1. (2 Punkte) Gegeben sei die Tupelliste `a = [(3,5),(7,3),(1,2),(10,1)]`. Erzeuge mit dem lambda-Operator zwei Listen b1 und b2. b1 ist sortiert nach der Summe der Komponenten, b2 ist absteigend nach der 2. Komponente sortiert.

Lösung:

```
b1 = sorted(a,key = lambda x: x[0]+x[1])
b2 = sorted(a,key = lambda x: x[1], reverse=True)
```

2. (2 Punkte) Gegeben sei die Tupelliste `a = [(3,5,4),(7,3,2),(1,2,1),(10,1,6)]`. Erzeuge mit dem lambda-Operator zwei Listen b1 und b2. b1 ist sortiert nach dem Produkt der drei Komponenten, b2 ist absteigend nach der letzten Komponente sortiert.

Lösung:

```
b1 = sorted(a,key = lambda x: x[0]*x[1]*x[2])
b2 = sorted(a,key = lambda x: x[2], reverse=True)
```

3. (3 Punkte) Gegeben sei die Liste:

```
a = []
a.append("Emma Elsenpeter aus Willegassen")
a.append("Emilia Poggenpohl aus Mastholte")
a.append("Clara Mörtenkötter aus Haddenhausen")
a.append("Noah Schniggendiller aus Barntrup")
a.append("Ben Trendelberend aus Brakelsiek")
a.append("Leon Eschengerd aus Altendonop")
```

Schreibe jeweils eine Zeile um die Liste a zu sortieren nach
- Vorname
- Nachname
- Ort (abwärts)

Lösung:

```
a.sort(key = lambda x: x.split()[0])
a.sort(key = lambda x: x.split()[1])
a.sort(key = lambda x: x.split()[3],reverse=True)
```

4. (2 Punkte) Gegeben sei die Liste `a = [2301,5401,1303,6403,4302,2402]`.

Sortiere die Liste mit einer Zeile nach den 100er-Zahlen. Das Ergebnis sieht dann so aus:

`2301 1303 4302 5401 6403 2402`.

Lösung:

```
a.sort(key = lambda x: (x%1000)//100)
```

5. (2 Punkte) Gegeben sei eine Liste mit positiven ganzen Zahlen. Sortiere die Liste abwärts nach der Quersumme der Zahlen.

Lösung:

```
a.sort(key = lambda x: sum([int(k) for k in list(str(x))]),reverse=True)
```