

Aufgabe 5: Marktwaaage

20. November 2021

Team-ID: 00821

Team: IaC 0.9524

Bearbeiter dieser Aufgabe: Paul Buda

Lösungsidee:

Zuerst muss man alle Möglichkeiten betrachten, wie man die Gewichte auf der Waage verteilen kann. Danach muss man doppelt erzielte Gewichte aussortieren, wobei man hier die Möglichkeiten behält, für welche man die geringste Anzahl an Gewichten benötigt, bzw. die Summe aller benötigten Gewichte am geringsten ist. Jetzt muss man nur noch zu allen gesuchten Gewichten, das Nächstmögliche finden und ausgeben, welche Gewichte man dazu auf welche Seiten legen muss.

Umsetzung:

Zuerst wird die Bibliothek numpy importiert um schneller und effektiver mit Listen bzw. Arrays rechnen zu können, da wir davon viele benötigen.

Danach liest das Programm alle Gewichte aus der Datei aus und speichert alle verschieden schwere Gewichte einmal in einer Liste.

Zudem generiert man sich eine Liste in welcher alle Möglichkeiten, wie man die Gewichte auf der Waage anordnen kann und jeweils das damit erzielte Gewicht, vorhanden sind. Um Rechenaufwand zu sparen kann man statt dem Gewicht von 10g (bzw. -10g wenn es auf der anderen Seite liegt oder 0g wenn man es gar nicht benutzt), wenn dieses Gewicht mehrmals (bspw. dreimal) vorhanden ist direkt mit den Gewichten 30g, 20g, 10g, 0g, -10g, -20g, -30g rechnen, so kann es auch nicht passieren, dass man das gleiche Gewicht sowohl links als auch rechts auf die Waage legen muss.

Bsp. für 1x10g und 2x50g: `[[-10, 0, 10], [-100, -50, 0, 50, 100]]`

Jetzt werden mittels der Meshgrid Funktion alle Möglichen Kombinationsmöglichkeiten der Gewichten berechnet, diese Liste wird wie unten im Beispiel auf die Hälfte der Länge reduziert, da man immer zweimal die gleiche Verteilung nur mit negativem Vorzeichen bekommt.

Bsp. für 1x10g und 2x50g: `[[-10, -100], [0, -100], [10, -100], [-10, 50], [0, 50], [10, 50], [-10, 0]]`

`[0, 0]` kann weggelassen werden, da kein Gewicht auf der Waage liegt

`[10, 0]` kann weggelassen werden, da Äquivalent zu `[-10, 0] * -1` usw.

Um weiter Rechenleistung sparen zu können werden dann alle möglichen Verteilungen mit dem Betrag der Summe kleiner 11kg aussortiert, diese werden später vermutlich nicht mehr benötigt (da der Abstand zu 10kg schon 10% beträgt und die Messungenauigkeit somit hoch wird).

Im Folgenden wird die Liste um die Summe der Gewichte erweitert.

Bsp. für 1x10g und 2x50g: `[[[-10, -100], -110], [[0, -100], -100], [[10, -100], -90], [[-10, -50], -60], ...]`

Danach wird die Liste nach Anzahl der benötigten Gewichten und Summe aller benötigten Gewichte sortiert, um später bei gleichen Summen der Gewichte jene Möglichkeiten zu behalten für welche man die wenigsten (bei gleich wenigen, die leichtesten) Gewichte zu verwenden. Zwischendurch werden die benötigten Gewichte und Summe der Gewichte noch negiert, falls die Summe der Gewichte negativ ist (tauschen der Seiten der Gewichte von rechts nach links bzw. links nach rechts).

Bsp. für 1x10g und 2x50g: `[[[10, 50], 60], [[-10, 100], 90], [[0, 100], 100], [[10, 100], 110], ...]`

Zum Schluss wird jetzt mittels einer For-Schleife welche von 10 bis 10000 in Zehner schritten zählt noch zu jedem gesuchten Gewicht das nächstmöglich Erzielbare gesucht und dann ausgegeben, welche Gewicht man dazu auf welche Seite der Waage legen muss.

