes Papier, das eine Reihe von Wörtern enthält:

```
+>+<+>>+
>><<+|+>+
+>>>+2<<<+
...
```

Er fragte sich, ob dies ein Geheimcode ist und beschloss, einen Algorithmus zu schreiben, der diese "geheimen" Nachrichten entschlüsseln kann. Seine Idee ist es, mit einer großen Zeichenfolge zu beginnen, die alle Buchstaben und Zahlen enthält: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789. Anschließend gab er jedem der speziellen Symbole im Code eine bestimmte Bedeutung:

```
> bewegt einen Zeiger vorwärts
< bewegt einen Zeiger rückwärts
+ druckt das aktuelle Symbol
+x druckt das aktuelle Symbol x-mal (0 <x <10)
| Entfernt das aktuelle Symbol
$ kopiert das aktuelle Symbol
```

Ihre Aufgabe ist es, Tom zu helfen, indem Sie eine Python-Funktion umsetzen sollen, die eine Zeichenfolge solcher Symbole interpretiert und die Ausgabe auf dem Bildschirm ausgibt.

Beispiel:

```
encode('+') => A
```

```
encode (+>+<+>>+) => ABAC  
encode ('>><<+|+>+') => ABC  
encode ('+>>>+2<<<+') => ADDA
```

Hinweis: Der Zeiger beginnt immer bei 0