

```
1 def BK(bedrag, *claims):
2     Error = is_bankroetprobleem(bedrag,*claims)
3     if Error:
4         return Error
5     # komt neer op coalitionele procedure van Aumann en Maschler
6     from collections import OrderedDict
7     claims = sorted(list(claims))
8     subclaims = sum(claims)
9     coalitieomvang = len(claims)
10    result = []
11
12    for i in range(coalitieomvang - 1):
13        subbedrag = min(claims[i],bedrag) / 2
14        verlies = claims[i] - subbedrag
15        # iemand met een lagere claim dient niet meer uitbetaald te krijgen dan
16        # iemand met een hogere claim
17        if subbedrag >= bedrag/coalitieomvang:
18            subbedrag = bedrag/coalitieomvang
19            for _ in range(coalitieomvang):
20                result.append(round(subbedrag,2))
21            break
22        # iemand met een lagere claim dient niet meer te verliezen dan iemand met een
23        # hogere claim
24        elif (subclaims - coalitieomvang * verlies) < bedrag:
25            verlies = (subclaims - bedrag) / coalitieomvang
26            for j in range(coalitieomvang):
27                result.append(round(claims[j] - verlies, 2))
28            break
29        else:
30            result.append(round(subbedrag,2))
31            bedrag -= subbedrag
32            coalitieomvang -= 1
33            subclaims -= claims[i]
34
35    # de laatste persoon krijgt de rest van het bedrag
```