Beispiele:

Bei Gewichtsstücke 0 erzeugt das Programm folgende Ausgabe:

```
10g -> 10g
Rechts: 1*10g
Links:
20g -> 20g
Rechts: 2*10g
Links:
30g -> 30g
Rechts: 3*10g
Links:
40g -> 40g
Rechts: 1*50g
Links: 1*10g
50g -> 50g
Rechts: 1*50g
Links:
60g -> 60g
Rechts: 1*10g, 1*50g
Links:
70g -> 70g
Rechts: 2*10g, 1*50g
Links:
80g -> 80g
Rechts: 3*10g, 1*50g
Links:
90g -> 90g
Rechts: 1*100g
Links: 1*10g
100g -> 100g
Rechts: 1*100g
Links:
110g -> 110g
Rechts: 1*10g, 1*100g
Links:
120g -> 120g
Rechts: 2*10g, 1*100g
Links:
130g -> 130g
Rechts: 3*10g, 1*100g
Links:
140g -> 140g
Rechts: 1*50g, 1*100g
Links: 1*10g
150g -> 150g
Rechts: 1*50g, 1*100g
Links:
160g -> 160g
Rechts: 1*10g, 1*50g, 1*100g
Links:
170g -> 170g
Rechts: 2*10g, 1*50g, 1*100g
Links:
180g -> 180g
Rechts: 3*10g, 1*50g, 1*100g
```

Links:
190g -> 190g
Rechts: 2*100g
Links: 1*10g
200g -> 200g
Rechts: 2*100g
Links:
210g -> 210g
Rechts: 1*10g, 2*100g
Links:
220g -> 220g
Rechts: 2*10g, 2*100g
Links:
230g -> 230g
Rechts: 3*10g, 2*100g
Links:
240g -> 240g
Rechts: 1*50g, 2*100g
Links: 1*10g
250g -> 250g
Rechts: 1*50g, 2*100g
Links:
260g -> 260g
Rechts: 1*10g, 1*50g, 2*100g
Links:
270g -> 270g
Rechts: 2*10g, 1*50g, 2*100g
Links:
280g -> 280g
Rechts: 3*10g, 1*50g, 2*100g
Links:
290g -> 290g
Rechts: 3*100g
Links: 1*10g
300g -> 300g
Rechts: 3*100g
Links:
310g -> 310g
Rechts: 1*10g, 3*100g
Links:
320g -> 320g
Rechts: 2*10g, 3*100g
Links:
330g -> 330g
Rechts: 3*10g, 3*100g
Links:
340g -> 340g
Rechts: 1*50g, 3*100g
Links: 1*10g
350g -> 350g
Rechts: 1*50g, 3*100g
Links:
360g -> 360g
Rechts: 1*10g, 1*50g, 3*100g
Links:
370g -> 370g
Rechts: 2*10g, 1*50g, 3*100g
Links:

380g -> 380g
Rechts: 3*10g, 1*50g, 3*100g
Links:
390g -> 390g
Rechts: 2*50g, 3*100g
Links: 1*10g
400g -> 400g
Rechts: 2*50g, 3*100g
Links:
410g -> 410g
Rechts: 1*10g, 2*50g, 3*100g
Links:
420g -> 420g
Rechts: 2*10g, 2*50g, 3*100g
Links:
430g -> 430g
Rechts: 3*10g, 2*50g, 3*100g
Links:
440g -> 440g
Rechts: 1*500g
Links: 1*10g, 1*50g
450g -> 450g
Rechts: 1*500g
Links: 1*50g
460g -> 460g
Rechts: 1*10g, 1*500g
Links: 1*50g
470g -> 470g
Rechts: 1*500g
Links: 3*10g
480g -> 480g
Rechts: 1*500g
Links: 2*10g
490g -> 490g
Rechts: 1*500g
Links: 1*10g
500g -> 500g
Rechts: 1*500g
Links:

[...]

9500g -> 9500g
Rechts: 3*500g, 3*1000g, 1*5000g
Links:
9510g -> 9510g
Rechts: 1*10g, 3*500g, 3*1000g, 1*5000g
Links:
9520g -> 9520g
Rechts: 2*10g, 3*500g, 3*1000g, 1*5000g
Links:
9530g -> 9530g
Rechts: 3*10g, 3*500g, 3*1000g, 1*5000g
Links:
9540g -> 9540g
Rechts: 1*50g, 3*500g, 3*1000g, 1*5000g
Links: 1*10g

9550g -> 9550g
Rechts: 1*50g, 3*500g, 3*1000g, 1*5000g
Links:
9560g -> 9560g
Rechts: 1*10g, 1*50g, 3*500g, 3*1000g, 1*5000g
Links:
9570g -> 9570g
Rechts: 2*10g, 1*50g, 3*500g, 3*1000g, 1*5000g
Links:
9580g -> 9580g
Rechts: 3*10g, 1*50g, 3*500g, 3*1000g, 1*5000g
Links:
9590g -> 9590g
Rechts: 1*100g, 3*500g, 3*1000g, 1*5000g
Links: 1*10g
9600g -> 9600g
Rechts: 1*100g, 3*500g, 3*1000g, 1*5000g
Links:
9610g -> 9610g
Rechts: 1*10g, 1*100g, 3*500g, 3*1000g, 1*5000g
Links:
9620g -> 9620g
Rechts: 2*10g, 1*100g, 3*500g, 3*1000g, 1*5000g
Links:
9630g -> 9630g
Rechts: 3*10g, 1*100g, 3*500g, 3*1000g, 1*5000g
Links:
9640g -> 9640g
Rechts: 1*50g, 1*100g, 3*500g, 3*1000g, 1*5000g
Links: 1*10g
9650g -> 9650g
Rechts: 1*50g, 1*100g, 3*500g, 3*1000g, 1*5000g
Links:
9660g -> 9660g
Rechts: 1*10g, 1*50g, 1*100g, 3*500g, 3*1000g, 1*5000g
Links:
9670g -> 9670g
Rechts: 2*10g, 1*50g, 1*100g, 3*500g, 3*1000g, 1*5000g
Links:
9680g -> 9680g
Rechts: 3*10g, 1*50g, 1*100g, 3*500g, 3*1000g, 1*5000g
Links:
9690g -> 9690g
Rechts: 2*100g, 3*500g, 3*1000g, 1*5000g
Links: 1*10g
9700g -> 9700g
Rechts: 2*100g, 3*500g, 3*1000g, 1*5000g
Links:
9710g -> 9710g
Rechts: 1*10g, 2*100g, 3*500g, 3*1000g, 1*5000g
Links:
9720g -> 9720g
Rechts: 2*10g, 2*100g, 3*500g, 3*1000g, 1*5000g
Links:
9730g -> 9730g
Rechts: 3*10g, 2*100g, 3*500g, 3*1000g, 1*5000g
Links:
9740g -> 9740g

Rechts: 1*50g, 2*100g, 3*500g, 3*1000g, 1*5000g
Links: 1*10g
9750g -> 9750g
Rechts: 1*50g, 2*100g, 3*500g, 3*1000g, 1*5000g
Links:
9760g -> 9760g
Rechts: 1*10g, 1*50g, 2*100g, 3*500g, 3*1000g, 1*5000g
Links:
9770g -> 9770g
Rechts: 2*10g, 1*50g, 2*100g, 3*500g, 3*1000g, 1*5000g
Links:
9780g -> 9780g
Rechts: 3*10g, 1*50g, 2*100g, 3*500g, 3*1000g, 1*5000g
Links:
9790g -> 9790g
Rechts: 3*100g, 3*500g, 3*1000g, 1*5000g
Links: 1*10g
9800g -> 9800g
Rechts: 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9810g -> 9810g
Rechts: 1*10g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9820g -> 9820g
Rechts: 2*10g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9830g -> 9830g
Rechts: 3*10g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9840g -> 9840g
Rechts: 1*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links: 1*10g
9850g -> 9850g
Rechts: 1*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9860g -> 9860g
Rechts: 1*10g, 1*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9870g -> 9870g
Rechts: 2*10g, 1*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9880g -> 9880g
Rechts: 3*10g, 1*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9890g -> 9890g
Rechts: 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links: 1*10g
9900g -> 9900g
Rechts: 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9910g -> 9910g
Rechts: 1*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9920g -> 9920g
Rechts: 2*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
Links:
9930g -> 9930g
Rechts: 3*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g

Links:
 9940g -> 9930g
 Rechts: 3*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
 Links:
 9950g -> 9930g
 Rechts: 3*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
 Links:
 9960g -> 9930g
 Rechts: 3*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
 Links:
 9970g -> 9930g
 Rechts: 3*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
 Links:
 9980g -> 9930g
 Rechts: 3*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
 Links:
 9990g -> 9930g
 Rechts: 3*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
 Links:
 10000g -> 9930g
 Rechts: 3*10g, 2*50g, 3*100g, 3*500g, 3*1000g, 1*5000g
 Links:

Es wird immer zuerst angegeben, wie nah man an das gesuchte Gewicht ran kommt und dann welche Gewichtsstücke man nach rechts bzw. links legen muss. Man sieht auch, dass immer die minimale Anzahl an Gewichten mit der geringsten Gesamtmasse ausgegeben wird. So wird für 30g 3*10g rechts angegeben und nicht 1*50g links und 2*10g Rechts da die Gesamtmasse von 30g kleiner als die Gesamtmasse von 70g.

Bei Gewichtsstücke 1 erzeugt das Programm folgende Ausgabe:

10g -> 10g
 Rechts: 2*42g
 Links: 5*371g
 20g -> 19g
 Rechts: 2*42g, 3*127g, 1*2000g
 Links: 2*371g
 30g -> 29g
 Rechts: 3*42g, 1*2000g
 Links: 1*127g, 3*371g
 40g -> 42g
 Rechts: 3*127g, 4*371g
 Links: 1*42g
 50g -> 51g
 Rechts: 3*42g, 1*2000g
 Links: 1*371g
 60g -> 60g
 Rechts: 2*127g, 1*2000g
 Links: 1*42g, 1*371g
 70g -> 70g
 Rechts: 1*2000g
 Links: 2*127g, 2*371g
 80g -> 84g
 Rechts: 3*127g, 4*371g
 Links:
 90g -> 93g

Rechts: 1*2000g
Links: 3*42g
100g -> 100g
Rechts: 2*371g
Links: 1*42g, 3*127g, 1*2000g
110g -> 110g
Rechts: 1*371g
Links: 1*42g, 1*2000g
120g -> 118g
Rechts: 2*42g, 2*127g, 4*371g
Links:
130g -> 128g
Rechts:
Links: 1*42g, 1*127g, 4*371g
140g -> 141g
Rechts: 2*42g, 3*127g, 2*371g
Links: 1*2000g
150g -> 151g
Rechts: 3*42g, 1*371g
Links: 1*127g, 1*2000g
160g -> 160g
Rechts: 3*42g, 2*127g, 4*371g
Links:
170g -> 170g
Rechts:
Links: 1*127g, 4*371g
180g -> 178g
Rechts: 3*42g, 1*127g, 1*2000g
Links: 1*371g
190g -> 188g
Rechts: 1*2000g
Links: 3*42g, 3*127g, 1*371g
200g -> 201g
Rechts: 1*127g, 5*371g
Links:
210g -> 211g
Rechts: 1*42g, 4*371g
Links: 3*127g
220g -> 220g
Rechts: 1*127g, 1*2000g
Links: 3*42g
230g -> 230g
Rechts: 1*2000g
Links: 2*42g, 3*127g, 1*371g
240g -> 239g
Rechts: 1*42g, 1*2000g
Links: 1*127g, 2*371g
250g -> 253g
Rechts: 2*42g, 4*371g
Links: 3*127g
260g -> 261g
Rechts: 1*42g, 1*2000g
Links:
270g -> 269g
Rechts: 2*371g
Links: 2*127g, 1*2000g
280g -> 281g
Rechts: 2*42g, 1*2000g

Links: 1*127g, 2*371g
290g -> 287g
Rechts: 3*42g, 3*127g, 4*371g
Links:
300g -> 297g
Rechts:
Links: 4*371g
310g -> 310g
Rechts: 3*371g
Links: 3*42g, 3*127g, 1*2000g
320g -> 320g
Rechts: 2*371g
Links: 3*42g, 1*2000g
330g -> 329g
Rechts: 3*127g, 5*371g
Links: 3*42g
340g -> 339g
Rechts: 1*42g
Links: 4*371g
350g -> 352g
Rechts: 3*371g
Links: 2*42g, 3*127g, 1*2000g
360g -> 361g
Rechts: 6*371g
Links: 2*42g
370g -> 370g
Rechts: 1*42g, 2*127g, 5*371g
Links:
380g -> 380g
Rechts: 2*42g, 4*371g
Links: 2*127g
390g -> 390g
Rechts: 2*42g, 3*127g, 1*2000g
Links: 1*371g
400g -> 400g
Rechts: 3*42g, 1*2000g
Links: 1*127g, 2*371g
410g -> 413g
Rechts: 3*127g, 5*371g
Links: 1*42g
420g -> 422g
Rechts: 3*42g, 1*2000g
Links:
430g -> 431g
Rechts: 2*127g, 1*2000g
Links: 1*42g
440g -> 441g
Rechts: 1*2000g
Links: 2*127g, 1*371g
450g -> 446g
Rechts: 1*127g, 6*371g
Links: 3*42g
460g -> 456g
Rechts: 5*371g
Links: 2*42g, 3*127g
470g -> 473g
Rechts: 2*127g, 1*2000g
Links:

480g -> 480g
Rechts: 2*42g, 2*371g
Links: 1*127g, 1*2000g
490g -> 489g
Rechts: 2*42g, 2*127g, 5*371g
Links:
500g -> 499g
Rechts: 3*42g, 4*371g
Links: 2*127g

[...]

9500g -> 9501g
Rechts: 1*42g, 1*127g, 3*371g, 5*2000g
Links:
9510g -> 9511g
Rechts: 2*42g, 2*371g, 5*2000g
Links: 3*127g
9520g -> 9517g
Rechts: 3*42g, 1*127g, 5*2000g
Links: 5*371g
9530g -> 9527g
Rechts: 5*2000g
Links: 3*42g, 3*127g, 5*371g
9540g -> 9536g
Rechts: 5*2000g
Links: 1*127g, 6*371g
9550g -> 9554g
Rechts: 2*371g, 5*2000g
Links: 2*127g
9560g -> 9559g
Rechts: 1*127g, 5*2000g
Links: 3*42g, 4*371g
9570g -> 9569g
Rechts: 5*2000g
Links: 2*42g, 3*127g, 5*371g
9580g -> 9578g
Rechts: 1*42g, 5*2000g
Links: 1*127g, 6*371g
9590g -> 9587g
Rechts: 3*127g, 3*371g, 5*2000g
Links: 3*42g
9600g -> 9600g
Rechts: 1*42g, 5*2000g
Links: 4*371g
9610g -> 9610g
Rechts: 1*42g, 3*127g, 5*2000g
Links: 5*371g
9620g -> 9620g
Rechts: 2*42g, 1*127g, 3*371g, 5*2000g
Links:
9630g -> 9630g
Rechts: 3*42g, 2*371g, 5*2000g
Links: 3*127g
9640g -> 9639g
Rechts: 2*371g, 5*2000g
Links: 1*42g, 1*127g
9650g -> 9653g

Rechts: 5*2000g
Links: 3*127g, 5*371g
9660g -> 9661g
Rechts: 4*371g, 5*2000g
Links: 1*42g
9670g -> 9671g
Rechts: 3*127g, 3*371g, 5*2000g
Links: 1*42g
9680g -> 9681g
Rechts: 2*371g, 5*2000g
Links: 1*127g
9690g -> 9686g
Rechts: 2*127g, 5*2000g
Links: 3*42g, 4*371g
9700g -> 9703g
Rechts: 4*371g, 5*2000g
Links:
9710g -> 9713g
Rechts: 3*127g, 3*371g, 5*2000g
Links:
9720g -> 9719g
Rechts: 2*42g, 5*2000g
Links: 4*371g
9730g -> 9729g
Rechts: 2*42g, 3*127g, 5*2000g
Links: 5*371g
9740g -> 9739g
Rechts: 3*42g, 5*2000g
Links: 1*127g, 6*371g
9750g -> 9747g
Rechts: 2*42g, 2*127g, 3*371g, 5*2000g
Links:
9760g -> 9761g
Rechts: 3*42g, 5*2000g
Links: 4*371g
9770g -> 9770g
Rechts: 2*127g, 5*2000g
Links: 1*42g, 4*371g
9780g -> 9780g
Rechts: 5*2000g
Links: 2*127g, 5*371g
9790g -> 9789g
Rechts: 3*42g, 2*127g, 3*371g, 5*2000g
Links:
9800g -> 9799g
Rechts: 3*371g, 5*2000g
Links: 3*42g, 2*127g
9810g -> 9812g
Rechts: 2*127g, 5*2000g
Links: 4*371g
9820g -> 9822g
Rechts: 1*42g, 5*2000g
Links: 2*127g, 5*371g
9830g -> 9830g
Rechts: 1*127g, 4*371g, 5*2000g
Links:
9840g -> 9840g
Rechts: 1*42g, 3*371g, 5*2000g

Links: 3*127g
 9850g -> 9846g
 Rechts: 2*42g, 1*127g, 5*2000g
 Links: 4*371g
 9860g -> 9856g
 Rechts: 3*42g, 5*2000g
 Links: 3*127g, 5*371g
 9870g -> 9872g
 Rechts: 1*42g, 1*127g, 4*371g, 5*2000g
 Links:
 9880g -> 9882g
 Rechts: 2*42g, 3*371g, 5*2000g
 Links: 3*127g
 9890g -> 9888g
 Rechts: 3*42g, 1*127g, 5*2000g
 Links: 4*371g
 9900g -> 9898g
 Rechts: 5*2000g
 Links: 3*42g, 3*127g, 4*371g
 9910g -> 9907g
 Rechts: 5*2000g
 Links: 1*127g, 5*371g
 9920g -> 9916g
 Rechts: 3*42g, 3*127g, 3*371g, 5*2000g
 Links:
 9930g -> 9930g
 Rechts: 1*127g, 5*2000g
 Links: 3*42g, 3*371g
 9940g -> 9940g
 Rechts: 5*2000g
 Links: 2*42g, 3*127g, 4*371g
 9950g -> 9949g
 Rechts: 1*42g, 5*2000g
 Links: 1*127g, 5*371g
 9960g -> 9958g
 Rechts: 3*127g, 4*371g, 5*2000g
 Links: 3*42g
 9970g -> 9971g
 Rechts: 1*42g, 5*2000g
 Links: 3*371g
 9980g -> 9981g
 Rechts: 1*42g, 3*127g, 5*2000g
 Links: 4*371g
 9990g -> 9990g
 Rechts: 5*371g, 5*2000g
 Links: 2*42g
 10000g -> 10000g
 Rechts: 3*127g, 4*371g, 5*2000g
 Links: 2*42g

Hier sieht man jetzt, dass die 20g nicht genau erreicht werden können und das Programm deshalb die Verteilung für 19g angibt, das 19g am nächst möglichen an 20g liegt.

Bei Gewichtsstücke 5 erzeugt das Programm folgende Ausgabe:

10g -> 11g

Rechts: 1*99480g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
20g -> 20g
Rechts: 1*11g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g
30g -> 31g
Rechts: 1*11g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links:
40g -> 42g
Rechts: 1*11g, 1*99480g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
50g -> 47g
Rechts: 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*99480g
60g -> 58g
Rechts: 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g
70g -> 69g
Rechts: 1*99480g, 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g
80g -> 78g
Rechts: 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g
90g -> 89g
Rechts: 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
100g -> 100g
Rechts: 1*99480g, 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
110g -> 109g
Rechts: 1*11g, 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g
120g -> 120g
Rechts: 1*11g, 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
130g -> 131g
Rechts: 1*11g, 1*99480g, 1*99511g, 1*599761g, 1*4497786g, 1*1499171g, 4*10499654g,
1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
140g -> 139g
Rechts: 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*99511g, 1*599761g
150g -> 150g
Rechts: 1*99480g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*11g, 1*99511g, 1*599761g
160g -> 159g
Rechts: 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g, 1*99511g, 1*599761g
170g -> 170g
Rechts: 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99511g, 1*599761g
180g -> 181g
Rechts: 1*99480g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99511g, 1*599761g
190g -> 190g
Rechts: 1*11g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g, 1*99511g, 1*599761g
200g -> 201g
Rechts: 1*11g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99511g, 1*599761g
210g -> 212g
Rechts: 1*11g, 1*99480g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99511g, 1*599761g
220g -> 217g
Rechts: 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*99480g, 1*599761g
230g -> 228g
Rechts: 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*599761g
240g -> 239g
Rechts: 1*99480g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*599761g
250g -> 248g
Rechts: 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g, 1*599761g
260g -> 259g
Rechts: 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
270g -> 270g
Rechts: 1*99480g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
280g -> 279g
Rechts: 1*11g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g, 1*599761g
290g -> 290g
Rechts: 1*11g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
300g -> 301g

Rechts: 1*11g, 1*99480g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
310g -> 306g
Rechts: 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*99480g, 1*599761g
320g -> 317g
Rechts: 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*599761g
330g -> 328g
Rechts: 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*599761g
340g -> 337g
Rechts: 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g, 1*599761g
350g -> 348g
Rechts: 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
360g -> 359g
Rechts: 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
370g -> 368g
Rechts: 1*11g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g, 1*599761g
380g -> 379g
Rechts: 1*11g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
390g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g,
1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
400g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g,
1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
410g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g,
1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
420g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g,
1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
430g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g,
1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
440g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g,
1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*599761g
450g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
460g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
470g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
480g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
490g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g
500g -> 390g
Rechts: 1*11g, 1*99480g, 1*99511g, 2*299836g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*599761g

[...]

9500g -> 9503g
Rechts: 1*11g, 1*99480g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
9510g -> 9508g
Rechts: 1*99511g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*99480g
9520g -> 9519g
Rechts: 1*99511g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g
9530g -> 9530g
Rechts: 1*99480g, 1*99511g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g
9540g -> 9539g
Rechts: 1*99511g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g
9550g -> 9550g
Rechts: 1*99511g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
9560g -> 9561g
Rechts: 1*99480g, 1*99511g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
9570g -> 9570g

Rechts: 1*11g, 1*99511g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g
9580g -> 9581g
Rechts: 1*11g, 1*99511g, 2*299836g, 2*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links:
9590g -> 9589g
Rechts: 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*99480g, 1*99511g
9600g -> 9600g
Rechts: 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*99511g
9610g -> 9611g
Rechts: 1*99480g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g, 1*99511g
9620g -> 9620g
Rechts: 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g, 1*99511g
9630g -> 9631g
Rechts: 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99511g
9640g -> 9642g
Rechts: 1*99480g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99511g
9650g -> 9651g
Rechts: 1*11g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99480g, 1*99511g
9660g -> 9662g
Rechts: 1*11g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99511g
9670g -> 9668g
Rechts: 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g,
3*661499326g, 1*1984505261g
Links: 1*99511g, 1*299836g, 1*599761g
9680g -> 9679g
Rechts: 1*99480g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*99511g, 1*299836g, 1*599761g
9690g -> 9689g
Rechts: 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g
9700g -> 9700g
Rechts: 1*99480g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g,
3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g
Links: 1*11g
9710g -> 9710g
Rechts: 1*11g, 1*99480g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g,
1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*99511g, 1*299836g, 1*599761g

9720g -> 9720g

Rechts: 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links:

9730g -> 9731g

Rechts: 1*99480g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links:

9740g -> 9740g

Rechts: 1*11g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*99480g

9750g -> 9751g

Rechts: 1*11g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links:

9760g -> 9762g

Rechts: 1*11g, 1*99480g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links:

9770g -> 9768g

Rechts: 1*99480g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*299836g, 1*599761g

9780g -> 9778g

Rechts: 1*99511g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*11g

9790g -> 9789g

Rechts: 1*99480g, 1*99511g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*11g

9800g -> 9799g

Rechts: 1*11g, 1*99480g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*299836g, 1*599761g

9810g -> 9809g

Rechts: 1*99511g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links:

9820g -> 9820g

Rechts: 1*99480g, 1*99511g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links:

9830g -> 9829g

Rechts: 1*11g, 1*99511g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links: 1*99480g

9840g -> 9840g

Rechts: 1*11g, 1*99511g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links:

9850g -> 9851g

Rechts: 1*11g, 1*99480g, 1*99511g, 1*299836g, 1*599761g, 3*4497786g, 1*1499171g, 4*10499654g, 1*41999427g, 3*94499810g, 1*283501867g, 3*661499326g, 1*1984505261g

Links:

9860g -> 9857g

[illegible]

Links: 1*299836g, 1*599761g

Hier muss man nun noch dazusagen, dass das Programm ca. 35sec rechnet, wobei die reshape Funktion welche nach der meshgrid Funktion ausgeführt werden muss davon die meiste Zeit benötigt. Zudem gibt es in der Konsole einen overflow error, dieser muss jedoch nicht weiter betrachtet werden, da dieser erst beim sortieren der Gewichte auftritt. So kann es zwar passiert, dass für ein bestimmtes Gewicht nicht die beste Möglichkeit der zu verteilenden Gewichte gefunden wird, jedoch wird dadurch sonst weitestgehend unnötig verbrauchter Speicher vermieden.

Quelltext:

```
#importiert numpy für schnellere Berechnungen mit arrays
from numpy import array, meshgrid, sum

#liest die Datei mit den Infos zu den Gewichtsstücken ein
weights = open("gewichtsstuecke.txt", "r")
#speichert die Anzahl verschieden schwerer Gewichtsstücke
numberWeights = int(weights.readline())

#Liste mit allen Gewichten
weightsList = []

#Liste mit möglicher Anzahl der Gewichte multipliziert mit dem Gewicht (Bsp. für 2*10g ->
[-20, -10, 0, 10, 20])
amounts = []

for r in range(numberWeights):
    weight, amount = weights.readline().split()

    weightsList.append(int(weight))

    amounts.append([r * int(weight) for r in range(int(amount) * -1, int(amount) + 1)])

#berechnet alle Kombinationsmöglichkeiten der Gewichte
exec("meshgrid = meshgrid(" + str(amounts)[1:-1] + ")")
combs = array(meshgrid).T.reshape(-1, numberWeights)

#es muss nur die Hälfte der Kombinationsmöglichkeiten betrachtet werden, da die andere
Hälfte genau den negativen Wert ergibt
combs = combs[:len(combs) // 2]

#berechnet die Summe der jeweiligen Verteilung
combsSum = sum(combs, axis = 1)

#es werden nur die Summen unter 11kg berücksichtigt
combsSum = combsSum[abs(combsSum) < 11000]

#erstellt eine Liste mit den Kombinationen und deren Summen [[[10, 100], 110], [[-10,
100], 90], ...]
possibleWeights = list(zip(combs, combsSum))

#sortiert die Gewichte nach Gewicht und dann nach Summe der einzelnen benötigten Gewichte
possibleWeights.sort(key = lambda w: (abs(w[1]), sum(abs(x) for x in w[0])))

#negiert die Gewichte falls die Summe der Gewichte negativ ist
```

```
possibleWeights = [[(-w if (weight[1] < 0) else w for w in weight[0]), abs(weight[1])) for
weight in possibleWeights]
```

```
#sortiert doppelt erzielte Summen aus, es werden hierbei die behalten, bei welchen man am
wenigsten Gewichte braucht
```

```
first = possibleWeights[1]
possibleWeights = [(possibleWeights[r + 1]) for r in range(len(possibleWeights) - 1)
if(possibleWeights[r + 1][1] != possibleWeights[r][1])]
possibleWeights.append(first)
```

```
#gibt die Ergebnisse aus
```

```
for r in range(10, 10010, 10):
```

```
    #sucht das Gewicht, welches am nächsten am gesuchten Gewicht liegt
```

```
    mini = min(possibleWeights, key = lambda x:abs(x[1] - r))
```

```
    weightsRight = [(str(int(mini[0][r] / weightsList[r])) + "*" + str(weightsList[r]) +
"g") for r in range(len(weightsList)) if(mini[0][r] > 0)]
```

```
    weightsLeft = [(str(int(abs(mini[0][r] / weightsList[r]))) + "*" + str(weightsList[r])
+ "g") for r in range(len(weightsList)) if(mini[0][r] < 0)]
```

```
    print(str(r) + "g -> " + str(mini[1]) + "g\nRechts: " + str(weightsRight)[1:-
1].replace("'", "") + "\nLinks: " + str(weightsLeft)[1:-1].replace("'", ""))
```